Broken Hosts App for Splunk Documentation

Release 4.0.1

Hurricane Labs

Jul 05, 2023

Contents

The Broken Hosts App for Splunk is a useful tool for monitoring data going into Splunk. It has the ability to alert when hosts stop sending data into Splunk, as well as inspect the last time the final combination of data was received by Splunk.

If the arrival of the final log for the index/sourcetype/host combination is later than expected, the Broken Hosts App will send an alert. This allows for quick status detection of the hosts and fast issue resolution.

The Broken Hosts App for Splunk is the app for monitoring missing data in Splunk. The app's three main objectives include:

1. Alerting when data is missing from Splunk in order to determine the cause.
2. Utilizing saved searches to facilitate rapid detection of the missing data.
3. Creating dashboards for visualization to help with further investigations.

CHAPTER 1

Features

- Detects gaps in data being collected into Splunk
- Detects unexpected latency in data being collected into Splunk
- Generates statistics about data being collected into Splunk for other uses
- Includes dashboards for investigating broken data sources
- Use Splunk modular alert actions for sending alerts
- Lookup- and Eventtype-based configuration

CHAPTER 2

Quickstart

If you're an existing Broken Hosts user, please be sure to review our *Upgrading* documentation.

1. Install the Broken Hosts App for Splunk on your ad-hoc search head.
2. Use the `Broken Hosts` dashboard to determine appropriate baselines for all of your critical data.
3. Use the `Configure Broken Hosts Lookup` dashboard to configure your baselines and create suppressions.
4. Configure alert actions on the `Broken Hosts Alert Search` saved search in the Broken Hosts App for Splunk.
5. Enable the `Broken Hosts Alert Search` saved search in the Broken Hosts App for Splunk.

CHAPTER 3

Known Issues

- Future hosts in the `Broken Hosts Alert Search` may not match the future hosts displayed on the `Broken Hosts` dashboard. Future host detection will be moved to a separate search in a future release of the Broken Hosts App.
- search-time renaming of sourcetypes is not taken into account
- "Configure Broken Hosts Lookup" doesn't handle additional fields added to expectedTime lookup

CHAPTER 4

Documentation

4.1 Usage

4.1.1 Installation

New Installations

Note: If you are installing the Broken Hosts App on a search head cluster, follow Splunk's documentation for app installation

1. On the Splunk toolbar, select **Apps > Find More Apps**.
2. In the search box, search for **broken hosts**.
3. Next to the Broken Hosts App for Splunk, select the **Install** button.
4. Follow the prompts and, if necessary, restart Splunk.
5. (Optional, but recommended) - Backfill summary index by running this CLI command:

```
cd $SPLUNK_HOME/bin && ./splunk cmd python fill_summary_index.py -app broken_hosts -
↪name bh_stats_gen -dedup true -et -30d@d -lt now -j 10 -showprogress true
```

Once the app is installed, please review the *Configuration* documentation.

Upgrading

1. On the Splunk toolbar, select **Apps > Manage Apps**.
2. Find the Broken Hosts App for Splunk.
3. Under the Version column, select **Update to 4.0.x**.
4. Follow the prompts and, if necessary, restart Splunk.
5. Follow any version-specific upgrade instructions below.

Upgrading to 4.0.x from 3.x or below

Starting with Broken Hosts 4.0.1, the `Broken Hosts Sanity Check` has been split into two pieces, neither of which is enabled by default. To restore similar behavior to previous versions, follow these steps:

1. (Optional, but recommended) - Backfill summary index by running this CLI command:

```
cd $SPLUNK_HOME/bin && ./splunk cmd python fill_summary_index.py -app broken_hosts -
↪name bh_stats_gen -dedup true -et -30d@d -lt now -j 10 -showprogress true
```

2. Review your `search_additions` macro to determine which functionality must occur in the stats generation phase, and which must occur in the alert generation phase.
3. Copy the stats generation parts of your existing `search_additions` macro to the new `bh_stats_gen_additions` macro.
4. Copy the alert generation parts of your existing `search_additions` macro to the new `bh_alert_additions` macro.
5. Enable the `Broken Hosts Alert - by contact` search.

Afterwards, we recommend reviewing the *Configuration* documentation to get a feel for how the new split searches work, and things you can do now with the standalone alerting searches that were impossible previously with the unified search.

Upgrading to 3.3.3

Starting with Broken Hosts 3.3.3, the Broken Hosts Lookup is stored in KV store rather than in a CSV file. Once you have completed this upgrade, follow these steps to convert your lookup file to KV Store:

1. Open a search panel and run the following search:

```
| inputlookup expectedTime.csv
```

2. Confirm the results appear as expected - this should display your existing Broken Hosts Lookup.
3. Run the following search to dump the existing lookup into the new KV Store lookup.

```
| inputlookup expectedTime.csv | outputlookup expectedTime
```

4. Go to the `Configure Broken Hosts Lookup` dashboard to confirm that the configuration is correct.

4.1.2 Configuration

Enabling the saved search(es)

Prior to Broken Hosts 4.0, the majority of the app was contained in a search called "Broken Hosts Sanity Check." This search was a slow, monolithic search that made customized alerting difficult. Beginning in Broken Hosts 4.0, the search has been broken into two pieces (for more details, see the *Saved Searches* documentation). The `bh_stats_gen` search is enabled by default and does not require configuration. Alerting, however, is done through a separate search.

Broken Hosts 4.2 ships with four example alerting searches, `Broken Hosts Alert Search`, `Broken Hosts Alert - by contact`, `Broken Hosts Alert - Volume Alerting`, and `Broken Hosts Alert - Volume Alerting with Seasonality`. These searches are meant primarily to be examples of how to build alerting using the Broken Hosts data, and can easily be duplicated, tweaked, or replaced altogether depending on your requirements. If you're new to Broken Hosts, we suggest starting with `Broken Hosts Alert Search` and

customizing from there. If you're upgrading from an older version of Broken Hosts and want to continue getting the alerts you're used to, you can use `Broken Hosts Alert - by contact`.

`Broken Hosts Alert - Volume Alerting` and `Broken Hosts Alert - Volume Alerting with Seasonality` provide new options for alerting on indexes that may have ceased proper logging in a way that wouldn't be detected with traditional alerting. Both of these searches use stastical tests on the existing data from `bh_stats_gen`. The values set as thresholds for these statistical methods are meant to be a base to work off of, and may need to be adjusted to work correctly with your data.

Modifying the macros

There are a number of macros defined within the Broken Hosts app to allow users to customize the behavior of the stock searches without significant effort. Some of the macros apply to the stats collection search, while others are used within the alert searches.

For more information on the macros available for fine-tuning the Broken Hosts app, see the *Macros* documentation.

4.1.3 Advanced Examples

Custom stats gen searches

```
multiple check point firewalls, one management server
"firewall" isn't tracked by index/sourcetype/host
use stats gen search to output stats gen data w/ extra field ("firewall")
use eventtype aggregation
    eventtype: orig_index=checkpoint orig_host=management
    name: bh_aggregate-%orig_index%,%orig_sourcetype%,%firewall%
entries in lookup are index=firewall, sourcetype=checkpoint\*, host=firewall
```

4.2 Architecture

4.2.1 Saved Searches

bh_stats_gen

The `bh_stats_gen` search is responsible for generating statistics about data coming into Splunk. The results are written to the `summary` index, to be picked up and read by other searches for alerting purposes. It can be fine-tuned using the `bh_stats_gen_contraints` and `bh_stats_gen_additions` macros. By default, it will look over all data sources that have sent logs over the past 36 hours. If you have a data source that is regularly delayed longer that you wish to monitor, this time range will need to be adjusted.

Broken Hosts - Auto Sort

The `Broken Hosts - Auto Sort` search was implemented in order to optimize the ordering of the Broken Hosts Lookup. Because the lookup is evaluated in a first-match fashion, the ordering of the lookup is critical to preventing incorrect matches. You can view more information about the ordering of the lookup in the *Saved Searches* documentation.

This search modifies the Broken Hosts Lookup in the following ways:

1. Entries are reordered based on the ordering rules defined in the *Saved Searches* documentation.

2. All fields are converted to lower case, as the lookup is case insensitive.

Broken Hosts Alert Search

`Broken Hosts Alert Search` is the recommended way to get started building your own custom alerting search. This search produces a single output row for each broken item, and ignores the `contact` field from the lookup completely. There are no alert actions defined on this search, so you are free to configure them as needed. A few examples include:

- Add an email alert action to send a tuning report to your Splunk admins
- Add a webhook alert action to create tickets in your ticketing system

You can also create clones of this search to enable different alerting for different types of data. For example, you may want to send email notifications to your Windows server admins when a server stops sending `WinEventLog:Security` but want to trigger a ticket to your helpdesk when your anti-virus system stops sending logs. You can even run a version of this search on your `Enterprise Security` search head to generate notable events.

Broken Hosts Alert - by contact

`Broken Hosts Alert - by contact` is primarily intended for anyone upgrading from an older version of Broken Hosts. This search groups the alert lines by the `contact` field from the lookup, and each contact will receive one email (the email action is configured by default on this search). This search also relies on the `default_contact` macro to populate the contact when none is defined in the lookup table.

If you're coming from an older version of Broken Hosts and choose to implement this search, we'd still recommend you review the new `Broken Hosts Alert Search` as you may find additional uses from it that were difficult or impossible in previous versions of the app.

Broken Hosts Alert - Volume Alerting

`Broken Hosts Alert - Volume Alerting` and `Broken Hosts Alert - Volume Alerting with Seasonality` are two example searches that can be used to generate alerts on indexes that may have stopped ingesting data properly while still generating some amount of logs. Both searches use a combination of standard score (z-score), moving averages, and percentiles to determine whether or not log volume is anamalously low for that index. `Broken Hosts Alert - Volume Alerting with Seasonality` additionally factors in the time of day, day of the week, and whether the day is a holiday to determine normal logging activity for indexes whose volume may be sensitive to user activity.

The macro `bh_volume_alerting_indexes` is used to designate which indexes should be alerted on. If both `Broken Hosts Alert - Volume Alerting` and `Broken Hosts Alert - Volume Alerting with Seasonality` are needed, a new macro can be created and used to designate the indexes that should be used for each search.

4.2.2 Macros

bh_stats_gen_constraints

The `bh_stats_gen_constraints` macro is used to control what data is examined by the `bh_stats_gen` search when generating the metrics used by the alerting searches. The default behavior is to exclude all data in the summary index, and all data from the stash sourcetype, but include all other data.

NOTE: This macro is used within a `tstats` command, and therefore the macro's must be valid `tstats` syntax.

bh_stats_gen_additions

The `bh_stats_gen_additions` macro is used to insert arbitrary SPL into the `bh_stats_gen` search in order to transform data before it is written to the summary index.

Example: use `eventstats` and `eval` statements to calculate custom metrics to be stored in the summary data.

bh_alert_additions

The `bh_alert_additions` macro is used to insert arbitrary SPL into the alerting searches, in order to transform data before it is written to the summary index.

Example: Apply subsearch logic from a monitoring system to automatically exclude hosts that are known to be offline

default_contact

The `default_contact` macro is used only for the `Broken Hosts Alert - by contact` search. It is used to set the default email address for items that don't have a separate contact listed in the `contact` column of the lookup table.

default_expected_time

The `default_expected_time` macro is used to set a default `lateSecs` value for things not defined in the lookup. The `lateSecs` value tells Broken Hosts how long a specific source of data is allowed to go without sending data before an alert should be triggered. This setting is in seconds, and defaults to 14400 (4 hours).

bh_volume_alerting_indexes

The `bh_volume_alerting_indexes` macro is used in the searches `Broken Hosts Alert - Volume Alerting` and `Broken Hosts Alert - Volume Alerting with Seasonality`. It contains a comma separated list of indexes.

4.2.3 Eventtype Aggregations

Using eventtypes to aggregate data

A common request for users of older versions of Broken Hosts was to be able to aggregate certain types of data together. For example, if any of the `WinEventLog` sourcetypes are coming in from a particular Windows host, that's usually enough to feel comfortable that things are working as expected. While this was possible in those versions of Broken Hosts thanks to the `search_additions` macro, that macro would quickly become complex and hard to manage. Starting with Broken Hosts 4.0, however, there's now an easier mechanism for defining complex aggregations.

Eventtype Aggregations provide a simple, Splunk-native way to define these complex aggregations. Eventtype Aggregations are eventtypes named in a specific format: `bh_aggregate-$index,$sourcetype,$host`. The `$index`, `$sourcetype`, and `$host` here can be replaced by either a field placeholder (`%orig_index%`, for example) or with a static value. Using a static value, such as `WinEventLog` for `$sourcetype`, allows you to group matching data sources together. It is important to note

This concept is likely best illustrated by an example: Imagine you have a pfSense firewall, along with the pfSense TA. This means the syslog from your firewall is coming into Splunk, and is split into several different sourcetypes. However, pfSense has one stream of syslog, and if any of these sourcetypes is working, it is generally safe to assume

that the syslog function in pfSense is operational. To aggregate these sourcetypes together, you could use an eventtype similar to the following:

```
orig_sourcetype=pfsense*
```

You would then name this eventtype something like `bh_aggregate-%orig_index%,pfsense,%orig_host%`. Once you have this created, you can add a single line to the Broken Hosts Lookup, using the actual index and host, but using "pfsense" for the sourcetype. Now, for each pfSense firewall you have, you will receive one alert if **all** of the sourcetypes stop coming in for that firewall. Without aggregations, you would instead receive an alert if **any** sourcetype stopped coming in for that firewall.

Suppressions

In addition to setting `lateSecs` to `0` in the Broken Hosts Lookup, the Broken Hosts app also supports an eventtype-based suppression mechanism. This allows you to access all of the fields available in the summary data, including the `date_*` fields, allowing you to create some very complex suppressions using eventtypes that would otherwise be impossible with just the lookup. The naming scheme for these eventtypes is `bh_suppress-label`, where label can be any arbitrary text (assuming it produces a valid eventtype name).

For example, if you wanted to suppress events in your proxy index off-hours, you could create an eventtype called `bh_suppress-proxy_off_hours` similar to the following:

```
orig_index=proxy date_wday="saturday" OR date_wday="sunday" OR date_hour<8 OR date_
↪hour>17
```

4.2.4 Broken Hosts Lookup

:: implemented in kvstore describe how ordering matters case matching wildcards how editing works through the dashboard

Using the Broken Hosts Lookup

There are seven fields in this lookup table (all fields are case *insensitive*):

- `index` - The index for the data that you would like to match - this field does accept wildcards - this field is required
- `sourcetype` - The sourcetype for the data that you would like to match - this field does accept wildcards - this field is required
- `host` - The host for the data that you would like to match - this field does accept wildcards - this field is required
- `lateSecs` - The amount of time (in seconds) that the index/sourcetype/host combination is allowed to be late before it alerts - this field is required
- `suppressUntil` - Alerts for the index/sourcetype/host combination will be suppressed until this date - since we use the "convert auto()" function for this field, you can use any date format that converts to a number - we recommend: "MM/DD/YYYY HH:MM:SS" or epoch time - this field is optional
- `contact` - The email address where you would like the alert to be sent - if this is blank, the email address from the default_contact macro will be used - this field is optional
- `comments` - Any comments that you would like to add for that line of the lookup table. This information is not used in the alert. This field is typically used to record information about why the entry is needed, when it was added, who added it, or any other details. This field is optional

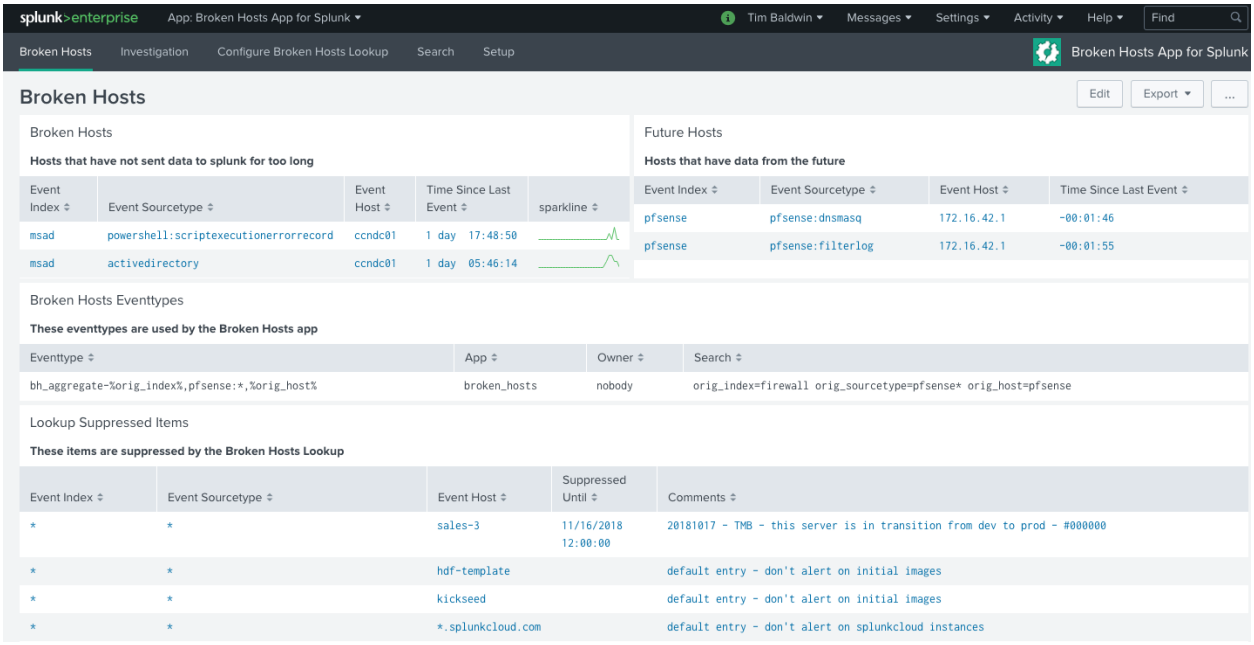
Ordering

Ordering of entries in the Broken Hosts Lookup is important, but the Broken Hosts App ships with a saved search that will re-order the lookup table in a logical way. As a result of several years analyzing expected behavior across our customers, we've determined that the following order is as follows:

1. Entries where index=* AND sourcetype=* AND alerting is temporarily suppressed
2. Entries where sourcetype=* AND alerting is temporarily suppressed
3. Entries where index=* AND alerting is temporarily suppressed
4. Entries where host=* AND alerting is temporarily suppressed
5. Entries where index=* AND host=* AND alerting is temporarily suppressed
6. Entries where sourcetype=* AND host=* AND alerting is temporarily suppressed
7. Entries where alerting is temporarily suppressed
8. Entries where index=* AND sourcetype=* AND alerting is permanently suppressed
9. Entries where lateSecs is temporarily modified
10. Entries where sourcetype=* AND lateSecs is temporarily modified
11. Entries where index=* AND lateSecs is temporarily modified
12. Entries where host=* AND lateSecs is temporarily modified
13. Entries where index=* AND sourcetype=* AND lateSecs is temporarily modified
14. Entries where index=* AND host=* AND lateSecs is temporarily modified
15. Entries where sourcetype=* AND host=* AND lateSecs is temporarily modified
16. Entries where alerting is permanently suppressed
17. Entries where lateSecs is permanently modified, or host=* AND alerting is permanently suppressed, or host=* AND lateSecs is permanently modified, or sourcetype=* AND host=* AND alerting is permanently suppressed
18. Entries where index=* AND host=* AND alerting is permanently suppressed
19. Entries where sourcetype=* AND alerting is permanently suppressed
20. Entries where index=* AND alerting is permanently suppressed
21. Entries where sourcetype=* AND lateSecs is permanently modified
22. Entries where index=* AND lateSecs is permanently modified
23. Entries where index=* AND sourcetype=* AND lateSecs is permanently modified
24. Entries where index=* AND host=* AND lateSecs is permanently modified
25. Entries where sourcetype=* AND host=* AND lateSecs is permanently modified
26. Default entries

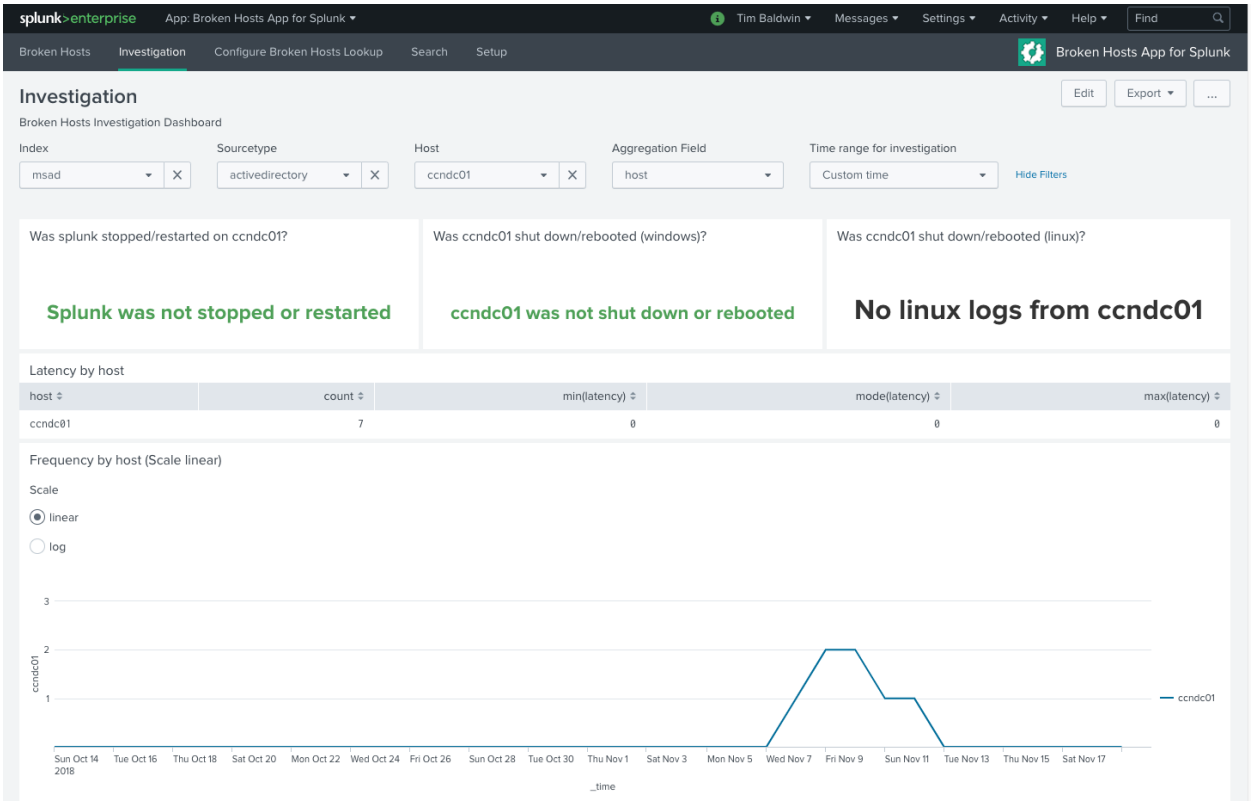
4.2.5 Dashboards

Broken Hosts Dashboard



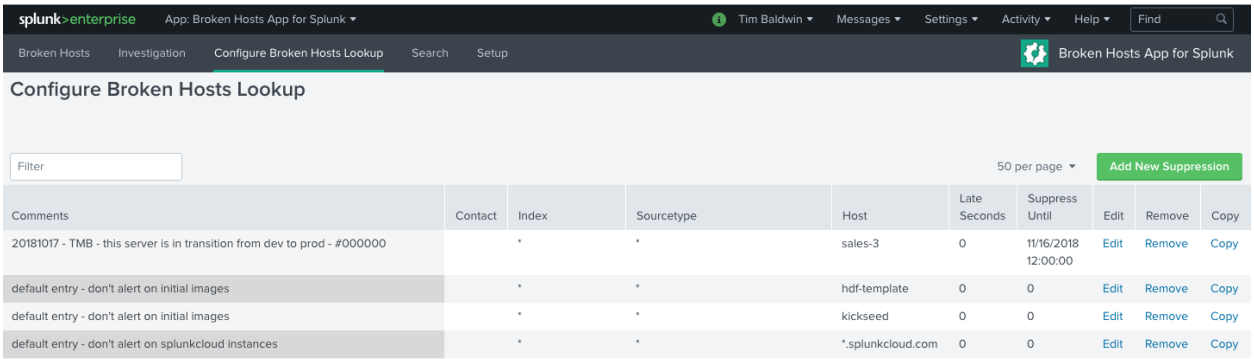
The `Broken Hosts` dashboard is the main overview dashboard for the Broken Hosts app. It provides you with a quick glance into hosts that are not sending data, hosts that are sending data with a timestamp in the future, Eventtype Aggregations and Suppressions, and your "suppressed" items (note: suppressed here refers to items with a `lateSecs` value of `0`, meaning to never alert). Clicking on any of the broken or future hosts will take you to the `Investigation Dashboard` where you can get additional information in order to troubleshoot the data.

Investigation Dashboard



The `Investigation` dashboard can be used to troubleshoot why data has stopped coming in for a particular `index`/`sourcetype`/`host` combination. The filters let you select the data you are interested in, and you can also select the field to aggregate by. This is useful to determine whether a particular host or source is having issues. You can also identify the frequency at which data comes into Splunk in order to determine an appropriate `lateSecs` value, and quickly see whether Splunk, or the host itself, was recently stopped or restarted.

Configure Broken Hosts Lookup



The `Configure Broken Hosts Lookup` dashboard is where you configure the lateSecs for a particular `index`/`sourcetype`/`host` combination. You can also provide comments and an expiration time for the configuration (if, for example, you have a maintenance window for a firewall and it is expected to be offline and not sending logs for a certain period of time). You can also set the `contact` field if you're using the `Broken Hosts Alert - by contact` search.

4.2.6 Advanced Configuration

In addition to all of the Splunk-native configurations, the Broken Hosts app has additional internal configuration. These items are considered "advanced" and may or may not be useful to you. These settings can be found in `bh.conf`.

[validation]

- `comments_must_have_ticket_number` (boolean) - Primarily intended for Hurricane Labs managed Splunk customers. Enforces a restriction on the `comment` field of the Broken Hosts Lookup requiring a 5-or-more digit number to be entered for change management purposes (in the format `#12345`). The default value for this setting is `false`.

CHAPTER 5

Changelog

5.1 Version 4.0.5 (RELEASE PENDING)

- update bh_stats_gen to use a more meaningful time for the summary events
- update the alert searches to no longer look into the future for summary events, since that's not possible
- include wineventlog aggregation
- make pfsense aggregation work with splunk web validation
- make pfsense aggregation more generic to apply more broadly
- dropdowns on Configure Broken Hosts lookup now paginate to help prevent against browser crashing when loading

extremely large data-sets

5.2 Version 4.0.4 (2018-12-12)

- updated bh_stats_gen search to fix a bug that might cause false positives
- set eventtypes to be local to the app instead of global

5.3 Version 4.0.3 (2018-12-10)

- updated AutoSort to allow for arbitrary fields
- update investigation panel to have a more useful graph
- fixed type in app.conf that was preventing successful vetting

5.4 Version 4.0.2 (2018-11-14)

- Revamped architecture
  - Decouple stats generation from alert generation
  - Eventtype-based aggregations and suppressions
- Additional investigation dashboards
- KV Store auto-sort functionality (enabled by default) to prevent false positive matches
- Fixed an issue when using Chrome v70 that caused loss of data in the `expectedTime` lookup

5.5 Version 3.3.6 (2018-03-18)

- Row reordering feature added to 'Configure Broken Hosts Lookup' page. Can drag rows using the 'Comments' column.
- 'Add New Suppression' button added to top right to make more visible.
- Ability to Copy formatted row data to clipboard
- Added expectedTime_tmp for backup purposes.
  - In edge cases where KV Store is being updated after a row-reorder on Configure page and user refreshes, KV Store data could be lost. For this reason, every change made backs up the current version to a expectedTime_tmp KV Store first
  - On initial load of the table it will check if expectedTime is empty, if it is it will then check expectedTime_tmp for data and use that as a backup in case the KV Store was emptied. If both are empty then it is assumed this is a new install and the user has an option to add default values to the KV Store.

5.6 Version 3.3.5 (2018-01-04)

- updated the savedsearch to account for sourcetype rewrites

5.7 Version 3.3.4 (2017-12-13)

- Removed unnecessary inputs.conf

5.8 Version 3.3.3 (2017-12-12)

- The expectedTime lookup definition now references a KV Store instead of a lookup file
- Removed bin/ directory - Python script for generating lookup is no longer needed
- Removed lookups directory as it is now using a KV Store [expectedTime]
- lateSecs field now accepts Splunk's relative time format e.g. -1d@d OR 0 for 'Always Suppress'
- New dashboard: "Configure Broken Hosts Lookup" allows for CRUDing expectedTime KV Store
  - Applies validation to help ensure proper values are added into the lookup

  - Table highlights when two conditions are met:
    * If lateSecs is set to 'Always Suppress' and but a suppressUntil date has been provided.
    * If suppressUntil has a date that is in the past.
- New alert: "Broken Hosts – Suppress Until Is Set Past Date"
  - Runs nightly at 12:01am to check if any suppressUntil values are in the past
  - Alerts pre-defined contact

5.9 Version 3.3.2 (2017-08-29)

- fixed a bug where the the broken hosts dashboard would show the wrong value for "Time Since Last Event"
- updated the app to work if the app directory is renamed
- updated the order of fields in the broken hosts dashboard
- reordered default expectedTime lookup table to be alphabetical
- added "cim_modactions" index to the default suppressions
- added cisco:ios default suppression
- added pan_config and pan:config default suppressions

5.10 Version 3.3.1 (2017-06-12)

- bug fixes for splunk certification
- scale icon sizes down to splunk approved sizes

5.11 Version 3.3.0 (2017-06-02)

- updated savedsearch to include any hosts that are sending logs from the future
- added the ability to add custom search additions to make the search more flexible
- added dashboard panel to show suppressed items
- updated dashboard panels to show currently broken items, and all items from the future
- added sparkline to the dashboard panels

5.12 Version 3.2.1 (2016-11-14)

- updated suppression so that alerts are triggered properly
- added a link to 'setup' in the nav menu

5.13 Version 3.2.0 (2016-11-14)

- modified the savedsearch to use 'tstats' instead of 'metadata' to allow use of sourcetype for tuning
- updated the savedsearch schedule to run every 30 minutes (because tstats takes longer than metadata)
- updated the savedsearch suppression to suppress for 2 hours instead of 1
- updated the savedsearch suppression to include sourcetype
- updated expectedTime lookup table to add a 'sourcetype' column
- updated first_time script to add 'sourcetype' column to lookup table
- added Broken Hosts dashboard
- updated documentation to include Broken Hosts dashboard information
- added app nav color

5.14 Version 3.1.1 (2016-08-09)

- added script to automatically create the lookup if it doens't already exist
- expanded readme information

5.15 Version 3.1.0 (2016-06-29)

- Added setup page with default contact and default allowable lateness

5.16 Version 3.0.0 (2016-06-24)

- Another major rewrite
- Added the ability to suppress an item
- Added the ability to send different items to different contacts

5.17 Version 2.2.1 (2016-04-14)

- force host to lowercase for comparisons

5.18 Version 2.2 (2016-04-14)

- fixed issue with the index exclusions in the search
- reversed the order of the release notes, putting new version at the top

5.19 Version 2.1 (2016-01-05)

- wildcard in lookup table instead of empty quoted string
- app is visible (to allow the "run" button on the saved search to work)
- initial lookup table is now named with .sample extention to not over-write any previous tuning

5.20 Version 2.0 (2015-10-20)

- complete re-write of the app from scratch
- uses dbinspect and metadata commands to make this search much faster
- uses a lookup table to make tuning a breeze

CHAPTER 6

License

```
The MIT License (MIT)

Copyright (c) 2018 Hurricane Labs, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```