Preventing concurrency issues and skipped searches

A correlation search scans multiple data sources for defined patterns. When the search finds a pattern, it performs an adaptive response action. This is a powerful capability, but when multiple correlation searches all run at the same time, you can have search concurrency issues and skipped searches.

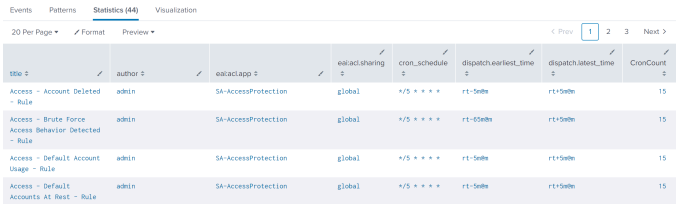
Solution

Identify overcrowded scheduling and then shuffle the searches around to ensure they aren't all running at the same time.

1. (*Optional*) Run the following SPL to list out the cron schedules used by the enabled searches. The results give you a high-level overview of overcrowded schedules so you can easily identify problem areas. You might want to create a pie chart visualization of the results and add it to a dashboard for monitoring.

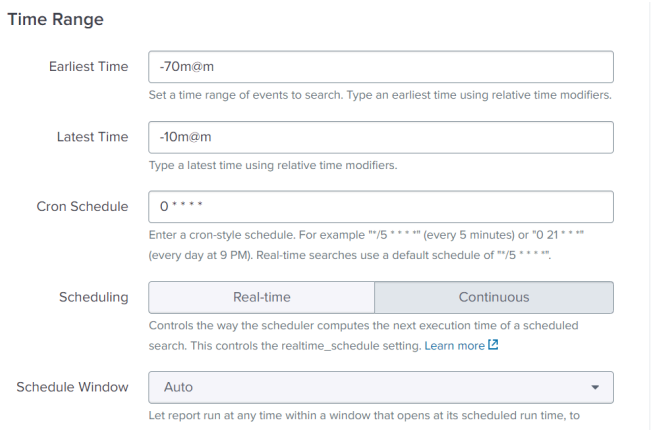
   ```
   | rest splunk_server=local /servicesNS/-/-/saved/searches search="is_scheduled=1" search="action.correlationsearch.enabled=1" search="disabled=0"
   | fields title author eai:acl.app eai:acl.sharing cron_schedule dispatch.earliest_time dispatch.latest_time
   | stats count AS CronCount BY cron_schedule
   | sort - CronCount
   ```

2. Run the previous search again, but change the `stats` command to `eventstats` to list out details for each of the enabled searches.



3. Make note of searches with the same cron schedule.
4. To edit these searches, go to **Configure** > **Content** > **Content Management**.
5. Click the name of a search to open the editing page.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.



6. In the Time Range section, manually adjust the Cron Schedule. For example:
    - If the alerts are run hourly, then stagger which minute the alert starts on. Set your first search to 1 * * * *. Then open another search and set it to 2 * * * *. Then 3* * * * for a third search and so on.
    - If the searches need to run every 5 or 10 minutes, you can splay the cron schedules to allocate searches to run on different minutes.
        - For five-minute intervals, this would be 0-55/5 * * * * for the first search, 1-56/5 * * * * for the second search, then 2-57/5 * * * *, 3-58/5 * * * *, 4-59/5 * * * *, and so on.
        - For ten-minute intervals, this would be 0-50/10 * * * * for the first search, 1-51/10 * * * * for the second search, then 2-52/10 * * * * and so on.
7. In the Time Range section, for the search window, you should account for a delay in ingestion and data model acceleration. We recommend the following settings:
    - Earliest Time:-70m@m
    - Latest Time: -10m@m

Next steps

These additional Splunk resources might help you understand and implement this product tip:

- Docs: Correlation search overview for Splunk Enterprise Security
- Docs: Use cron expressions for alert scheduling

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.