Splunk® Enterprise Security
Administer Splunk Enterprise Security 7.0.1

Upgrade correlation searches in Splunk Enterprise Security

Generated: 6/13/2022 9:54 am

Upgrade correlation searches in Splunk Enterprise Security

Starting in Splunk Enterprise Security version 4.6.0, `correlationsearches.conf` is no longer used to define correlation searches. Instead, `savedsearches.conf` uniquely identifies correlation searches using the `action.correlationsearch.enabled=1` parameter. The `correlationsearches.conf` file is deprecated.

Changes Splunk Enterprise Security makes at upgrade

When you upgrade to Splunk Enterprise Security 4.6.0, Splunk Enterprise Security migrates all correlation searches in your environment from `correlationsearches.conf` to `savedsearches.conf` using the `confcheck_es_correlationmigration.py` script. The migration can take up to five minutes to complete after the upgrade. In a search head cluster, the captain performs the migration.

During the upgrade, Splunk Enterprise Security continues to create notable events without interruption. This change does not prevent or delay notable events from appearing on Incident Review because the `Threat - Correlation Searches - Lookup Gen` saved search continues to use the contents of both `correlationsearches.conf` and `savedsearches.conf` to populate the `correlationsearches` KV Store collection used by Incident Review.

Changes you have to make after upgrade

After upgrading to Splunk Enterprise Security 4.6.0 or later, you have to make additional changes.

- Check `correlationsearches.conf` for search definitions that would indicate that a search did not migrate successfully. Migrated searches only exist in `savedsearches.conf`. If a search did not get migrated, migrate the `correlationsearches.conf` entries manually to `savedsearches.conf` using the parameter definitions below.
- Update searches that call the `correlationsearches` REST endpoint.
    - For example, a search that displays a list of correlation searches in your environment would change from

      ```
      | rest splunk_server=local /services/alerts/correlationsearches | rename eai:acl.app as app, title as csearch_name | table app security_domain csearch_name description
      ```

      to

      ```
      | rest splunk_server=local count=0 /services/saved/searches | where match('action.correlationsearch.enabled', "1|[Tt]|[Tt][Rr][Uu][Ee]") | rename eai:acl.app as app, title as csearch_name, action.correlationsearch.label as csearch_label, action.notable.param.security_domain as security_domain | table csearch_name, csearch_label, app, security_domain, description
      ```
    - See List correlation searches in Splunk Enterprise Security for more examples of updated searches.

Custom search macros that reference the `correlationsearches` KV Store collection continue to work as before, but consider updating them anyway.

`correlationsearches.conf` parameter translation to `savedsearches.conf`

All `correlationsearches.conf` parameters now exist in `savedsearches.conf` and the `correlationsearches.conf` file has been deprecated. Do not update it directly except to manually migrate correlation search definitions.

Identification parameters for correlation searches

New parameters identify whether a saved search is a correlation search and the name of the correlation search.

correlationsearches.conf parameter in pre-4.6.0 versions	savedsearches.conf parameter starting in 4.6.0	Notes
N/A	action.correlationsearch=0	This is an internal parameter and can be ignored.
A stanza for the search exists	action.correlationsearch.enabled=1	This parameter identifies a saved search as a correlation search.
rule_name	action.correlationsearch.label	This parameter provides the name of the correlation search.
description	description	This parameter provides the description of the correlation search.

Notable event parameters for correlation searches

The action.notable parameter identifies a notable event associated with a correlation search. The parameters that describe additional details associated with the notable event now exist in the savedsearches.conf file.

correlationsearches.conf parameter in pre-4.6.0 versions	savedsearches.conf parameter starting in 4.6.0
security_domain	action.notable.param.security_domain
severity	action.notable.param.severity
rule_title	action.notable.param.rule_title
rule_description	action.notable.param.rule_description
nes_fields	action.notable.param.nes_fields
drilldown_name	action.notable.param.drilldown_name
drilldown_search	action.notable.param.drilldown_search
default_status	action.notable.param.default_status
default_owner	action.notable.param.default_owner

Related search parameters for correlation searches

Searches related to a correlation search, such as the context-generating searches associated with a correlation search that uses extreme search, are now part of a JSON blob action.correlationsearch.related_searches parameter.

correlationsearches.conf parameter in pre-4.6.0 versions	savedsearches.conf parameter starting in 4.6.0
related_search_name = Endpoint - Emails By Source - Context Gen related_search_name.0 = Endpoint - Emails By Destination Count - Context Gen	action.correlationsearch.related_searches = [\ "Endpoint - Emails By Source - Context Gen",\ "Endpoint - Emails By Destination Count - Context Gen"\ ]

Example correlation search stanzas from this version and previous versions

The savedsearches.conf stanza for a correlation search looks as follows starting in 4.6.0.

```
[Access - Concurrent App Accesses - Rule]
action.correlationsearch = 0
action.correlationsearch.enabled = 1
```

```
action.correlationsearch.label = Concurrent Login Attempts Detected
action.email.sendresults = 0
action.notable = 0
action.notable.param.security_domain = access
action.notable.param.severity = medium
action.notable.param.rule_title = Concurrent Access Event Detected For $user$
action.notable.param.rule_description = Concurrent access attempts to $app1$ by $user$ from two different
sources( $src1$, $src2$ ) have been detected.
action.notable.param.nes_fields = user
action.notable.param.drilldown_name = View access attemps by $user$
action.notable.param.drilldown_search = | datamodel Authentication Authentication search | search
Authentication.user="$user$"
action.risk = 1
action.risk.param._risk_object = user
action.risk.param._risk_object_type = user
action.risk.param._risk_score = 20
alert.suppress = 1
alert.suppress.fields = user
alert.suppress.period = 86300s
alert.track = false
cron_schedule = 10 * * * *
description = Alerts on concurrent access attempts to an app from different hosts. These are good indicators
of shared passwords and potential misuse.
disabled = True
dispatch.earliest_time = -70m@m
dispatch.latest_time = -5m@m
enableSched = 1
is_visible = false
request.ui_dispatch_app = SplunkEnterpriseSecuritySuite
search = | tstats `summariesonly` count from datamodel=Authentication.Authentication by
_time,Authentication.app,Authentication.src,Authentication.user span=1s |
`drop_dm_object_name("Authentication")` | eventstats dc(src) as src_count by app,user | search src_count>1
| sort 0 + _time | streamstats current=t window=2 earliest(_time) as previous_time,earliest(src) as
previous_src by app,user | where (src!=previous_src) | eval time_diff=abs(_time-previous_time) | where
time_diff<300
```

In previous versions of Splunk Enterprise Security, the `savedsearches.conf` and `correlationsearches.conf` definitions for the same correlation search would look as follows. `savedsearches.conf`

```
[Access - Concurrent App Accesses - Rule]
action.email.sendresults            = 0
action.risk                         = 1
action.risk.param._risk_object      = user
action.risk.param._risk_object_type = user
action.risk.param._risk_score       = 20
alert.suppress                      = 1
alert.suppress.fields               = user
alert.suppress.period               = 86300s
alert.track                         = false
cron_schedule                       = 10 * * * *
disabled                            = True
dispatch.earliest_time              = -70m@m
dispatch.latest_time                = -5m@m
enableSched                         = 1
is_visible                          = false
request.ui_dispatch_app             = SplunkEnterpriseSecuritySuite
search                              = | tstats `summariesonly` count from
datamodel=Authentication.Authentication by _time,Authentication.app,Authentication.src,Authentication.user
span=1s | `drop_dm_object_name("Authentication")` | eventstats dc(src) as src_count by app,user | search
src_count>1 | sort 0 + _time | streamstats current=t window=2 earliest(_time) as
previous_time,earliest(src) as previous_src by app,user | where (src!=previous_src) | eval
time_diff=abs(_time-previous_time) | where time_diff<300
```

correlationsearches.conf

```
[Access - Concurrent App Accesses - Rule]
security_domain     = access
severity            = medium
rule_name           = Concurrent Login Attempts Detected
description         = Alerts on concurrent access attempts to an app from different hosts. These are good
indicators of shared passwords and potential misuse.
rule_title          = Concurrent Access Event Detected For $user$
rule_description    = Concurrent access attempts to $app1$ by $user$ from two different sources( $src1$,
$src2$ ) have been detected.
nes_fields          = user
drilldown_name      = View access attemps by $user$
drilldown_search    = | datamodel Authentication Authentication search | search
Authentication.user="$user$"
default_owner       =
default_status      =
```