Splunk Crash Course – Creating Alerts

Blue Team Level 1 Certification (Standard) > SI5) Using Splunk > Splunk Crash Course – Creatin... IN PROGRESS

Before we jump into how to create alerts, let's cover what alerts actually are. Alerts trigger when search results meet specific conditions, allowing us to monitor and detect particular activity, such as a user failing to login, or an external IP scanning the organisation. Alerts are triggered, and are then typically investigated by human analysts, who will perform the initial triage of the event, and escalate it to senior analysts if required. This is the main role of a SOC analysts, to triage and investigate SIEM alerts, while senior analysts may also be responsible for developing new detection rules, as well as tuning existing rules to reduce noise and make them more effective.

ALERTING PROCESS

The process of creating alerts can be split into four main steps:

1. **Search Query**
2. **Search Timing**
3. **Alert Triggers**
4. **Alert Actions**

We will cover each of these steps in more detail below.

1 – Search Query

The first task is to decide what activity we want to generate an alert. Is it an external IP trying to SSH into a corporate server? Is it a user account that is having a high number of login failures? Is it local administrator account usage? Rules in Splunk are essentially search queries, so we need to work out what activity we want to detect, and how to write a search query that will identify it.

2 – Search Timing

Now that we have an alert with a search query, we need to set how often Splunk is going to run the search query to look for any activity that makes the alert conditions. Primarily we'll use one of the two options:

- Continuously run this search query to look for related activity in real-time (the majority of SIEM rules)
- Run this search query on a set schedule (typically used to identify changes from baselines and behavioural profiles)

3 – Alert Trigger

If we have a rule that is looking for account logon failures, having it alert every time one logon failure event is generated in the environment, the SIEM is going to get smashed with tons of alerts. To combat this, we can create thresholds within the alert. For example, everyone gets their password wrong sometimes – but getting your password wrong 5, 6 times in a row isn't normal behaviour. We can create an alert that is looking for login failures from Windows domain controllers using Windows Event logs, and set a threshold of 6 per user. Now when an account hits 6 login failures, an alert will generate. We can also combine thresholds with time ranges, so if an account fails to login 6 times within 5 minutes, generate an alert.

4 – Alert Action

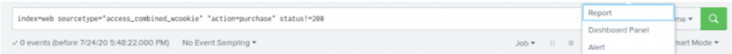
This is where we determine what actually happens when an alert triggers. Some of the default actions include:

- Sending an email notification (typically used for high-profile events that need an immediate response from senior analysts)
- Adding an alert to the list of recently trigged alerts (this is how analysts can identify alerts and work to investigate them)
- Log and index searchable alert events (this actually allows analysts to quickly view all the information related to the alert that trigged)

Splunk also has the ability to allow administrators to write their own custom actions using web hooks, so messages can be created in their own applications, such as a mobile app that informs analysts when a new alert has been triggered and needs to be investigated.

CREATING YOUR OWN RULES

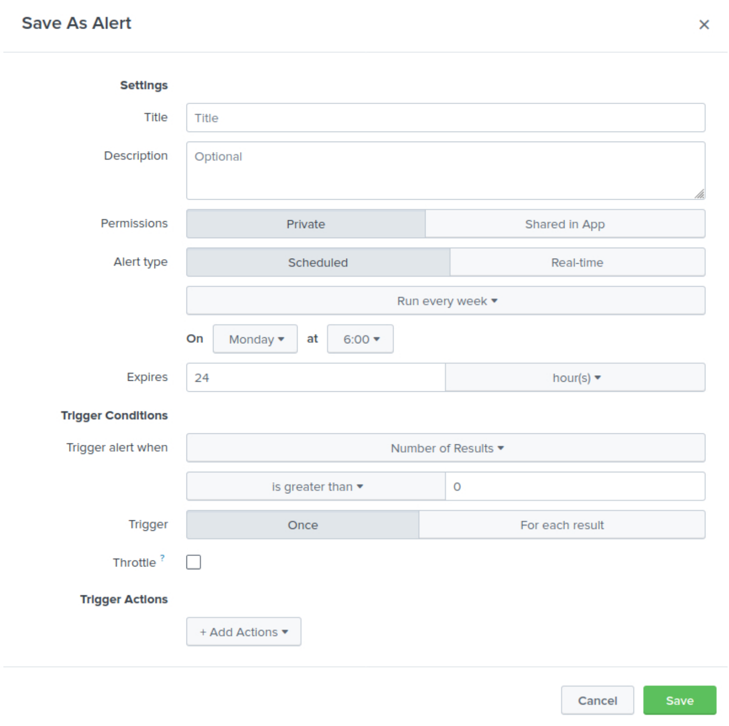
First, we need a search query. Once we've written it, we can click on **'Save As'** and then **'Alert'**.



```
index=web sourcetype="access_combined_wcookie" "action=purchase" status!=200
```

✓ 0 events (before 7/24/20 5:48:22.000 PM)

Then the following windows will be displayed:



Save As Alert

Settings

Title: Title

Description: Optional

Permissions: Private | Shared in App

Alert type: Scheduled | Real-time

Run every week

On Monday at 6:00

Expires: 24 hour(s)

Trigger Conditions

Trigger alert when: Number of Results

is greater than 0

Trigger: Once | For each result

Throttle ? ☐

Trigger Actions

Give the alert a title and description.

Permissions

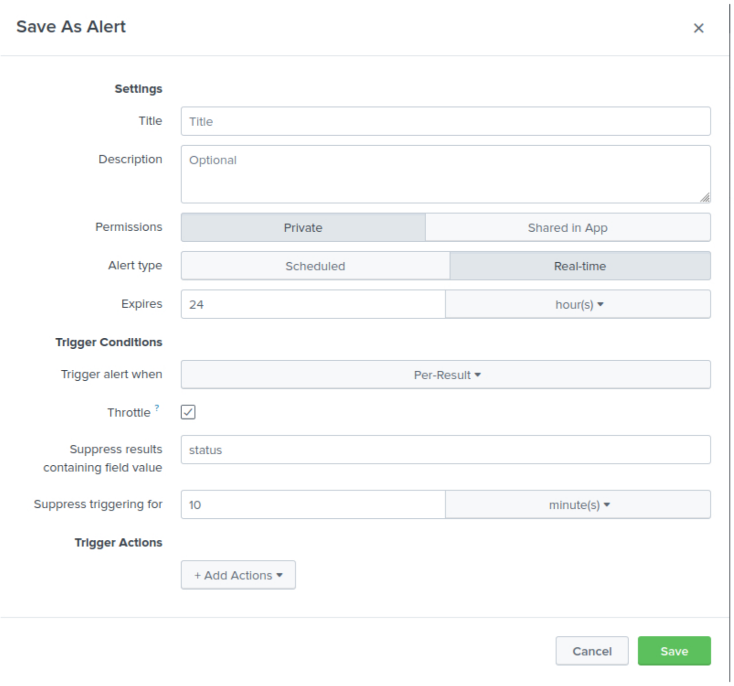
- **Private –** Only you can access, edit, and view triggered alerts.
- **Shared in App –** All users of the app can view triggered alerts, by default, everyone has read access and power user has write access to the alert.

Alert type:

Here we tell splunk how to search for these events that match our alert.

- Scheduled alerts, will search at a defined interval and evaluate trigger condition when the search completes.
- Real time alerts, these run constantly in the background and valuate trigger conditions within a window of

time based on the conditions you define.



Save As Alert ×

Settings

Title: Title

Description: Optional

Permissions: Private | Shared in App

Alert type: Scheduled | Real-time

Expires: 24 hour(s) ▾

Trigger Conditions

Trigger alert when: Per-Result ▾

Throttle ?: ☑

Suppress results containing field value: status

Suppress triggering for: 10 minute(s) ▾

Trigger Actions

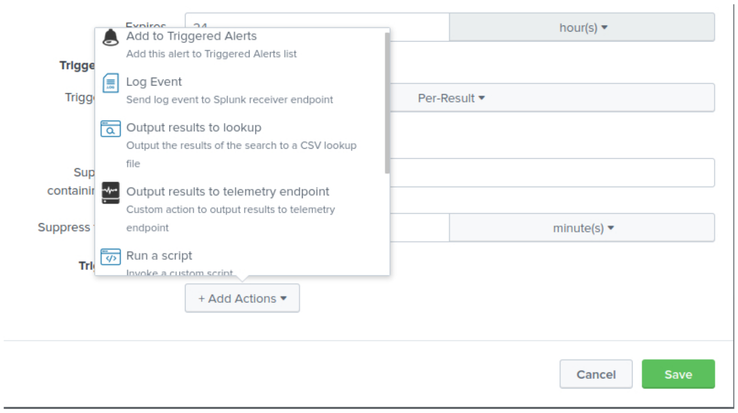
+ Add Actions ▾

Cancel | Save

In the screenshot above we have set the Alert Type as real-time, Expiry after 24 hours, and Trigger Alert is set to 'per result'. We will get results for each match, we can throttle this if too many alerts are being generated. Here we have set it as 'suppress results' which contain 'status' as our search alert contains the 'status' keyword, and will suppress it after 10 minutes.

Lastly, **trigger actions:**

Here we tell Splunk actions to take when an alert is generated, as mentioned it can do the following; trigger alerts, log the event, output results to lookup file, and run a script or output results to an endpoint so it can get the attention of a SOC analyst.



Add to Triggered Alerts
Add this alert to Triggered Alerts list

Log Event
Send log event to Splunk receiver endpoint

Output results to lookup
Output the results of the search to a CSV lookup file

Output results to telemetry endpoint
Custom action to output results to telemetry endpoint

Run a script
Invoke a custom script

Previous Topic

Mark Complete

Next Topic

Back to Lesson