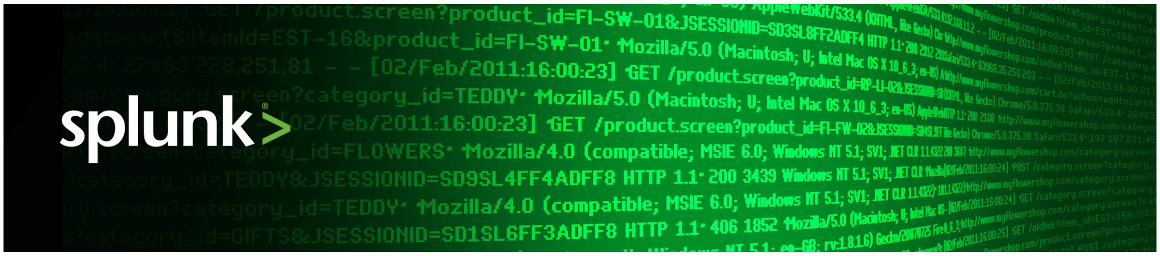

Splunk® Enterprise Alerting Manual 7.2.4

Generated: 11/25/2019 12:36 am

Copyright (c) 2019 Splunk Inc. All Rights Reserved

Table of Contents

Table of Contents

View and update alerts

Alerting overview

Getting started with alerts

Use alerts to monitor for and respond to specific events. Alerts use a saved search to look for events in real time or on a schedule. Alerts trigger when search results meet specific conditions. You can use alert actions to respond when alerts trigger.

This resource includes information, instructions, and scenarios for using alerts and alert actions. To start learning about alerts, see The alerting workflow. Review alerting options in Alert types.

The alerting workflow

Alerts combine a saved search, configurations for type and trigger conditions, and alert actions. Here are some details about how the different parts of an alert work together.

Search: What do you want to track?

> Start with a search for the events you want to track. Save the search as an alert.

Alert type: How often do you want to check for events?

> The alert uses the saved search to check for events. Adjust the alert type to configure how often the search runs. Use a scheduled alert to check for events on a regular basis. You can also use a real-time alert to monitor for events continuously.

Alert trigger conditions and throttling: How often do you want to trigger an alert?

> An alert does not have to trigger every time it generates search results. Set trigger conditions to manage when the alert triggers. You can also throttle an alert to control how soon the next alert can trigger after an initial alert.

Alert Action: What happens when the alert triggers?

When an alert triggers, it can initialize one or more alert actions. An alert action can notify you of a triggered alert and help you start responding to it. You can configure alert action frequency and type.

Choose an alert type

Alert types

There are two alert types, scheduled and real-time. Alert type definitions are based on alert search timing. Depending on the scenario, you can configure timing, triggering, and other behavior for either alert type.

Alert type comparison

Here is a comparison of scheduled and real-time alerts.

Alert type	When it searches for events	Triggering options	Throttling options
Scheduled	Searches according to a schedule. Choose from the available timing options or use a cron expression to schedule the search.	Specify conditions for triggering the alert based on result or result field counts. When a set of search results meets the trigger conditions, the alert can trigger one time or once for each of the results.	Specify a time period for suppression.
Real-time	Searches continuously.	**Per-result**: Triggers every time there is a search result.	Specify a time period and optional field values for suppression.
Real-time	Searches continuously.	**Rolling time window**: Specify conditions for triggering the alert based on result or result field counts within a rolling time window. For example, a real-time alert can trigger whenever there are more than ten results in a five	Specify a time period for suppression.

Alert type	When it searches for events	Triggering options	Throttling options
		minute window.	

Alert type and triggering scenarios

Once you choose a scheduled or a real-time alert, you can configure how results trigger the alert. Depending on the events that you are monitoring, you might need a real-time alert that triggers with every result or a scheduled alert that only triggers if results meet certain conditions. The following scenarios show different use cases for alert types and triggering.

Scheduled alert

Use a scheduled alert to search for events on a regular basis and monitor whether they meet specific conditions. A scheduled alert is useful if immediate or real-time monitoring is not a priority.

Scenarios

- An online retailer has a daily goal of 500 sales. An admin for the retailer creates a scheduled alert to monitor sales performance. The admin schedules the alert to search for sales events each day at 23:00. She configures the alert to trigger if the number of results is lower than 500.
- An admin wants to monitor how frequently users follow a bad link to the 404 error page. The admin creates a scheduled alert that searches for 404 errors every hour and triggers if there are more than 100 results.
- An admin creates a scheduled alert to check if a particular host has not sent any data to the Splunk platform in the last few hours. He schedules the alert to search for events from the host every three hours. The admin configures the alert to trigger if there are no search results.

Real-time alert

Real-time alerts search for events continuously. They can be useful in situations where immediate monitoring and responses are important. You can use real-time alerts that trigger once per result or only if certain conditions are met within a specific rolling time window.

Per-result triggering

A real-time alert with a per-result triggering condition is sometimes known as a "per-result alert". Use this alert type and triggering to search continuously for events and to receive notifications when events occur.

> **Caution:** In a high availability deployment, use per-result triggering with caution. If a peer is not available, a real-time search does not warn that the search might be incomplete. It is recommended to use a scheduled alert for this deployment.

Scenarios

Here are some examples of using a real-time alert with per-result triggering.

- A social networking website admin wants to know whenever login failures occur. She sets up a real-time alert to search for failed login attempts. She chooses a per-result trigger condition so that she can track every failed login attempt.

- An admin wants to monitor a set of hosts for errors in real-time. Some errors need a more urgent response than others. The admin sets up a real-time alert with a per-result trigger condition. He throttles the alert using a field that represents the less urgent error code and a one hour suppression period. The alert triggers for every urgent error, but at most once in an hour for less urgent errors.

Rolling time window triggering

A real-time alert with rolling time window triggering is sometimes known as a "rolling window alert". This alert type and triggering are useful when a specific time window is an important part of the event pattern you are monitoring in real time.

Scenarios

Here are some examples of using a real-time alert with rolling time window triggering.

- An admin wants a notification whenever a user has three failed logins within a ten minute period. The admin sets up a real-time alert to search for failed logins and configures a rolling ten minute time window. The admin throttles the alert so that it triggers only once in an hour for failed logins from the same user.

- An admin wants to know whenever a web application has more than five database connection errors in a minute. She configures a real-time alert to search for error events and specifies a one minute rolling window. If the search returns one result and then four more results five minutes later, the alert does not trigger. However, the alert triggers if the search returns five results within a one minute span.

Additional resources

See Throttle alerts to review additional throttling scenarios.

Create alerts

Create scheduled alerts

Create a scheduled alert to search for events on a regular schedule. You can configure scheduling, trigger conditions, and throttling to customize the alert.

To compare scheduled and real-time alerts, see Alert types. To review scenarios for alert types and triggering, see Alert type and triggering scenarios.

Using cron expressions

You can use a cron expression to customize alert scheduling. See Use cron expressions for scheduling to learn more.

Create a scheduled alert

Prerequisites

- Use cron expressions for scheduling
- Alert scheduling tips
- Configure alert trigger conditions
- Monitor triggered alerts

Steps

1. Navigate to the **Search** page in the **Search and Reporting** app.
2. Create a search.
3. Select **Save As>Alert**.
4. Enter a title and optional description.
5. Specify permissions.
6. Configure alert scheduling. There are two options for scheduling.

Option	Next steps for this option
Select one of the available scheduling options and set a time.	None.
For further customization, select **Run on Cron Schedule**	1. Enter the **Earliest** and **Latest** values for the search time range. These values override the original search time range.

Option	Next steps for this option
to use a time range and cron expression.	To avoid overlaps or gaps, the execution schedule should match the search time range. For example, to run a search every 20 minutes the search time range should also be 20 minutes (-20m). 2. Enter a cron expression to schedule the search. See cron expression examples here: Use cron expressions for scheduling.

7. (Optional) Change the **Expires** setting. This setting controls the lifespan of triggered alert records, which appear on the **Triggered Alerts** page.
8. Configure trigger conditions.
9. (Optional) Configure a trigger throttling period.
10. Select one or more alert actions that should happen when the alert triggers.
11. Click **Save**.

Additional resources

- Review scheduled alert best practices in Alert scheduling tips.
- See also Alert examples.

Use cron expressions for alert scheduling

You can customize alert scheduling using a time range and cron expression.

Cron expression syntax

A cron expression is a data string of five fields separated by spaces.

From left to right, the five cron fields have the following chronological value ranges:

- Minute: 0-59
- Hour: 0-23
- Day of the month: 1-31
- Month: 1-12
- Day of the week: 0-6 (where 0 = Sunday)

Commonly used cron field formats

The following cron field formats suit most use cases.

Format	Description	Explanation of description	Hour field example	Example meaning
N	One value	Only this value	9	9:00 AM
N,M	Multiple comma-separated values	Only the listed values	9,15	9:00 AM and 3:00 PM
I-J	Value range, inclusive	All values in this range, including the range start and end values	9-17	9:00 AM through 5:00 PM
*	Asterisk (indicates "all values")	Each value in this field	*	Every hour
*/N	Every N value in this field	All values in this field are intervals of N	*/3	Every 3 hours 0, 3, 6, 9, 12, 15, 18, 21

Cron field formats for ranges and intervals

In some cases, you might want to use multiple value ranges or combine ranges and an interval in a cron expression. The following format options are available.

Format	Description	Meaning	Hour field example	Example meaning
I-J,K-L	Multiple comma-separated value ranges	All values in each of these ranges, including the range start and end values	9-12,15-17	9:00 AM through 12:00 PM and 3:00 PM through 5:00 PM
I-J/N	Range and /N interval	Each value in this field that is an interval of /N and is within this	9-12/2	9:00 AM and 11:00 AM

Format	Description	Meaning	Hour field example	Example meaning
		range		
`I-J,K-L/N`	Multiple comma-separated ranges and `/N` interval	Each value in this field that is an interval of `/N` and is within the specified ranges	`9-12,15-17/2`	9:00 AM 11:00 AM and 3:00 PM 5:00 PM

Working with cron intervals

In cron expressions with an interval of `/N`, all values in the specified range that are intervals of `N` are used. If a number in the range is outside of the interval `N`, the value resets to 0.

For example, `*/9 * * * *` means "every nine minutes". The following minute field values are used:

```
0, 9, 18, 27, 36, 45, 54
```

After `54`, the value resets to `0`.

Cron value ranges and intervals

When using a `I-J/N` range and interval format, the interval `N` is applied to the first number in the range.

For example, `13-36/10` in the minute field results in the following values used:

```
13, 23, 33
```

Example expressions

Here are some example cron expressions.

```
*/5 * * * *        Every 5 minutes.
*/30 * * * *       Every 30 minutes.
0 */12 * * *       Every 12 hours, on the hour.
*/20 * * * 1-5     Every 20 minutes, Monday through Friday.
0 9 1-7 * *        The first 7 days of every month at 9 AM.
```

Alert scheduling tips

This topic presents best practices and suggestions for working with scheduled alerts.

Best practices

Coordinate an alert schedule and search time range

Coordinating an alert schedule with the search time range prevents event data from being evaluated twice by the search. If search time range exceeds the search schedule, event data sets can overlap.

When a search time range is shorter than the time range for the scheduled alert, an event might never be evaluated.

Schedule alerts with at least one minute of delay

This practice is important in distributed search deployments where event data might not reach the indexer immediately. A delay ensures that you are counting all events, not just the events that were indexed first.

Best practices example

This example shows how to configure an alert that builds 30 minutes of delay into the alert schedule. Both the search time range and the alert schedule span one hour, so there are no event data overlaps or gaps.

1. From the Search Page, create a search and select **Save As > Alert**.
2. In the **Save As Alert** dialog, specify the following options as shown.
    - **Title**: Alert Example (30 Minute Delay)
    - **Alert Type**: Scheduled
    - **Time Range**: Run on Cron Schedule
    - **Earliest**: -90m
    - **Latest**: -30m
    - **Cron Expression**: 30 * * * *
3. Continue defining actions for the alert.

Earliest and **Latest** values set the search time range from 90 minutes before the search launches to 30 minutes before the search launches. The alert runs runs hourly at 30 minutes past the hour. It collects event data from a one hour period. When the scheduled search begins at a designated time, such as 3:30 p.m., it collects the event data indexed from 2:00 pm to 3:00 pm.

Manage concurrent scheduled search priority

Depending on your deployment, you might be able to run only one scheduled search at a time. In this case, even if you schedule multiple searches to run at the same time, the search scheduler ensures that scheduled searches run consecutively.

You might need to change scheduled search priority to ensure that a search obtains current data or to prevent gaps in data collection. If you have Splunk Enterprise, you can configure scheduled search priority by editing the `savedsearches.conf` configuration file. See Configure the priority of scheduled reports in the *Reporting Manual* for more information.

Differences between scheduled reports and alerts

A scheduled report is like a scheduled or real-time alert in certain ways. You can schedule a report and set up an action to run each time the scheduled report runs.

Scheduled reports are different from alerts, however, because a scheduled report's action runs every time the report is run. The report action does not depend on trigger conditions.

As an example, you can monitor guest check-ins at a hotel using an hourly search. Here are the differences between a scheduled report and a scheduled alert with email notification actions.

- **Scheduled report**: runs its action and sends an email every time the report completes, even if there are no search results showing check-ins. In this case, you get an email notification every hour.
- **Scheduled alert**: only runs alert action when it is triggered by search results showing one or more check-in events. In this case, you only get an email notification if results trigger the alert action.

For more information, see Schedule reports in the *Reporting Manual*.

Create real-time alerts

Use a real-time alert to monitor events or event patterns as they happen. You can create real-time alerts with per-result triggering or rolling time window triggering. Real-time alerts can be costly in terms of computing resources, so consider using a scheduled alert when possible.

To compare scheduled and real-time alerts, see Alert types. To review scenarios for alert types and triggering, see Alert type and triggering scenarios.

Create a real-time alert with per-result triggering

Real-time alerts with per-result triggering are sometimes known as per-result alerts. This alert type and triggering use a continuous real-time search to look for events. Each search result triggers the alert.

> **Caution:** If you have a Splunk Enterprise high-availability deployment, use per-result triggering with caution. If a peer is not available, a real-time search does not warn that the search might be incomplete. To avoid this issue, use a scheduled alert

Follow these steps to create a real-time alert with per-result triggering.

1. Navigate to the **Search** page in the Search & Reporting app.
2. Create a search.
3. Select **Save As**>**Alert**.
4. Enter a title and optional description.
5. Specify permissions.
6. Select the **Real-time** alert type.
7. (Optional) Change the **Expires** setting. This setting controls the lifespan of triggered alert records, which appear on the Triggered Alerts page.
8. Select the **Per-Result** trigger option.
9. (Optional) Configure a trigger throttling period.
10. Select at least one alert action that occurs when the alert triggers.
11. Click **Save**.

Create a real-time alert with rolling window triggering

Real-time alerts with rolling time window triggering are sometimes known as rolling window alerts. The rolling time window is an interval or increment, such as five minutes. It is not a scheduled time. Because real-time alerts search continuously, the time window applied to events also rolls forward in time.

Use this alert type and triggering when a specific time interval is part of the event pattern you are monitoring in real time. This alert type and triggering are the most resource-demanding alerting option. It can be helpful to consider using another alert type if possible.

Follow these steps to create a real-time alert with rolling window triggering.

1. Navigate to the **Search** page in the Search & Reporting app.
2. Create a search.
3. Select **Save As**>**Alert**.
4. Enter a title and an optional description.
5. Specify permissions.
6. Select the **Real-time** alert type.
7. (Optional) Change the **Expires** setting. This setting controls the lifespan of triggered alert records, which appear on the Triggered Alerts page.
8. Select one of the available result-based conditions, or enter a custom triggering condition. Do not select per-result triggering.
9. Specify a time interval to add to the triggering condition.
10. (Optional) Configure a trigger throttling period.
11. Select at least one alert action that occurs when the alert triggers.
12. Click **Save**.

Additional resources

- Learn about alert and alert action permissions in Alert permissions.
- Step through alert examples in Alert examples.
- Learn more about using trigger conditions in Configure alert trigger conditions.
- Learn about using the Triggered Alerts page to review triggered alert records in Monitor triggered alerts

Manage alert trigger conditions and throttling

Configure alert trigger conditions

An alert can search for events on a schedule or in real time, but it does not have to trigger every time search results appear. Trigger conditions help you monitor patterns in event data or prioritize certain events.

Alert triggering and alert throttling
Throttling an alert is different from configuring trigger conditions. When you create trigger conditions, search results are evaluated to check if they match the conditions. If results match the trigger conditions, throttling controls whether triggering is suppressed for a period of time. For more information on throttling, see Throttle alerts.

Workflow for trigger configuration

When configuring alert triggering, it is helpful to consider the following questions.

What event pattern is the alert monitoring?
Trigger conditions evaluate the alert's search results for a particular pattern. This pattern combines result fields and their behavior. For example, you can select one of the built-in field count options, such as **Number of Hosts**, to focus on the `host` field. You can then specify the behavior to monitor, such as when that number drops by five. You can also enter a custom triggering condition.

Does the pattern trigger the alert once or for every result?
When the event pattern happens, the alert can trigger just once or one time for each result in the pattern. You can choose an option depending on the notification or other alert action behavior that you want.

Alert types and triggering options

Both alert types offer trigger configuration options for working with the alert search results. Here is a comparison of available triggering options for each type.

Alert type	Trigger options	Specifying trigger conditions	How matching results trigger the alert
Scheduled	Add trigger conditions to evaluate search results.	Built-in result and field count options or a custom triggering condition	Trigger the alert once each time search results match the specified condition or one time for every matching result.
Real-time	Per-result	N/A	By default, alert triggers one time for every matching result.
Real-time	Trigger conditions that include a rolling time window.	Built-in result and field count options or a custom condition. Also specify a rolling time window or interval.	Trigger the alert once each time search results match the specified condition, or one time for every matching result.

How searches and trigger conditions work together

Trigger conditions work as a secondary search to evaluate the alert's initial search results. If the secondary search does not return results, the alert does not trigger. When the secondary search does generate results, the alert triggers.

Depending on the alert actions you choose, you can access information about results that trigger the alert. The secondary search for trigger conditions does not determine what information is available for notifications or other alert actions. Result fields and other information come from the initial base search.

Using the alert base search without trigger conditions can limit the information available for notifications. The following example compares using a base search with a custom triggering condition and using a base search without trigger conditions.

Example

This scheduled alert triggers when there are ten or more urgent `log_level` events. When the alert triggers, it sends an email with the search results.

Using a search with custom trigger condition

The alert uses this search, with **Last 7 days** selected in the time range picker.

```
index=_internal (log_level=ERROR OR log_level=FATAL OR
log_level=CRITICAL) | stats count by log_level
```

The following custom triggering condition is added.

```
search count > 10
```

In this scenario, the original search results detail the count for all log levels, but the alert triggers only when the log_level counts are greater than ten. This means that all `log_level` counts are available to use as part of an alert notification.

Using a search without a trigger condition

The following search looks similar to the previous example. It generates similar alert triggering behavior. However, it creates different results and limits the `log_level` information available to notifications or other alert actions.

```
log_level=ERROR OR log_level=FATAL OR log_level=CRITICAL) | stats count
by log_level | search count > 10
```

In this case, the search results include only `log_level` values that are greater than ten. By comparison, using a search with conditional triggering in the previous example means that results include counts for all `log level` fields.

Throttle alerts

Use throttling to suppress alert triggering for a specific time period. Alerts can trigger frequently because of similar search results or scheduling.

Throttling an alert is different from configuring alert trigger conditions. Trigger conditions evaluate an alert's initial search results to check for specified field counts, event timing, or other patterns. To review alert triggering information, see Configuring alert trigger conditions.

Throttle configuration and scenarios

When creating or editing an alert, you can enable and configure alert throttling, also known as suppression.

Alert type	Triggering option	How to configure throttling
Scheduled	Once	Indicate a suppression period using the time value field and dropdown increments. Time values must be

Alert type	Triggering option	How to configure throttling
		greater than zero.
Scheduled	Per-result	1. Type one or more comma-separated fields to check for matching values in events. Events with the same value for these fields are suppressed. As an example, you might configure suppression on the field `product_category` and the field `best_sellers`. After an alert on one event where both the `product_category` field **AND** the `best_sellers` have value `arcade`, subsequent events with the `arcade` value in the `product_category` and the `best_sellers` field are suppressed during the throttling time period. 2. Indicate a suppression period using the time value field and dropdown increments. Time values must be greater than zero.
Real-time	Rolling time window	Indicate a suppression period using the time value field and dropdown increments. Time values must be greater than zero.
Real-time	Per-result	1. Type one or more comma-separated fields to check for matching values in events. Events with the same value for these fields are suppressed. As an example, you might configure suppression on a `product_category` field. After an alert on one event with the `product_category` value `arcade`, subsequent events with the `arcade` value in the `product_category` field are suppressed during the throttling time period. 2. Indicate a suppression period using the time value field and dropdown increments. Time values must be greater than zero.

Throttling scenarios

- An admin uses a real-time alert with per-result triggering to monitor system events, including errors. System events occur twenty or more times per minute. For notification purposes, alert triggers can be suppressed for an hour. The admin uses field values and a one hour suppression period to throttle the events.
- A real-time alert with per-result triggering monitors disk errors. Some events in the alert's search results have the same `clientip` or `host` values but can cause multiple alert triggers in a short amount of time. An admin throttles the alert so that, after an initial alert triggers, subsequent triggering is suppressed for ten minutes.
- A scheduled alert searches for sales events on an hourly basis. The alert triggers whenever the number of results rises by 100 and is configured to send an email notification to the sales team. The sales team wants to limit email notifications. An admin throttles the alert so that triggering is suppressed for three hours after an initial alert triggers and initializes an email notification.

Throttle scheduled and real-time searches

Throttling for alerting works similarly to throttling for scheduled and real-time searches.

If you have scheduled searches that run frequently and you do not want to be notified each time results generate, set the throttling controls to suppress the alert for a longer time period.

For real-time searches, if you configure an alert so that it triggers once when a specific triggering condition is met, you do not need to configure throttling. If the alert triggers for each result, you might need to configure throttling to suppress additional alerts.

When you configure throttling for a real-time search, start with a throttling period that matches the length of the base search time range. Expand the throttling period if necessary. This prevents multiple notifications for a given event.

Configure alert actions

Set up alert actions

Alert actions help you respond to triggered alerts. You can enable one or more alert actions. Learn about the available options.

To learn about	See
Sending email notifications when alerts trigger	Email notification action
Displaying a message in a chat room or updating another web resource	Use a webhook alert action
Writing the results of the triggered alert or scheduled report to a CSV lookup file	Output results to a CSV lookup
Logging and indexing searchable alert events	Log events
Adding an alert to a list of recently triggered alerts for monitoring	Monitor triggered alerts

The script alert action is deprecated. As an alternative you can define customized actions that can include scripts.

See About custom alert actions.

Email notification action

Send an email notification to specified recipients when an alert triggers. Email notifications can include information from search results, the search job, and alert triggering. You can set up an email notification action from the **Search** page, the **Alerts** page, or directly in a search command.

In addition to alerting, there are other email notification contexts. For information on email notifications for reports, see Schedule reports in the *Reporting Manual*. For information on dashboard PDF email delivery, see Generate Dashboard PDFs in *Dashboards and Visualizations*.

Configure email notification for your Splunk instance

You must configure email notification settings for your Splunk implementation before you can design an email notification action.

If your email notification settings are already configured you can skip this task.

Prerequisite

PDF delivery requires additional user role configuration. See "User role configuration for PDF delivery" at the bottom of this page.

Steps

1. From the **Search and Reporting** app home page, select **Settings** > **Server settings** > **Email settings**.
2. Select **Mail Server Settings**.
3. Specify values for the following settings.

Setting	Definition
Mail host	The default value is **localhost**.
Email security	Select one of the available options.
Username	Optional. Required for SMTP server authentication.
Password	Optional. Required for SMTP server authentication.

4. Specify **Email Format** settings.

Email Format Setting	Definition
Link hostname	The hostname for outgoing results URLs. Enclose IPv6 addresses in square brackets. Example: **[2001:db8:0:1]**
Send emails as	(Optional) Specify a sender identification, used in the **From** email header field. Use an email address or a string. Strings are concatenated with **@<hostname>**, using the hostname specified in `alert_actions.conf`for the machine sending the email notification or **@localhost** if no hostname is specified. Defaults to **splunk@<hostname>** or **splunk@localhost** if no hostname is specified.
Email footer	Footer for all emails. Use text and/or tokens.

5. Click **Save**.

Define an email notification for an alert or scheduled report

- Before you can send an email notification, configure the email notification settings in the **Settings** page. See Configure email notification for your Splunk instance.
- To send an email notification within a search to a mail server that requires SMTP authentication, you must have the `admin` role assigned.
- To send an email notification within a search to a mail server that does not require SMTP auth requires the `list_settings` capability. By default, only the `admin`, `splunk-system-role`, and `can_delete` roles are assigned the `list_settings` capability.
  If you want to allow users not belonging to any of these roles to send email notifications using the `sendemail` command in their search, you must assign them the `list_settings` capability. For more information on roles and capabilities, see "About defining roles with capabilities" in the Securing Splunk Enterprise Manual.
- PDF delivery requires additional user role configuration. See "User role configuration for PDF delivery" at the bottom of this page.
- To review token usage, see Use tokens in email notifications.

Steps

1. You can configure the email notification action when you create a new alert, edit the actions for an existing alert, or define or edit the schedule for a report. Follow one of the options below.

Option	Steps
Create a new alert	From the **Search** page in the **Search and Reporting** app, select **Save As > Alert**. Enter alert details and configure triggering and throttling as needed.
Edit an existing alert	From the **Alerts** page in the **Search and Reporting** app, select **Edit > Edit actions** for an existing alert.
Define or edit the schedule of a report	From the **Reports** page in the **Search and Reporting** app, select **Edit > Edit schedule** for a report.

2. Click **Add Actions** and select **Send email**.
3. Type a comma-separated list of **To** email recipients.
4. (Optional) Click **Show CC and BCC** to type comma-separated lists of CC, and BCC email recipients.

5. (Optional) Set the email **Priority**. Enforcement of email priority depends on your email client.
6. (Optional) Provide the email **Subject** and **Message**.
   You can optionally use tokens in the subject and message text.
7. (Optional) Select one or more of the following options to include material in the email.

Option	Adds to email
Link to Alert or **Link to Report**	A link to the alert or scheduled report that the email is associated with.
Link to Results	A link to the results for the related search job.
Search String	The search string used by the alert or scheduled report.
Inline...	Displays the results as an inline table, a list of raw events, or in CSV file format.
Trigger condition (for alerts only)	The condition that triggered the alert.
Trigger time (for alerts only)	The alert timestamp.
Attach CSV	A file attachment that provides the results in CSV format.
Attach PDF	A file attachment that provides the results in PDF format.

8. (Optional) Change the email **Type** to **Plain Text**.
   Type is set to **HTML & Plain Text** by default.
9. Click **Save**.

If you have Splunk Enterprise, you can configure email alert settings by editing the `alert_actions.conf` configuration file. For details, see alert_actions.conf.

Use a search command to send an email notification

You can send email notifications directly from the `sendemail` search command. Here is an example.

```
index=main | head 5 | sendemail to=<email address> server=<server info>
subject="Here is an email notification" message="This is an example
message" sendresults=true inline=true format=raw sendpdf=true
```

If you are sending an email notification to a server that requires SMTP authentication, you must have the admin role assigned.

See the sendemail command listing in the *Search Reference* for more details.

Example - Send email to different recipients based on search results

This example shows you how you can use the `$result.recipient$` token to make the Splunk software send notification emails to different recipients depending on the number of results returned by the search.

The `$result.recipient$` token works in conjunction with an `eval` statement in the search. This `eval` statement sets the conditions under which emails are sent to specific addresses.

Here is an example of a search that is designed to work with `$result.recipient$`.

```
"error" | stats count | eval recipient=case(count > 3500,
"recipient1@domain.com", count >= 500, "recipient2@domain.com", 1==1,
null()) | where isnotnull(recipient)
```

After this search is saved as an alert or scheduled report, you design an email notification action for it where you type `$result.recipient$` in the **To** field.

When the alert is triggered or the scheduled report runs on its schedule, a notification is sent `recipient1` if there are more than 3500 results. If there are fewer than 500 results, a notification is sent to `recipient2`. If neither condition applies, no notification is sent.

User role configuration for PDF delivery

The following capabilities are required for PDF delivery scheduling.

- schedule_search
- admin_all_objects. This capability is required if the mail host requires login credentials.
- list_settings

See About defining roles with capabilities in the *Security Manual* for more information.

Use tokens in email notifications

Tokens represent data that a search generates. They work as placeholders or variables for data values that populate when the search completes.

You can use tokens in the following fields of an email notification.

- To
- Cc
- Bcc
- Subject
- Message
- Footer

If you have Splunk Enterprise, you can change footer text by editing `alert_actions.conf`.

Use this token syntax to reference values from the search: `$<token>$`

For example, place the following text and token in the subject field of an email notification to reference the search ID of a search job:

```
Search results from $job.sid$
```

Tokens available for email notification

There are four categories of tokens that access data generated by searches. Token availability varies by context.

Category	Context: Alert Actions	Context: Scheduled Reports	Context: Scheduled PDF delivery
Search metadata	Yes	Yes	Yes
Search results	Yes	Yes	No
Job information	Yes	Yes	No
Server information	Yes	Yes	Yes
Dashboard information	No	No	Yes

If you have Splunk Enterprise, you can use tokens to access values for attributes listed in `savedsearches.conf` and `alert_actions.conf`. Use the attribute name with standard token syntax. For example, to access an email notification subject, use `$action.email.subject$`.

Search metadata tokens

Common tokens that access information about a search.

Token	Description
$action.email.hostname$	Email server hostname
$action.email.priority$	Search priority
$alert.expires$	Alert expiration time
$alert.severity$	Alert severity level
app	App context for the search
$cron_schedule$	Search cron schedule
$description$	Human-readable search description
$name$	Search name
$next_scheduled_time$	The next time the search runs
$owner$	Search owner
$results_link$	(Alert actions and scheduled reports only) Link to search results
$search$	Search string
$trigger_date$	(Alert actions only) Date when alert triggered, formatted as `Month(string) Day, Year`
$trigger_time$	(Alert actions only) Time when alert triggered, formatted as epoch time
$type$	Indicates if the search is from an alert, report, view, or the search command
$view_link$	Link to view saved search

Result tokens

You can access field values from the first result row that a search returns. Field availability for tokens depends on what fields are available in search results.

Token	Description
$result.*fieldname*$	First value for the specified field name from the first search result row. Verify that the search generates the field being accessed.

To include or exclude specific fields from the results, use the `fields` command in the base search for the alert. For more information, see fields in the *Search Reference*.

Job information tokens

Common tokens that access data specific to a search job, such as the search ID or messages generated by the search job.

Token	Description
$job.earliestTime$	Initial job start time
$job.eventSearch$	Subset of the search that appears before any transforming commands
$job.latestTime$	Latest time recorded for the search job
$job.messages$	List of error and debug messages generated by the search job
$job.resultCount$	Search job result count
$job.runDuration$	Time, in seconds, for search job completion
$job.sid$	Search ID
$job.label$	Search job name

Server tokens

Provide details about your Splunk deployment.

Token	Description
$server.build$	Build number of the Splunk deployment.
$server.serverName$	Server name hosting the Splunk deployment.
$server.version$	Version number of the Splunk deployment.

Dashboard metadata tokens

Access dashboard metadata and include it in dashboard delivery emails.

Token	Description
$dashboard.label$	Dashboard label
$dashboard.title$	Equivalent to `$dashboard.label$`
$dashboard.description$	Dashboard description
$dashboard.id$	Dashboard ID

Deprecated email notification tokens

The following tokens are deprecated.

Token	Alternative option
$results.count$	(Deprecated) Use $job.resultCount$.
$results.file$	(Deprecated) No equivalent available.
$results.url$	(Deprecated) Use $results_link$.
$search_id$	(Deprecated) Use $job.sid$.

Use a webhook alert action

Webhooks allow you to define custom callbacks on a particular web resource. For instance, you can set up a webhook to make an alert message pop up in a chat room or post a notification on a web page. When an alert triggers, the webhook makes an HTTP POST request on the URL. The webhook passes JSON formatted information about the alert in the body of the POST request.

Webhook data payload

The webhook POST request's JSON data payload includes the following details.

- Search ID or SID for the saved search that triggered the alert
- Link to search results
- Search owner and app
- First result row from the triggering search results

Example

```
{

        "result": {
                "sourcetype" : "mongod",
                "count" : "8"
        },
        "sid" : "scheduler_admin_search_W2_at_14232356_132",
        "results_link" :
"http://web.example.local:8000/app/search/@go?sid=scheduler_admin_search_W2_at_14232356
        "search_name" : null,
        "owner" : "admin",
        "app" : "search"
}
```

Depending on the webhook scenario, you can configure data payload handling on the resource receiving the POST.

Configure a webhook alert action

Set up a webhook when selecting alert actions for an alert.

1. You can configure the webhook action when creating a new alert or editing an existing alert's actions. Follow one of the options below.

Option	Steps
Create a new alert	From the **Search** page in the **Search and Reporting** app, select **Save As > Alert**. Enter alert details and configure triggering and throttling as needed.
Edit an existing alert	From the **Alerts** page in the **Search and Reporting** app, select **Edit**>**Edit actions** for an existing alert.

2. From the **Add Actions** menu, select **Webhook**.
3. Type a URL for the webhook.
4. Click **Save**.

Output results to a CSV lookup

This action writes the results of a triggered alert or a run of a scheduled report to a CSV **lookup** file that you specify. The results can replace the existing file contents, or they can be appended to the existing file contents.

The Splunk software uses the `outputlookup` command to write the search results to the CSV lookup file.

Prerequisites

- Learn how to upload CSV lookup files and create CSV lookup definitions. See Define a CSV Lookup in Splunk Web in the *Knowledge Manager Manual*.

Steps

1. You can configure the output results to lookup action when you create a new alert, edit the actions for an existing alert, or define or edit the schedule for a report. Follow one of the options below.

Option	Steps
Create a new alert	From the **Search** page in the **Search and Reporting** app, select **Save As > Alert**. Enter alert details and configure triggering and throttling as needed.
Edit an existing alert	From the **Alerts** page in the **Search and Reporting** app, select **Edit > Edit Alert** for an existing alert.
Define or edit the schedule of a report	From the **Reports** page in the **Search and Reporting** app, select **Edit > Edit schedule** for a report.

2. Click **Add Actions** and select **Output results to lookup**.
3. Provide a **File name** of a CSV lookup file. You can provide the name of a CSV lookup file that has already been uploaded to your Splunk implementation, or you can provide a CSV lookup file name that is not currently uploaded.
   If you provide a CSV lookup file name that has not been uploaded to your Splunk implementation, the Splunk platform creates a CSV file with the file name you provide. The Splunk platform then populates the new CSV file with the results of that first triggering search job.
   To see a list of the CSV lookup files currently uploaded to your Splunk implementation, select **Settings > Lookups > Lookup table files**.
4. Determine how you would like to have the **Results** written to the CSV lookup file.

Option	Description
Append	Append the results returned by a run of the search to the contents of the CSV file. This is the default setting.
Replace	Replace the contents of the CSV file with the results returned by a run of the search.

5. Click **Save**.

Log events

Construct custom log events to index and search metadata. Log events are sent to your Splunk deployment for indexing. As with other alert actions, log events can be used alone or in addition to other alert actions for a given alert.

Authorization requirement

Using the log event alert action requires the `edit_tcp` capability for users without the `admin` role.

Tokens for log events

When you set up a log event alert action, populate event fields with plain text or tokens representing search, job, or server metadata. You can also use tokens to access the first search results set.

Tokens available for email notifications are also available for log events. For more information on using tokens with alert actions, see Use tokens in email notifications in this manual.

Set up a log event alert action

Here are the steps for setting up a custom log event alert action after building a search.

Prerequisites
To review token usage, see Use tokens in email notifications in this manual.

Steps

1. You can configure the log event action when ceating a new alert or editing an existing alert's actions. Follow one of the options below.

Option	Steps
Create a new alert	From the **Search** page in the **Search and Reporting** app, select **Save As > Alert**. Enter alert details and configure triggering and throttling as needed.
Edit an existing alert	From the **Alerts** page in the **Search and Reporting** app, select **Edit**>**Edit actions** for an existing alert.

Option	Steps

The following steps are the same for saving new alerts or editing existing alerts.

2. From the **Add Actions** menu, select **Log event**.
3. Add the following event information to configure the alert action. Use plain text or tokens for search, job, or server metadata.
    - **Event text**
    - **Source and sourcetype**
    - **Host**
    - **Destination index for the log event.** The `main` index is the default destination. You can specify a different existing index.
4. Click **Save**.

Monitor triggered alerts

Add an alert to a list of triggered alerts. Review triggered alerts by app context, owner, and severity level.

Add an alert to the Triggered Alerts list

1. Use one of the following options depending on whether you are creating a new alert or editing an existing alert.

Option	Steps
Create a new alert	From the **Search** page in the **Search and Reporting** app, select **Save As > Alert**. Enter alert details and configure triggering and throttling as needed.
Edit an existing alert	From the **Alerts** page in the **Search and Reporting** app, select **Edit**>**Edit actions** for an existing alert.

2. From the **Add Actions** menu, select **Add to triggered alerts**.
3. Select an alert **Severity** level.
   Severity levels are informational only. They are used to group alerts in the **Triggered Alerts** list. The default level is Medium.
4. Click **Save**.

Reviewing recently triggered alerts

You can see records of recently triggered alerts from the Triggered Alerts page or from an Alert Details page. The Triggered Alerts page shows all instances of triggered alerts. See Review triggered alerts for more information on viewing and interpreting triggered alerts.

Records of triggered alert details are available for twenty-four hours by default. See Update triggered alert lifespans for information on changing the expiration setting for an individual alert.

Run a script alert action

The run a script alert action is officially deprecated. It has been replaced with custom alert actions as a more scalable and robust framework for integrating custom actions. See About custom alert actions for implementation and migration information.

If you have Splunk Enterprise, you can run an alert script when an alert triggers. Select **Run a script** from the **Add Actions** menu. Enter the file name of the script that you want to run.

For example, you can configure an alert to run a script that generates a Simple Network Management Protocol (SNMP) trap notification. The script sends the notification to another system such as a Network Systems Management console. You can configure a different alert that runs a script that calls an API, which in turn sends the triggering event to another system.

> **Note:** For security reasons, place all alert scripts in either of the following locations:
>
> ◊ `$SPLUNK_HOME/bin/scripts`
> ◊ `$SPLUNK_HOME/etc/<AppName>/bin/scripts`

For details on alert script configuration in `savedsearches.conf` for a shell script or batch file that you create, see Configure scripted alerts in this manual.

Custom alert actions

Using custom alert actions

App developers can build custom, user-configurable, alert actions into their apps. Users can find apps with built-in custom alert actions from the alert actions manager page.

To try using a custom alert action, you can use the built-in webhook alert action to send notifications to a web resource, like a chat room or blog. For more information, see Use a webhook alert action.

To learn how to find apps with built-in alert actions, see Using the alert actions manager.

For developer information, see the following topics in *Developing Views and Apps for Splunk Web*.

- Custom alert actions overview
- Convert a script alert action to a custom alert action

Manage alert and alert action permissions

Alert permissions

Alerts are knowledge objects with defined permissions. User roles and capabilities determine alert creation, usage, editing, and other permissions.

By default, only users with the Admin or Power roles can do the following.

- Create alerts.
- Run real-time searches.
- Schedule searches.
- Save searches.
- Share alerts.

Authorized users can share an alert with other app users by editing the alert permissions. When sharing an alert with a user without the Admin or Power role, the user needs permission to access the alerting features. For example, a user needs the capability to run a real-time search in order to access a real-time alert.

Admins can configure alert action permissions to change what alert actions are available to users in a particular app. For more information, see Alert Action Permissions.

Alerts can only run with the permissions of their owner, unlike unscheduled reports, which can run with the permissions of either their owner or their user.

See Determine whether to run reports as the report owner or report user in the *Reporting Manual*.

Sharing an alert

You can configure sharing preferences when creating an alert or edit alert permissions later. Here are the steps for editing alert permissions.

1. Navigate to the **Alerts** page in the **Search and Reporting** app.
2. Find the alert you want to share and select **Edit** > **Edit Permissions**.
3. Share the alert by configuring which users can access it. Here are the options.

Option	Sharing description
Owner	Makes the alert private to the alert creator.
App	Display the alert for all users of the app.
All apps	Display the alert for all users of this Splunk deployment.

4. Select read and write permissions for the user roles listed.
    - **Read**: Users can see the alert on the **Alerts** page and run the alert in the app.
    - **Write**: Users with appropriate permissions can modify, enable, and disable the alert.

Alert action permissions

Depending on the user **roles** that you have, you can configure alert action permissions for available alerts.

For example, an admin can adjust alert actions permissions for the **Search and Reporting** app. The admin can change what alert actions are available to users who create an alert in this app.

To review and change alert actions permissions, use the **Alert actions** manager page. For more information, see Using the alert actions manager.

Alert actions are **knowledge objects**. To learn more about managing knowledge object permissions, see Manage knowledge object permissions in the *Knowledge Manager* manual.

View and update alerts

Access and update alerts

There are several ways to access and edit alerts. Here is a comparison of typical alert management tasks and where to complete them in Splunk Web.

Task	Where to go
View all alerts in the current app context.	**Alerts** page
Select an alert to review or update.	**Alerts** page
View and edit alert details.	From the **Alerts** page, select an alert to open its detail page.
Review available alert actions and browse for more actions.	**Alert Actions** manager page.
Review recently triggered alerts.	**Triggered Alerts** listing page.

Use the Alerts page

The **Alerts** page lists all alerts for an app. It is available from the top-level navigation menu for an app. From the **Alerts** page you can use the following options.

Option	Description
Select a filtering option for displayed alerts.	• **All**. View all alerts for which you have view permission. • **Yours**. View alerts that you own. • **This App's**. View alerts for the current app. Only alerts for which you have permission to view display in the list.
Select any displayed alert	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.
Open in Search	View or modify the alert's search string in the **Search** page. Time range updates in Splunk Web are not supported.

Option	Description
Edit	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.

Edit an alert search

1. From the **Alerts page**, locate the alert and click **Open in Search**. The alert search opens in the **Search** page.
2. Edit the search string as needed.
3. Run the edited search.
4. Click **Save** to update the alert. If prompted again, click **Save**.
5. Select from the following options.

Option	Description
"View alert"	Opens the alert detail page.
"Continue editing"	Return to the **Search** page.
"Permissions"	View and edit alert permissions.

Access alert details

From the **Alerts** page, select an alert to review and update its settings. Authorized users can change the following alert settings.

- Enable or disable the alert
- App context
- Permissions
- Alert type and timing
- Trigger conditions
- Alert actions

Alerts page

The **Alerts** page lists all alerts for an app. It is available from the top-level navigation menu for an app. From the **Alerts** page you can use the following options.

Option	Description

Select a filtering option for displayed alerts.	• **All**. View all alerts for which you have view permission. • **Yours**. View alerts that you own. • **This App's**. View alerts for the current app. Only alerts for which you have permission to view display in the list.
Select any displayed alert	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.
Open in Search	View or modify the alert's search in the **Search** page.
Edit	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.

Edit an alert search

1. From the **Alerts page**, locate the alert and click **Open in Search**. The alert search opens in the **Search** page.
2. Edit the search as needed.
3. Run the edited search.
4. Click **Save** to update the alert. If prompted again, click **Save**.
5. Select from the following options.

Option	Description
"View alert"	Opens the alert detail page.
"Continue editing"	Return to the **Search** page.
"Permissions"	View and edit alert permissions.

Using the alert actions manager

You can review and configure settings for available alert actions on the alert actions manager page.

Prerequisites
(Optional) Review Alert action permissions.

Steps

1. From the top-level navigation bar, select **Settings**> **Alert actions**.
2. Depending on your permissions, you can do the following for an alert action.
    - ♦ Enable or disable the alert action
    - ♦ Update permissions
    - ♦ Check usage stats
    - ♦ View log events
3. (Optional) Click **Browse More** to find apps with built-in custom alert actions.

Triggered alerts

Review all recently triggered alerts on the **Triggered Alerts** page.

For information on configuring the "Add to Triggered Alerts" action, see Monitor triggered alerts.

Triggered alert listing

Alerts appear on the **Triggered Alerts** page under the following conditions.

- The "Add to Triggered Alerts" action is enabled for the alert.
- The alert triggered recently.
- The alert retention time is not complete.
- The triggered alert listing has not been deleted.

On the **Triggered Alerts** page, details appear in the following categories.

Category	Description
Time	Trigger date and time.
Fired alerts	Triggered alert name(s).
App	Alert app context.
Type	Alert type.
Severity	Assigned alert severity level. Severity levels can help you sort or filter alerts on this page.
Mode	Alert triggering configuration mode. "Per-result" means that the alert triggered because of a single event. "Digest" means that the alert triggered because of a group of events.

Category	Description

Records of triggered alerts are available for twenty-four hours by default. You can configure this expiration time on a per-alert basis. For example, you can arrange to have the triggered alert records for an alert have a lifespan of seven days instead of twenty-four hours. See Update triggered alert lifespans for information on changing the lifespan of the alert records for an individual alert.

Access and update triggered alerts

Here are steps for accessing and using the **Triggered Alerts** page.

Prerequisites
(Optional) Review Triggered alert listing.

Steps

1. From the top-level navigation bar, select **Activity** > **Triggered Alerts**.
2. Filter any displayed alerts according to **App**, **Owner**, **Severity**, and **Alert** (alert name).
3. (Optional) Use the keyword search to find triggered alerts by alert name or app context.
4. (Optional) Take the following actions from the Alert Manager.

   ◊ View alert search results.
   ◊ Edit the alert search.
   ◊ Delete a triggered alert listing.

Delete a triggered alert listing

By default, triggered alert records on the **Triggered Alerts** page expire after twenty-four hours. There are a few ways to change whether a triggered alert listing appears on this page.

- Update triggered alert listing expiration time.
- Delete a triggered alert listing from the **Triggered Alerts** page.
- Disable an alert to prevent it from triggering.

Additional alert configuration options

It is recommended to create alerts in the **Search** page and edit them from the **Alerts** page. In rare cases, authorized users might access the **Searches, reports, and alerts** page for the following configurations.

Enable summary indexing

Summary indexing is available on scheduled alerts. It can help you perform analysis or report on large amounts of data over long time ranges. Typically, this is time consuming and can impact performance if several users are running similar searches on a regular basis.

Prerequisites
Ensure that the alert's search generates statistical or summary data.

Steps

1. Using the top-level navigation bar, select **Settings** > **Searches, Reports, and Alerts**.
2. Click **Edit** > **Advanced Edit** for the alert you'd like to modify.
3. To enable the summary index to gather data on a regular interval, search for "alert_type" in the search widow in the upper-left section of the window. Set **alert_type** to **always**.
4. For a scheduled alert, search for "summary" to view the summary index options. Set **action.summary_index** to **true**. If not already specified, this sets the **Alert condition** to "Always". This option is not available for real-time alerts.
5. Click **Save**.

Searches and summary indexing

To use summary indexing with an alert, create a search that computes statistics or a summary for events over a period of time. Search results are saved into a summary index that you designate. You can search over this smaller summary index instead of working with the larger original dataset.

It is typical to use reporting commands in a search that populates a summary index. See Use summary indexing for increased reporting efficiency in the *Knowledge Manager* manual.

Update triggered alert record lifespans

By default, each triggered alert record on the **Triggered Alerts** page expires after 24 hours. You can update the lifespans for triggered alert records on a per-alert basis.

Here are steps for updating the lifespans of the triggered alert records for a specific alert. These steps apply only to alerts that have the "Add to Triggered Alerts" action enabled.

1. From the top-level navigation bar, select **Settings** > **Searches, reports, and alerts**.
2. (Optional) Select **Type** > **Alerts** to filter the list so it displays only alerts.
3. Locate the alert that you want to modify under **Name**.
4. Select **Edit** > **Edit Alert**.
5. Define the lifespan of the triggered alert record by setting the **Expires** field.
   Enter an integer and select a time unit from the dropdown. For example, to have all triggered alert records for this alert have a three-day lifespan, enter **3** and select **day(s)**.
6. Click **Save**.

Alert examples

Alert examples

Use these examples to learn how to use alert types and triggering options. Each example includes a summary of the alerting use case and components. The examples also include steps for creating the alerts.

Scheduled alert example

A scheduled alert searches for events on a regular basis. It triggers an alert action if results meet the conditions that you specify.

Alert example summary

Use case
: Track errors on a Splunk instance. Send an email notification if there are more than five errors in a twenty-four hour period.

Alert type
: Scheduled

Search
: Look for error events in the last twenty-four hours.

Schedule
: Run the search every day at the same time. In this case, the search runs at 10:00 A.M.

Trigger conditions
: Trigger the alert action if the search has more than five results.

Alert action
: Send an email notification with search result details.

Set up the alert

1. From the Search Page, create the following search. `index=_internal " error " NOT debug source=*splunkd.log* earliest=-24h latest=now`
2. Select **Save As > Alert**.

3. Specify the following values for the fields in the **Save As Alert** dialog box.
    - **Title**: Errors in the last 24 hours
    - **Alert type**: Scheduled
    - **Time Range**: Run every day
    - **Schedule:** At 10:00
    - **Trigger condition**: Number of Results
    - **Trigger when number of results**: is greater than 5.
4. Select the **Send Email** alert action.
5. Set the following email settings, using tokens in the **Subject** and **Message** fields.
    - **To**: email recipient
    - **Priority**: Normal
    - **Subject**: Too many errors alert: $name$
    - **Message**: There were $job.resultCount$ errors reported on $trigger_date$.
    - **Include**: Link to Alert and Link to Results

    Accept defaults for all other options.
6. Click **Save**.

Real-time alert example

A real-time alert searches continuously for results in real time. You can configure real-time alerts to trigger every time there is a result or if results match the trigger conditions within a particular time window.

Alert example summary

Use case

Monitor for errors as they occur on a Splunk instance. Send an email notification if more than five errors occur within one minute.

Alert type

Real-time

Search

Look continuously for errors on the instance.

Trigger conditions

Trigger the alert if there are more than five search results in one minute.

Alert action

Send an email notification.

Set up the alert

1. From the Search Page, create the following search. `index=_internal " error " NOT debug source=*splunkd.log*`
2. Select **Save As > Alert**.
3. Specify the following values for the alert fields.
   - ♦ **Title**: Errors reported (Real-time)
   - ♦ **Alert type**: Real-time
   - ♦ **Trigger condition**: Number of Results
   - ♦ **Trigger if number of results**: is greater than 5 in 1 minute.
4. Select the **Send email** alert action.
5. Specify the following email settings, using tokens in the **Subject** and **Message** fields.
   - ♦ **To**: email recipient
   - ♦ **Priority**: Normal
   - ♦ **Subject**: Real-time Alert: $name$
   - ♦ **Message**: There were $job.resultCount$ errors.
   - ♦ **Include**: Link to Alert, Link to Results, Trigger Condition, and Trigger Time.

   Accept defaults for all other options.
6. Click **Save**.

Throttle the real-time alert

Throttle an alert to reduce its triggering frequency and limit alert action behavior. For example, you can throttle an alert that generates more email notifications than you need.

Throttle the example real-time alert. The following settings change the alert triggering behavior so that email notifications only occur once every ten minutes.

1. From the **Alerts** page in the **Search and Reporting** app, select the alert. The alert details page opens.
2. Next to the alert **Trigger conditions**, select **Edit**.
3. Select the **Throttle** option. Specify a 10 minute period.
4. Click **Save**.

Custom trigger condition example

When you create an alert you can use one of the available result or field count trigger condition options. You can also specify a custom trigger condition. The custom condition works as a secondary search on the initial results set.

Alert example summary

Use case
: Use the **Triggered Alerts** list to record `WARNING` error instances.

Alert type
: Real-time

Search
: Look for all errors in real-time.

Triggering condition
: Check the alert search results for errors of type `WARNING`. Trigger the alert action if results include any `WARNING` errors.

Alert action
: List the alert in the **Triggered Alerts** page.

Set up the alert

1. From the **Search and Reporting** home page, create the following search.
   ```
   index=_internal source="*splunkd.log" ( log_level=ERROR OR
   log_level=WARN* OR
   log_level=FATAL OR log_level=CRITICAL)
   ```
2. Select **Save As > Alert**.
3. Specify the following alert field values.
   - **Title**: Warning Errors
   - **Alert type**: Real-time
   - **Trigger condition**: Custom
   - **Custom Condition**: search log_level=WARN* in 1 minute
4. Select the **List in Triggered Alerts** alert action.
5. Click **Save**.

Manual alert configuration with .conf files

Configure alerts in savedsearches.conf

You can use Splunk Web to configure most alerts. If you have Splunk Enterprise, you can configure alerts by editing savedsearches.conf.

- ◊ Only users with file system access, such as system administrators, can configure alerts using the configuration files.
- ◊ Review the steps in How to edit a configuration file in the *Admin Manual*.

Never change or copy the configuration files in the `default` directory. The files in the `default` directory must remain intact and in their original location. Make changes to the files in the `local` directory.

Configuration file paths

Create or edit `savedsearches.conf` in the local directory:
`$SPLUNK_HOME/etc/system/local/`

For apps, create or edit `savedsearches.conf` in the application directory:
`$SPLUNK_HOME/etc/apps/`

Example savedsearches.conf stanza

Alerts use a saved search to look for events. `savedsearches.conf` contains a stanza for each saved search. The following example shows the stanza for a saved search with its alert action settings. In this case, the alert sends an email notification when it triggers.

```
[Too Many Errors Today]
# send an email notification
action.email = 1
action.email.message.alert = The alert condition for '$name$' in the
$app$ fired with $job.resultCount$ error events.
action.email.to = address@example.com
action.email.useNSSubject = 1

alert.suppress = 0
alert.track = 0
```

```
counttype = number of events
quantity = 5
relation = greater than

# run every day at 14:00
cron_schedule = 0 14 * * *

#search for results in the last day
dispatch.earliest_time = -1d
dispatch.latest_time = now

display.events.fields = ["host","source","sourcetype","latitude"]
display.page.search.mode = verbose
display.visualizations.charting.chart = area
display.visualizations.type = mapping

enableSched = 1

request.ui_dispatch_app = search
request.ui_dispatch_view = search
search = index=_internal " error " NOT debug source=*splunkd.log*
earliest=-7d latest=now
disabled = 1
```

For more information, see the savedsearches.conf.example file in the *Admin Manual*.

Configure a script for an alert action

The run a script alert action is officially deprecated. It has been replaced with custom alert actions as a more scalable and robust framework for integrating custom actions. See Using custom alert actions for information on building customized alert actions that can include scripts.

If you have Splunk Enterprise, you can configure an alert to run a shell script or batch file when the alert triggers. This topic shows how to access information about an alert in a script that runs as an alert action.

The script or batch file that an alert triggers must be at either of the following locations:

```
$SPLUNK_HOME/bin/scripts
$SPLUNK_HOME/etc/apps/<AppName>/bin/scripts
```

Working directories for scripts

Specify an absolute path whenever a path is needed. If you use relative paths, it is important to remember that they are rooted in the **Search and Reporting** app's `bin` folder.

Access arguments to scripts that are run as an alert action

When you run a script as an alert action, positional arguments that capture alert information are passed to the script. The positional arguments are also available as environment variables.

You can access information from each argument using the notation in the following table.

Arg	Environment Variable	Value
0	**SPLUNK_ARG_0**	Script name
1	**SPLUNK_ARG_1**	Number of events returned
2	**SPLUNK_ARG_2**	Search terms
3	**SPLUNK_ARG_3**	Fully qualified query string
4	**SPLUNK_ARG_4**	Name of report
5	**SPLUNK_ARG_5**	Trigger reason For example, "The number of events was greater than 1."
6	**SPLUNK_ARG_6**	Browser URL to view the report.
7	**SPLUNK_ARG_7**	Not used for historical reasons.
8	**SPLUNK_ARG_8**	File in which the results for the search are stored. Contains raw results in gzip file format.

You can reference the information captured by these arguments in UNIX shell scripts or Microsoft batch files, as shown below. In other languages, such as perl and python, use the methods native to the language to access script arguments.

```
# UNIX scripts can access environment variables and positional args
$SPLUNK_ARG_0
$0
```

```
# Microsoft batch files capture environment variables reliably
%SPLUNK_ARG_0%
```

Test script that accesses positional arguments

Use the following test script to see the results of accessing the positional arguments.

To use this test script, create an alert that runs the script as an alert action. Then check the contents of the generated `echo_output.txt` file:

```
# $SPLUNK_HOME/bin/scripts/echo.sh
# simple script that writes parameters 0-7 to
# $SPLUNK_HOME/bin/scripts/echo_output.txt
# $SPLUNK_ARG_0 and $0 show how to use the long and short form.

read sessionKey
echo "'$SPLUNK_ARG_0' '$0' '$1' '$2' '$3' '$4' '$5' '$6' '$7' '$8'
'$sessionKey'" >> \
"$SPLUNK_HOME/bin/scripts/echo_output.txt"
```

- Note: The `sessionKey` is URL encoded.

Script example: Write to syslog

You can configure a script for an alert to write to the system log daemon. This is useful if you have syslog set up to send alerts to other applications and you want to include alerts from your Splunk deployment.

1. Create a script, `logIt` that calls `logger`, or any other program that writes to syslog.
   Place the script in `$SPLUNK_HOME/bin/scripts`.
2. Add the following in `logIt`:
   logger $5

   The script can access any of the arguments available when called as an alert action.

3. Create an alert on a report that runs `logIt` as an alert action.
   When the alert triggers, the log entry looks something like this:
   Aug 15 15:01:40 localhost logger: Report [j_myadmin]: The number of events(65) was greater than 10

See Best practices for using UDP when configuring a syslog input, a topic in the Splunk Community Wiki.

Script example: Write to the Windows Event Log

For Windows platforms, you can configure an alert action to run a script that writes to the Windows Event Log.

The following example shows a script that calls the `EVENTCREATE` utility that writes to the Event log. The script can access any of the environment variables available with an alert. You can substitute the EVENTCREATE utility with any command-line executable that writes to the Event Log.

1. Create the following batch file, `logIt.bat`.
   Place the script in `$SPLUNK_HOME/bin/scripts`.
2. Include the following command in the batch file:
   @echo off
   EVENTCREATE /T ERROR /SO Splunk /D %SPLUNK_ARG_5%
   Use the type that best suits the message contained in the argument. This example uses `ERROR`.

3. Create an alert to a report that runs `logIt.bat` as an alert action.