Forward-looking statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

splunk> .conf24

Maximizing Splunk Core

Analyzing Splunk Searches Using audittrail and Native Splunk Telemetry

What’s new

Larger dimensions

All presentation slides are now 1920x1080 size to provide higher fidelity on larger monitors and projections.

New typeface

We’ve moved from Arial to Lexend which was designed to optimize legibility on screens. It’s closer to our Splunk Sans typecase and therefore is more ownable. It also has several font weights to choose from.

Clean and minimal

Our brand has been evolving to be simpler, cleaner, and have greater visual impact. This includes refreshed image options and bold color moments.

More titling and layout variety

Amazing presentations are built on great storytelling. The updated layouts will help you provide pacing and variety.

DO NOT UPDATE THEME

There are so many new features and major updates made with this presentation template that we do not recommend updating your old slides to this theme. Instead click "File" > "Make a copy" > "Entire presentation"

To replace content:

1. Copy content from an old presentation.
2. Open a new presentation from this theme.
3. Create a new slide layout.
4. Highlight a text box and click "edit" > "paste without formatting" to maintain the styles of the new theme.

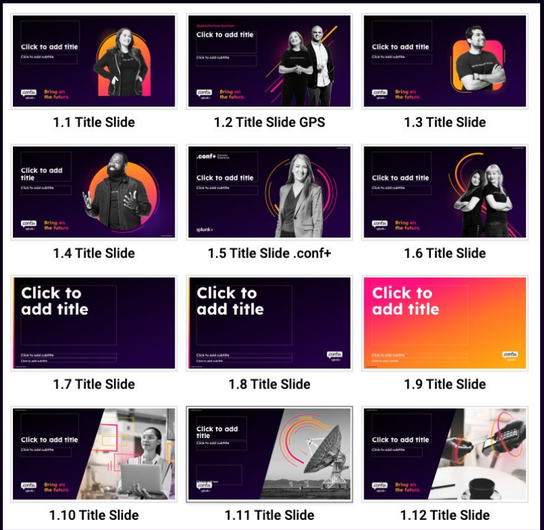
Frequently asked questions

For a full review of the changes, known issues, and how to troubleshoot problems see our Corporate Template FAQ

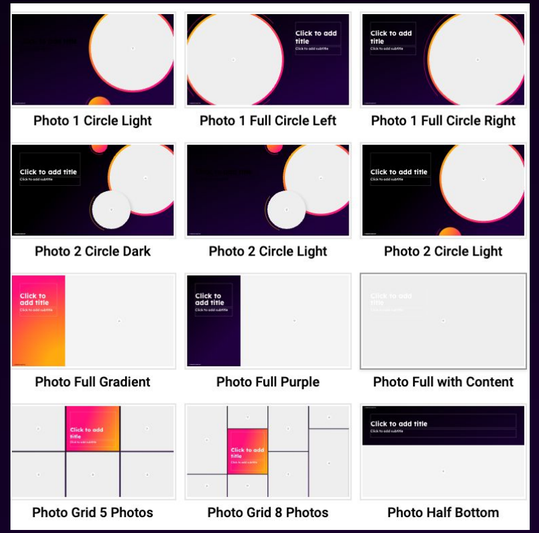
What's new

Layout options

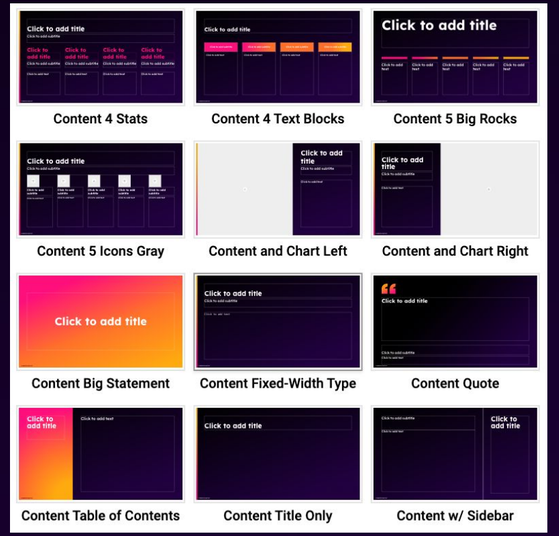
Title



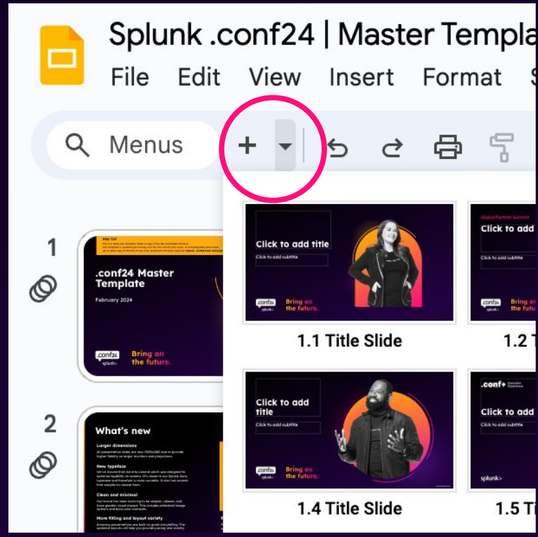
Photo



Content

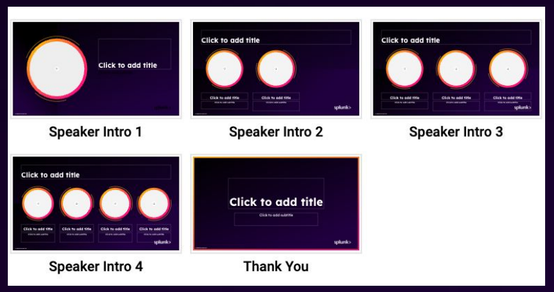


Access to Layouts



Click down arrow "New slide with layout" next to the "New slide" tab to view all available layouts.

Speaker



Only use approved artwork

Do not pull images from Google or off the internet. You may use **approved images provided by Brand and Creative** or from: **pexels.com**, **pixabay.com**, **burst.shopify.com** and **commons.wikimedia.org**.

.conf24 Do's

Questions? Contact marketinglegal@splunk.com

- Mark every slide deck with “© 2024 SPLUNK INC.”
- See the 'What's New Slide' for approved images from by Brand and Creative.
- Use externally referenceable customer stories
  - Note: Reference approval in a comment that you have received written permission from Customer Stories Team and Customer is aware.
- Be accurate when using descriptive words
  - Are we the leader, fastest, most comprehensive, only complete solution?
  - If using we need to be able to back up the claim with citation, i.e., Gartner or other industry analyst
- Use our trademark symbols when using our slogans or talking about products
  - Splunk® Enterprise
- Always add source footnotes when referencing TAM, numbers, percentages, or quotes
  - i.e. Where did we get the 80% number?
- Use externally referenceable timelines when sharing product roadmap details
  - Define timing as "Now", "Next" and "Future". They are meant to be super generic.
  - ALWAYS notify Rev Rec when discussing future functionality or roadmap details
  - If you need to disclose specific timing:
    - For on-prem roadmap, you can use either "Next 6-12 months" or "12+ months". Do not be more specific than this.
    - For cloud roadmap, can be as specific as the quarter (e.g., Q4 2023)
- Use the current version of the Forward Looking Statement if you are talking about products (Brand and Creative has the most recent comprehensive version) and receive clearance before discussing new features.
- Use pre-existing screenshots of actual product UX if we are simulating use cases within demos. Content should be free of confidential information, customer or company details, etc. For screenshots blur or redact.
- Make sure when making statements about partnerships that we do not mislead
  - i.e. Strategic partnerships, TAP or Partnerverse relationships

.conf24 Don'ts

Questions? Contact marketinglegal@splunk.com

- Use images/videos or other copyrighted materials from the internet
  - Screenshots taken from Social Media (Instagram posts, Twitter Tweets, Facebook posts, etc…)
- Use third party logos, trademarks and/or brands unless we have documented permission from the third party
  - *Note:* A third party in this context means any party that is not a Splunk employee or contractor. Partners and customers are considered to be third parties.
  - Please reference your documented permission to use a third party logo, trademark or brand in a comment to your deck in every instance where usage appears. Permission is sufficient if it clearly indicates our intended use is permissible (i.e. company has brand/trademark guidelines that provide for commercial use of their trademarks in marketing materials, company has a trademark usage agreement with Splunk, etc.)
- Disclose any Confidential Information (as defined in the Splunk Code of Conduct and Business Ethics)
- Include anything about deal financials
  - *Do not talk about our financial performance, in general or specifically*

Who are you?

Architect

Engineer

Admin

Manager

?

Someone who wants to…

- Identify what data sources are powering your searches
- Increase the value from your Splunk solutions
- Become familiar with OOTB Splunk internal logging
- ?

Base Knowledge Pre-Reqs

to get the most out of this presentation…

- Familiar with what a Splunk Search is
- You've seen some internal data sources
- ?

Why are you here?

To get some awesome insights into your Splunk environment!
Like "What data sources are my searches using?"

Objectives

It's our objective for you to leave this talk with...

A basic familiarity of the default Splunk internal data sources available in your environment

The knowledge of how to answer that key question - "What data sources are my searches using?"

References and resources that are available publicly, built by fellow Splunk users, to dive further into utilizing those internal data sources for adding value

Objectives

What are the goals that drove this presentation?

A bit of a surprise…

- In May 2023, `foreach` ranked #21 out of 147 commands in terms of usage *by users* across all Splunk Cloud™ stacks[1]
- While researching, I found less than 20 total example queries using `| foreach` publicly available across Github + Major Search Engines (first 3 pages)

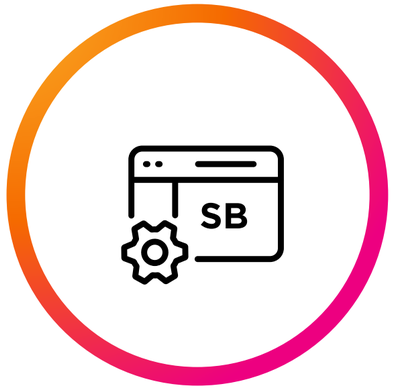
Introduce

Establish the behavior of `| foreach` & how to think about it's role.

Inspire

Walk through a series of methods utilizing `| foreach` in increasing complexity.

Enable



Provide examples and sample code to utilize in your own environment.

[1] Based on telemetry statistics provided by Splunk

Dashboard View, Teaser

Agenda

Split into two major groups

Introduction & Level Set

- Introductions
  - Background
  - Objectives
- Framing the scope of this presentation
  - Default Splunk Data sources

- Overview walkthrough?
  - Highlights of data sources
  - Examples
- Important considerations when using

Getting Technical

- Usecase Examples Walkthrough
  - Identifying problematic searches
  - How to combine search telemetry data
  - Data Source Usage

- Walkthrough of other insights we can gather from this data
- Stitching multiple internal data sources together to tell a story
- Wrap-up & Resources
  - Why reinvent the wheel?

Agenda

Another format

Overview of data sources

Walkthrough and examples

Necessary prerequisite pieces

"Let's get into the good stuff."

Get technical

Q&A

Who are we?

Ryan Wood

Sr. Splunk Solutions Architect
GuidePoint Security

Rich Galloway

Technical Account Manager
Splunk

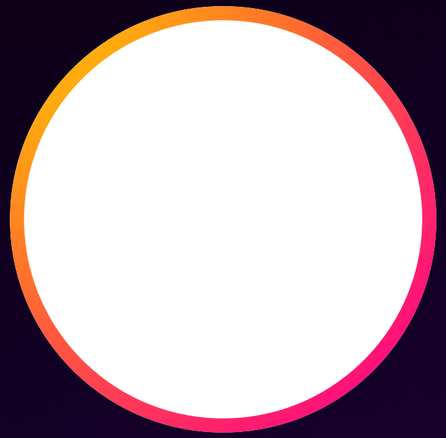
Rich Galloway

Background

- Splunk user since 2012
- SplunkTrust member since 2016
- Professional Services provider for 6 years
- Co-leader of the DC user group

"Wherever you go, there you are"

Ryan Wood



Background

- Certifications:
  - Splunk Core Certified Consultant
  - Splunk Enterprise Security Certified Admin
  - Security+
- Daily Splunk User for over 11 years
- Delivering Splunk Professional Services for over 6 years
- Conf 2023 Speaker
- Passionate about enabling others to find success

"Interesting quote"

Let's Get Into It

Defining the Scope of This Presentation

There's a *Lot* of Default Data Sources

Lorem ipsum dolor sit amet.

- Splunk does a lot, so there's a lot of logging about what it's doing.
- 8 underscore indexes by default as of v9.2
- 144 unique combinations of index + sourcetype found in our reference environments:
  - Enterprise v9.0.4
  - Enterprise v9.1.2
  - Enterprise v9.2.0
  - Cloud Victoria v9.1.2308
- Note: Add configtracker block

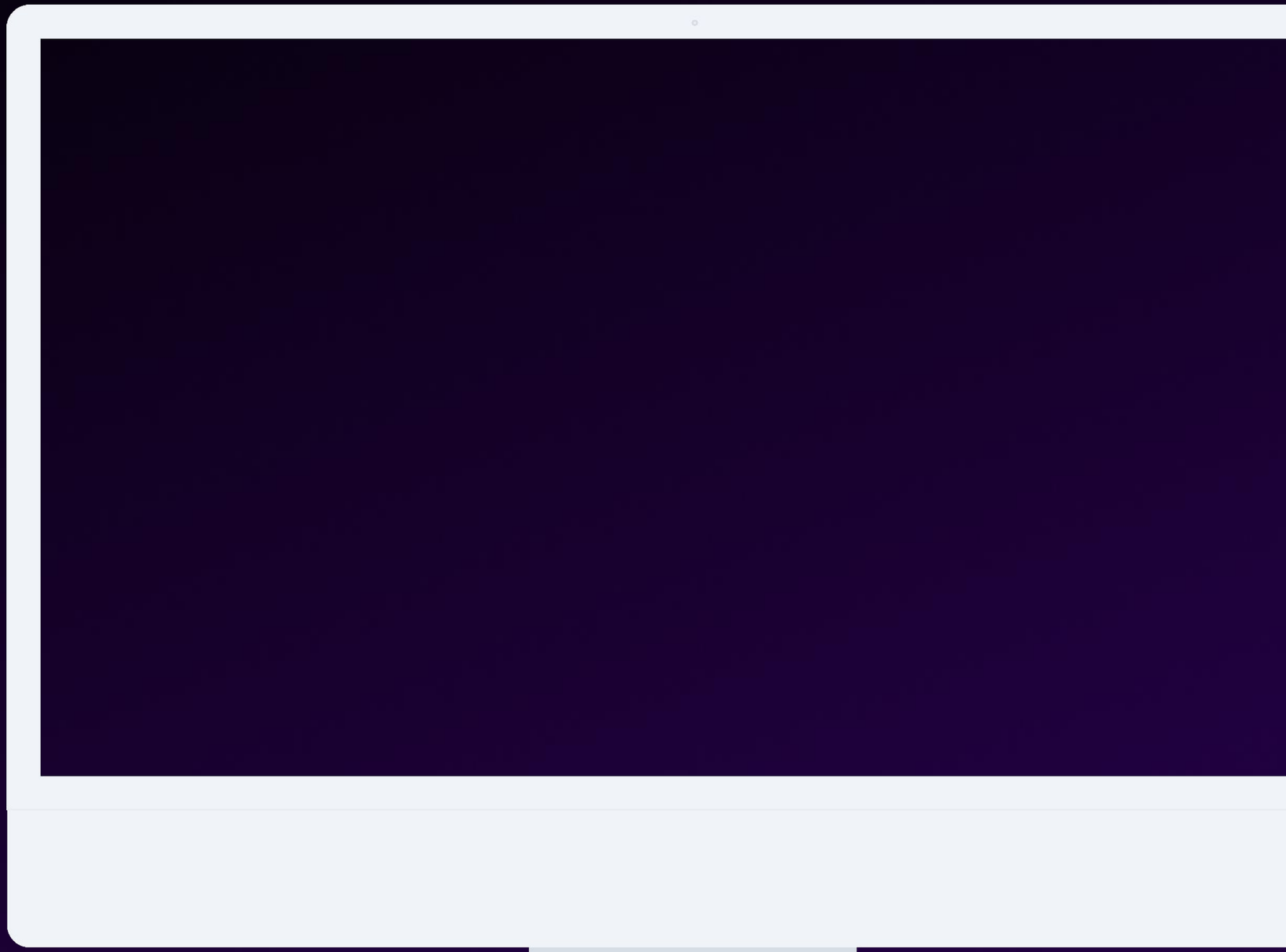
~_audit	~_dsappevent _dsclient _dsphonehome	~_internal	~_introspection	~_telemetry
audittrail	dsappevent	splunk_python	splunk_resource_usage	splunk_cloud_tel
incident_review	dsclient	splunkd	splunk_disk_objects	splunk_telemetry
	dsphonehome	splunkd_access	splunk_telemetry	splunk_telemetry
		splunkd_remote_searches	kvstore	splunkd
		splunk_btool	search_telemetry	
		scheduler	http_event_collector_metrics	
		splunkd_ui_access	scma:check	
		splunk_search_messages		
		splunk_web_service		
		splunk_assist_internal_log		
		splunk_web_access		
		splunk_secure_gateway.log		
		splunk_archiver		
		splunk_audit		
		splunk_instrumentation		
		splunkd_systemd_stdout		
		splunkd_conf		
		splunkd_stderr		
		audit_ingest.log		
		mongod		

~_configtracker
splunk_configuration_change

This is why we're here

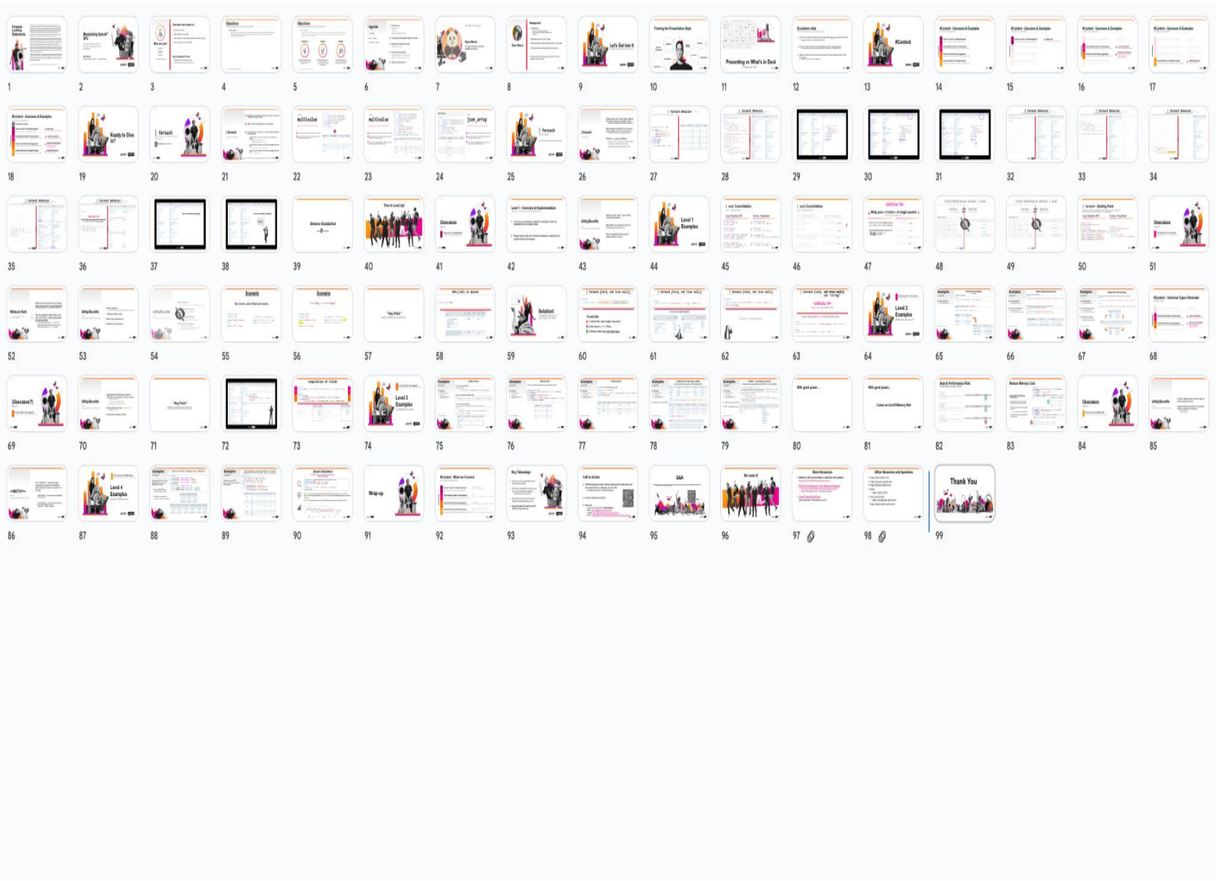
Data sources we're focusing on today

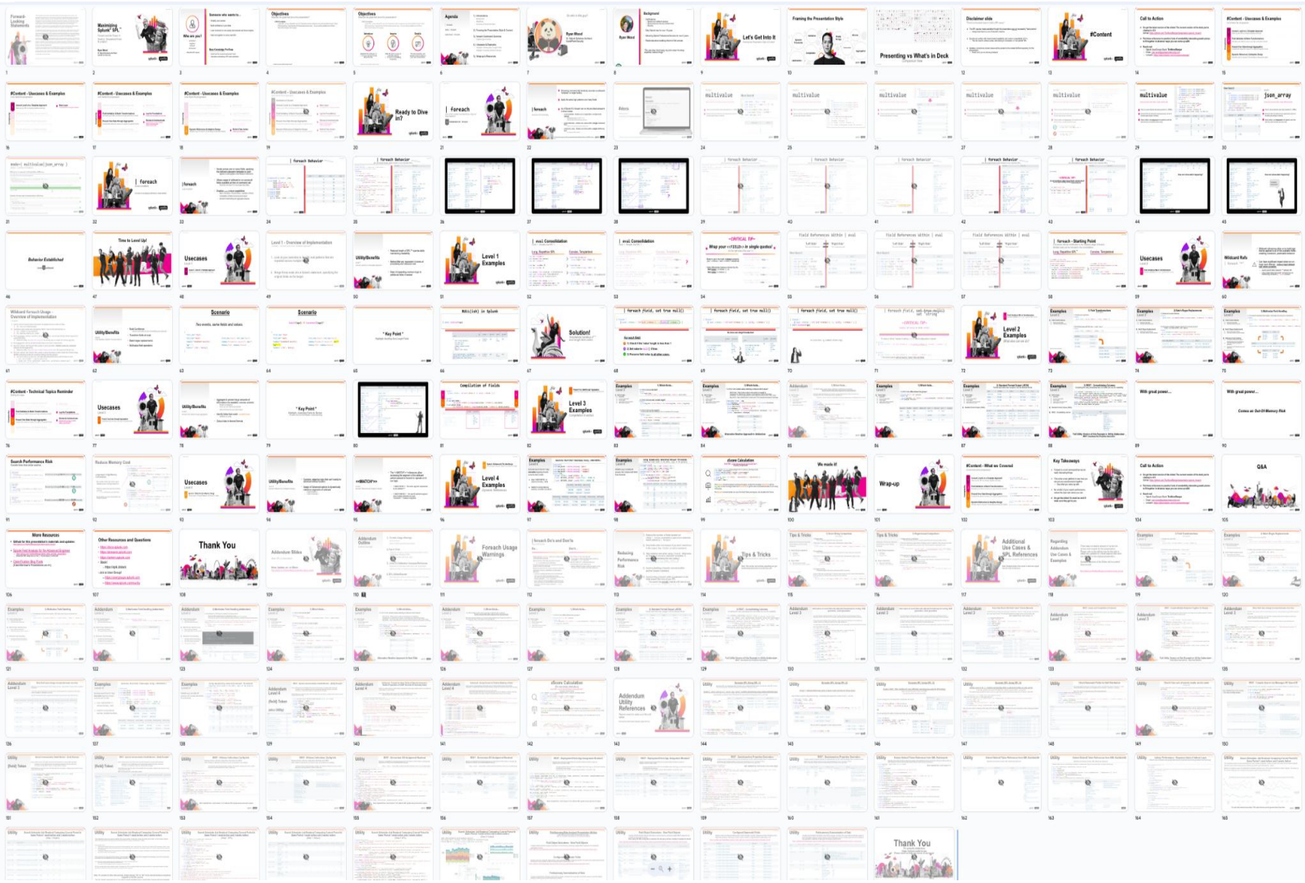
Ryan to add screenshot of filtered set similar to previous slide



Presenting vs What's in Deck

Comparison View





Presenting vs What's in Deck

Comparison View (v2.0)

splunk> .conf23

Placeholder - Technical Content

Placeholder - Marks when we move into Introducing the data sources relevant to this talk

Splunk Internal Indexes

An Overview.. (basic high level points here)

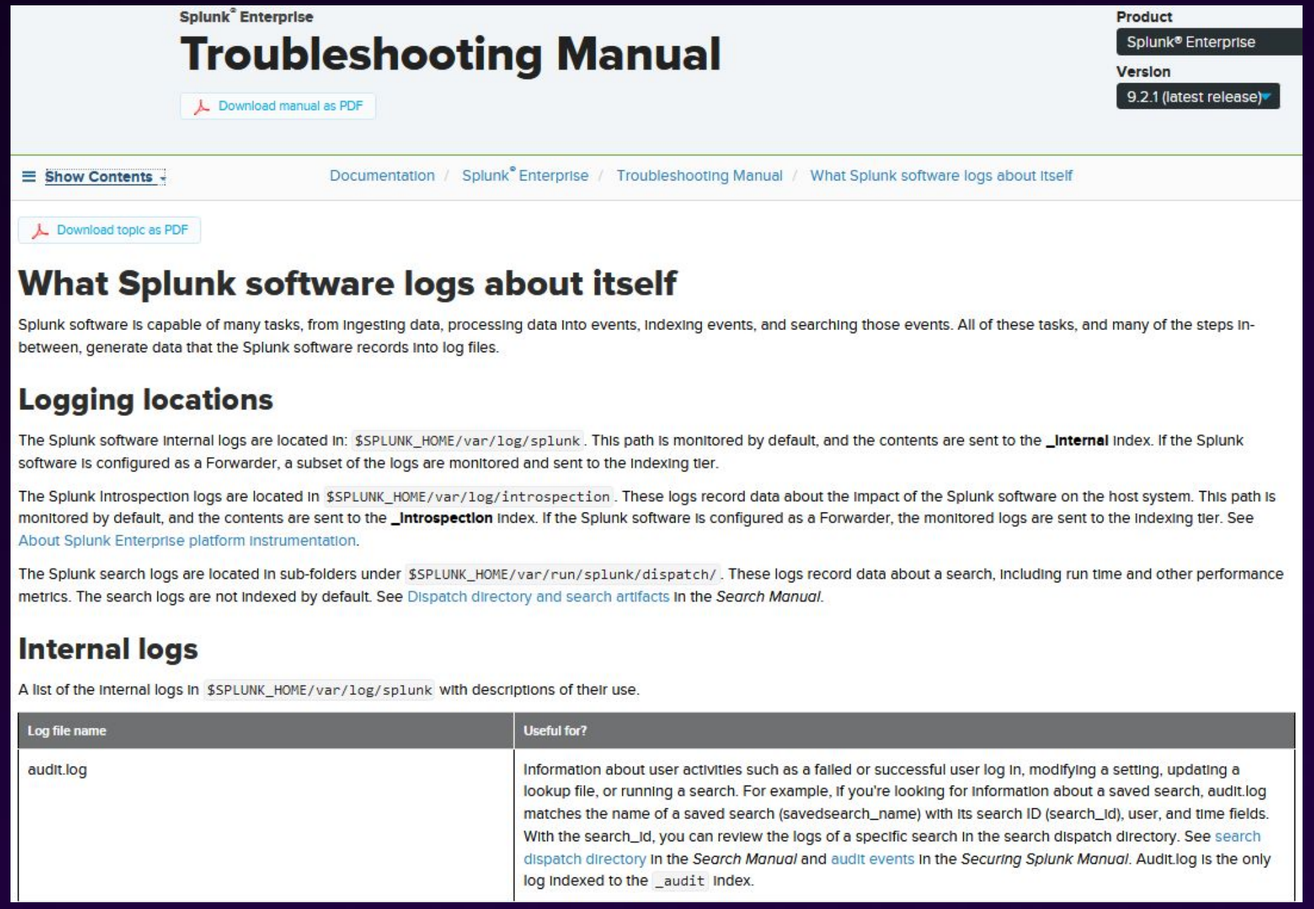
- _internal
  - Most internal logs
- _introspection
  - Resource Usage
- _audit
  - User activity, including search
- _telemetry
  - Collection of metrics related to Splunk…
- _configtracker
  - Changes to .conf files
- _ds(client|phonehome|appevent)
  - New in 9.2.0

("Secret" one - search.log)

index=_audit

The basics

- Information about user activity, including running searches
- Default retention: 6 years (3 years in Splunk Cloud)
- Used to populate the Search Usage Statistics and parts of the Expensive Searches dashboards in CMC
  - Probably others, too
- index = _audit source = *audittrail sourcetype = audittrail



Splunk® Enterprise

Troubleshooting Manual

Download manual as PDF

Product: Splunk® Enterprise

Version: 9.2.1 (latest release)

Show Contents | Documentation / Splunk® Enterprise / Troubleshooting Manual / What Splunk software logs about itself

Download topic as PDF

What Splunk software logs about itself

Splunk software is capable of many tasks, from ingesting data, processing data into events, indexing events, and searching those events. All of these tasks, and many of the steps in-between, generate data that the Splunk software records into log files.

Logging locations

The Splunk software internal logs are located in: `$SPLUNK_HOME/var/log/splunk`. This path is monitored by default, and the contents are sent to the **_internal** index. If the Splunk software is configured as a Forwarder, a subset of the logs are monitored and sent to the indexing tier.

The Splunk introspection logs are located in `$SPLUNK_HOME/var/log/introspection`. These logs record data about the impact of the Splunk software on the host system. This path is monitored by default, and the contents are sent to the **_introspection** index. If the Splunk software is configured as a Forwarder, the monitored logs are sent to the indexing tier. See About Splunk Enterprise platform instrumentation.

The Splunk search logs are located in sub-folders under `$SPLUNK_HOME/var/run/splunk/dispatch/`. These logs record data about a search, including run time and other performance metrics. The search logs are not indexed by default. See Dispatch directory and search artifacts in the *Search Manual*.

Internal logs

A list of the internal logs in `$SPLUNK_HOME/var/log/splunk` with descriptions of their use.

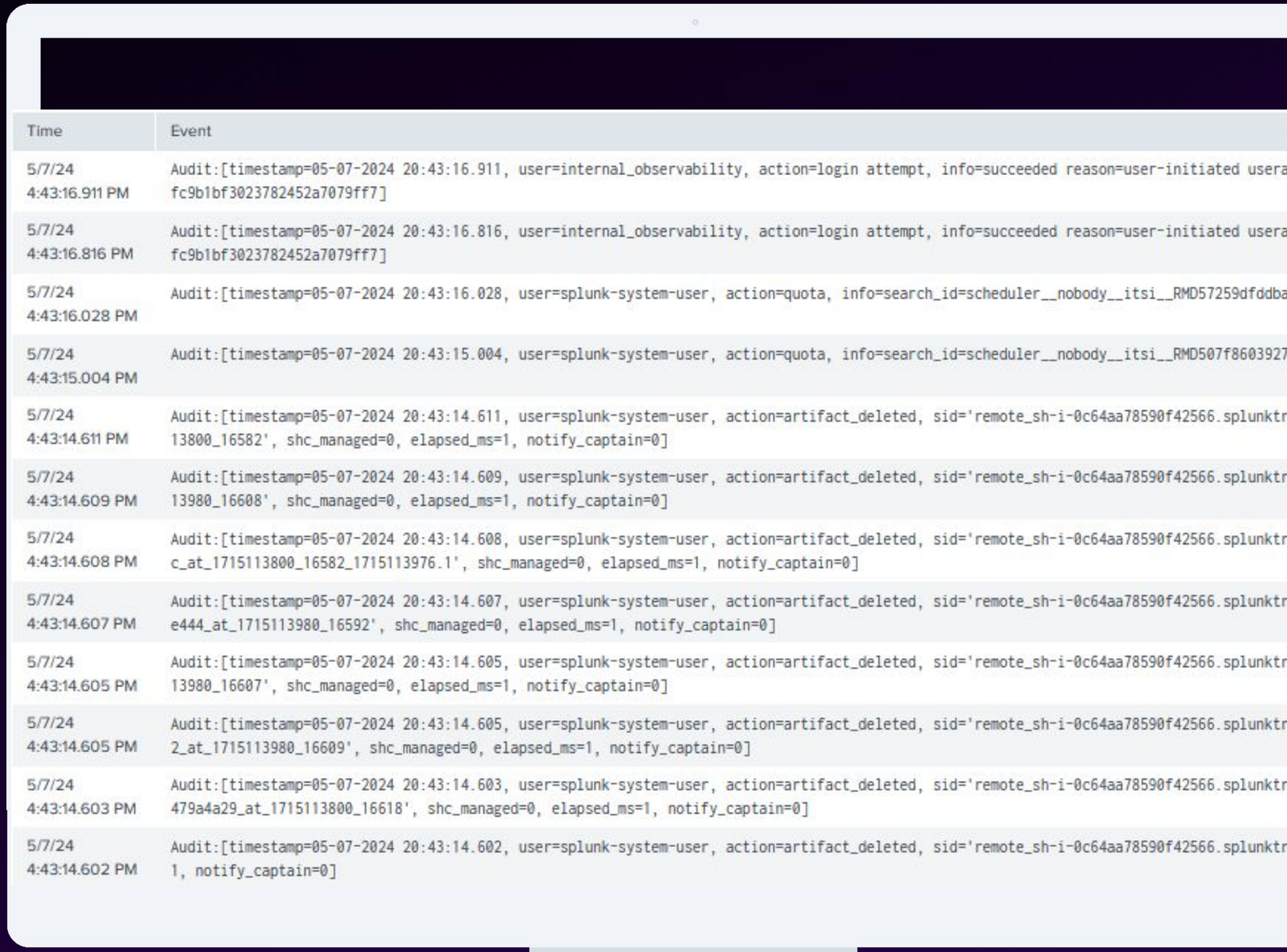
Log file name	Useful for?
audit.log	Information about user activities such as a failed or successful user log in, modifying a setting, updating a lookup file, or running a search. For example, if you're looking for information about a saved search, audit.log matches the name of a saved search (savedsearch_name) with its search ID (search_id), user, and time fields. With the search_id, you can review the logs of a specific search in the search dispatch directory. See search dispatch directory in the *Search Manual* and audit events in the *Securing Splunk Manual*. Audit.log is the only log indexed to the `_audit` index.

https://docs.splunk.com/Documentation/Splunk/9.2.1/Troubleshooting/WhatSplunklogsaboutitself#

audittrail

The basics

- Dozens of “actions” are logged
- Interesting actions
  - search
  - create_user
  - create_saved_search
  - alert_fired
  - login attempt [*sic*]
  - remote_bucket_download

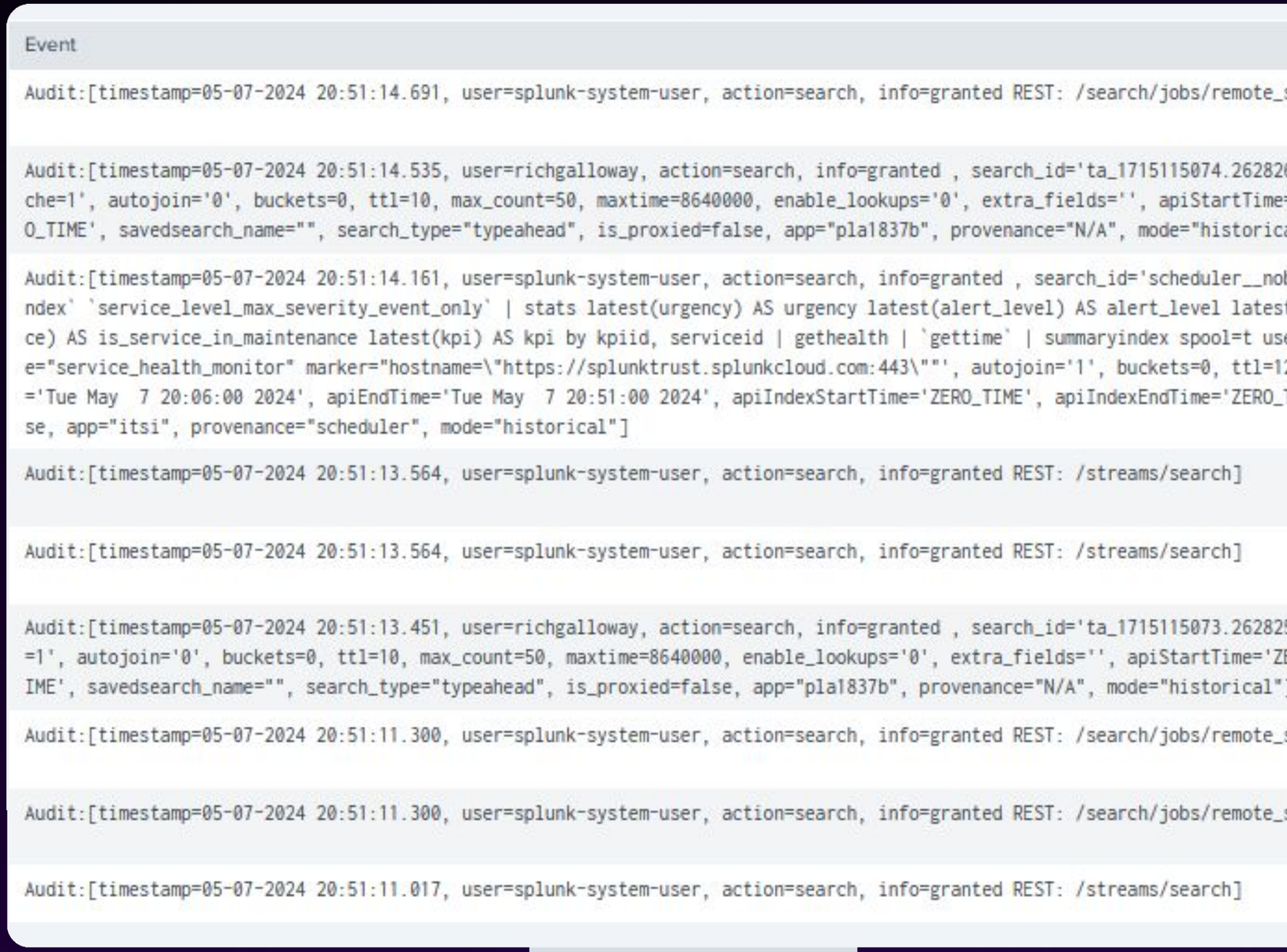


Time	Event
5/7/24 4:43:16.911 PM	Audit:[timestamp=05-07-2024 20:43:16.911, user=internal_observability, action=login attempt, info=succeeded reason=user-initiated usera fc9b1bf3023782452a7079ff7]
5/7/24 4:43:16.816 PM	Audit:[timestamp=05-07-2024 20:43:16.816, user=internal_observability, action=login attempt, info=succeeded reason=user-initiated usera fc9b1bf3023782452a7079ff7]
5/7/24 4:43:16.028 PM	Audit:[timestamp=05-07-2024 20:43:16.028, user=splunk-system-user, action=quota, info=search_id=scheduler__nobody__itsi__RMD57259dfddba
5/7/24 4:43:15.004 PM	Audit:[timestamp=05-07-2024 20:43:15.004, user=splunk-system-user, action=quota, info=search_id=scheduler__nobody__itsi__RMD507f8603927
5/7/24 4:43:14.611 PM	Audit:[timestamp=05-07-2024 20:43:14.611, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 13800_16582', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.609 PM	Audit:[timestamp=05-07-2024 20:43:14.609, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 13980_16608', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.608 PM	Audit:[timestamp=05-07-2024 20:43:14.608, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr c_at_1715113800_16582_1715113976.1', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.607 PM	Audit:[timestamp=05-07-2024 20:43:14.607, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr e444_at_1715113980_16592', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.605 PM	Audit:[timestamp=05-07-2024 20:43:14.605, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 13980_16607', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.605 PM	Audit:[timestamp=05-07-2024 20:43:14.605, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 2_at_1715113980_16609', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.603 PM	Audit:[timestamp=05-07-2024 20:43:14.603, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 479a4a29_at_1715113800_16618', shc_managed=0, elapsed_ms=1, notify_captain=0]
5/7/24 4:43:14.602 PM	Audit:[timestamp=05-07-2024 20:43:14.602, user=splunk-system-user, action=artifact_deleted, sid='remote_sh-i-0c64aa78590f42566.splunktr 1, notify_captain=0]

audittrail

The basics

- Dozens of “actions” are logged
- Interesting actions
  - **search**
  - create_user
  - create_saved_search
  - alert_fired
  - login attempt [*sic*]
  - remote_bucket_download



audittrail

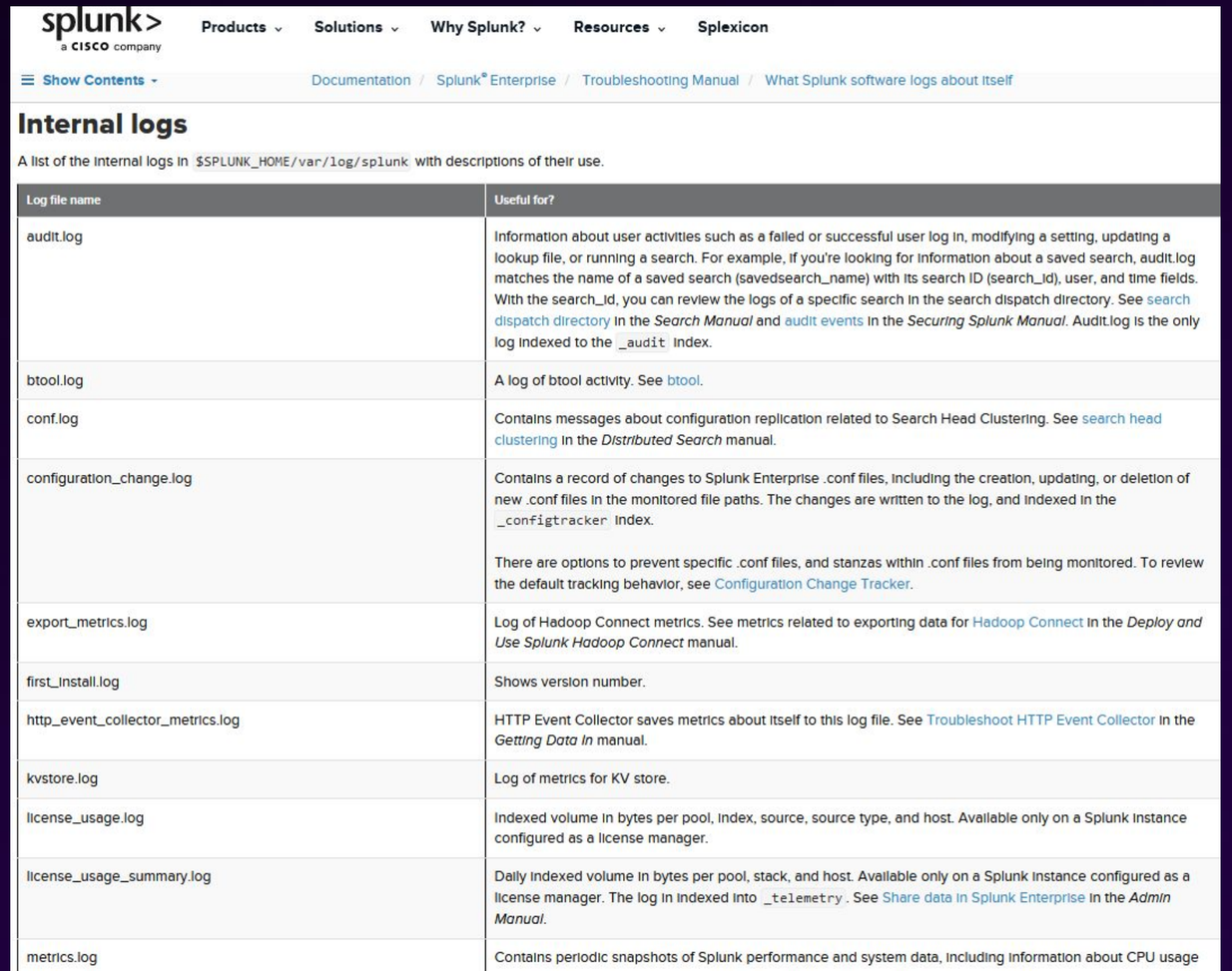
Sample search completed event

```
Audit:[timestamp=05-07-2024 20:59:46.266, user=mus, action=search, info=completed,
search_id='scheduler__mus_S2lhX09yYV8wMDE__RMD5f2f03341c2f3fd87_at_1715115420_20350', has_error_warn=true, fully_completed_search=true,
total_run_time=1.97, cpu_time=0.77, event_count=0, result_count=0, available_count=0, scan_count=0, drop_count=0, exec_time=1715115555, api_et=N/A,
api_lt=1715165820.000000000, api_index_et=N/A, api_index_lt=N/A, search_et=1715161980.000000000, search_lt=1715162160.000000000, is_realtime=0,
savedsearch_name="mus solar power to metric collector", search_startup_time="49", is_prjob=true, is_flex_search=false,
rate_limit_retry_enabled=false, dispatch_artifact_bytes=188416, status_csv_bytes=8192, is_fss3=false,
acceleration_id="C88762F5-BF4D-4B43-9537-DA44D898C9F2_Kia_Ora_001_mus_7c322824aedfb564", app="Kia_Ora_001", provenance="scheduler",
mode="historical_batch", is_proxied=false, searched_buckets=0, eliminated_buckets=0, considered_events=0, total_slices=0, decompressed_slices=0,
duration.command.search.index=0, invocations.command.search.index.bucketcache.hit=0, duration.command.search.index.bucketcache.hit=0,
invocations.command.search.index.bucketcache.miss=0, duration.command.search.index.bucketcache.miss=0,
invocations.command.search.index.bucketcache.error=0, duration.command.search.rawdata=0, invocations.command.search.rawdata.bucketcache.hit=0,
duration.command.search.rawdata.bucketcache.hit=0, invocations.command.search.rawdata.bucketcache.miss=0,
duration.command.search.rawdata.bucketcache.miss=0, invocations.command.search.rawdata.bucketcache.error=0, search_type=scheduled,
roles='itoa_admin+itoa_analyst+itoa_team_admin+itoa_user+metric_ad_admin+power+sc_admin+tokens_auth+user+user_ad_user', search='search index=mus
sourcetype="mus:pv" earliest=+13h-4m latest=+13h-1m
| timechart span=1min sum(gen) AS gen sum(cons) AS cons
| eval gen=gen/2, cons=cons/2, export=if(gen > cons, gen - cons, 0), import=if(cons > gen, cons - gen, 0)
| rename gen AS metric_name:gen, cons AS metric_name:cons, export AS metric_name:export, import AS metric_name:import
| mcollect index=mus_metric', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]
```

index=_internal

The basics

- Where most internal logs go
  - Except audit.log
- Default retention: 30 days
- 100+ sources
  - scheduler
  - splunkd.log
  - splunkd_access.log
  - python.log
  - metrics.log



splunk> a CISCO company | Products | Solutions | Why Splunk? | Resources | Splexicon

Show Contents | Documentation / Splunk® Enterprise / Troubleshooting Manual / What Splunk software logs about itself

Internal logs

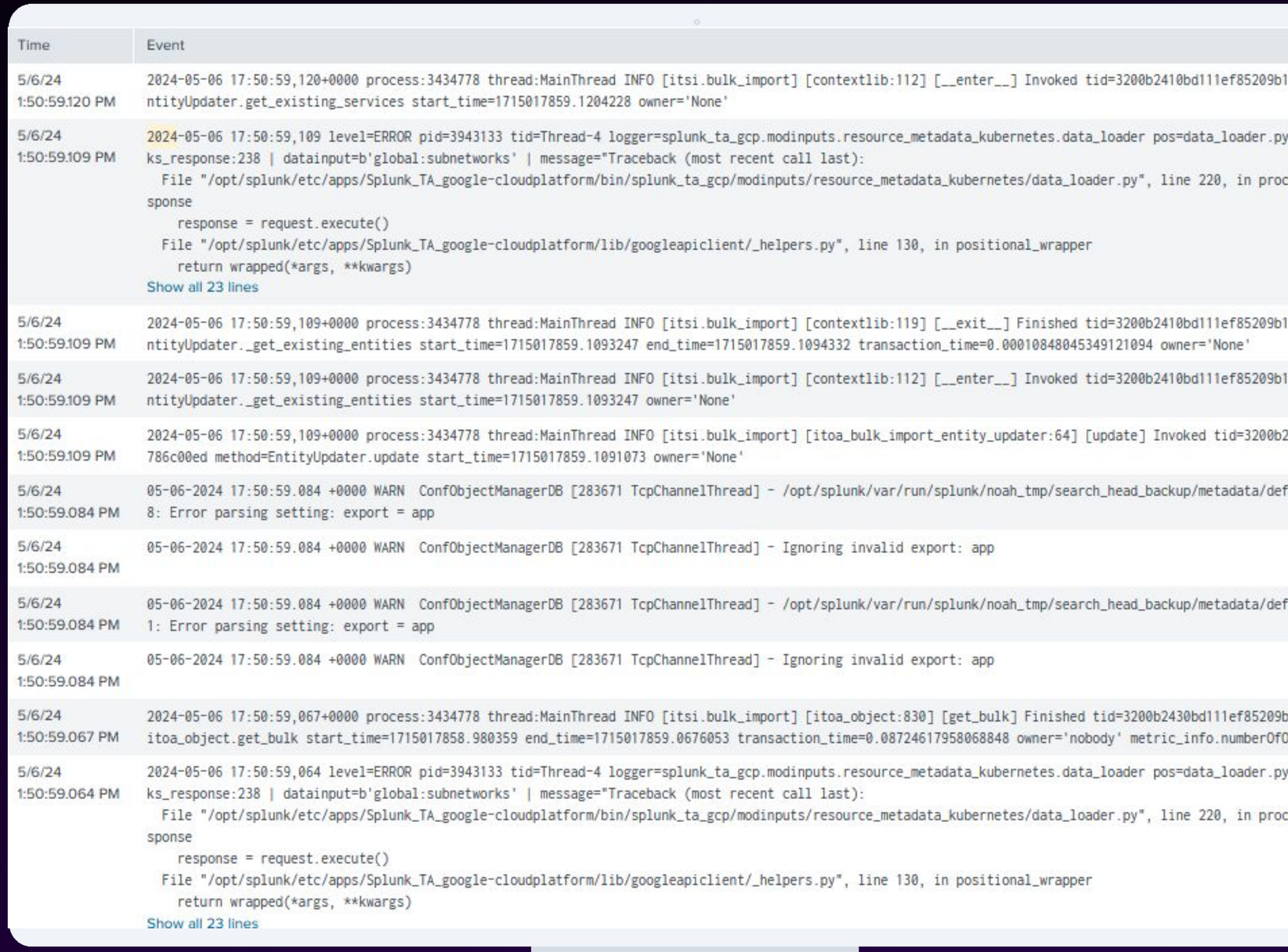
A list of the internal logs in `$SPLUNK_HOME/var/log/splunk` with descriptions of their use.

Log file name	Useful for?
audit.log	Information about user activities such as a failed or successful user log in, modifying a setting, updating a lookup file, or running a search. For example, if you're looking for information about a saved search, audit.log matches the name of a saved search (savedsearch_name) with its search ID (search_id), user, and time fields. With the search_id, you can review the logs of a specific search in the search dispatch directory. See search dispatch directory in the *Search Manual* and audit events in the *Securing Splunk Manual*. Audit.log is the only log indexed to the `_audit` index.
btool.log	A log of btool activity. See btool.
conf.log	Contains messages about configuration replication related to Search Head Clustering. See search head clustering in the *Distributed Search* manual.
configuration_change.log	Contains a record of changes to Splunk Enterprise .conf files, including the creation, updating, or deletion of new .conf files in the monitored file paths. The changes are written to the log, and indexed in the `_configtracker` index. There are options to prevent specific .conf files, and stanzas within .conf files from being monitored. To review the default tracking behavior, see Configuration Change Tracker.
export_metrics.log	Log of Hadoop Connect metrics. See metrics related to exporting data for Hadoop Connect in the *Deploy and Use Splunk Hadoop Connect* manual.
first_install.log	Shows version number.
http_event_collector_metrics.log	HTTP Event Collector saves metrics about itself to this log file. See Troubleshoot HTTP Event Collector in the *Getting Data In* manual.
kvstore.log	Log of metrics for KV store.
license_usage.log	Indexed volume in bytes per pool, index, source, source type, and host. Available only on a Splunk instance configured as a license manager.
license_usage_summary.log	Daily indexed volume in bytes per pool, stack, and host. Available only on a Splunk instance configured as a license manager. The log in indexed into `_telemetry`. See Share data in Splunk Enterprise in the *Admin Manual*.
metrics.log	Contains periodic snapshots of Splunk performance and system data, including information about CPU usage

_internal

The basics

- 80+ splunkd components
  - Metrics
  - SavedSplunker
  - HttpListener
  - PeriodicHealthReporter
  - ExecProcessor
  - TcpOutputProc (forwarders)
  - TcpInputProc
  - ModularInputs
  - UnionProcessor
  - sendmodalert

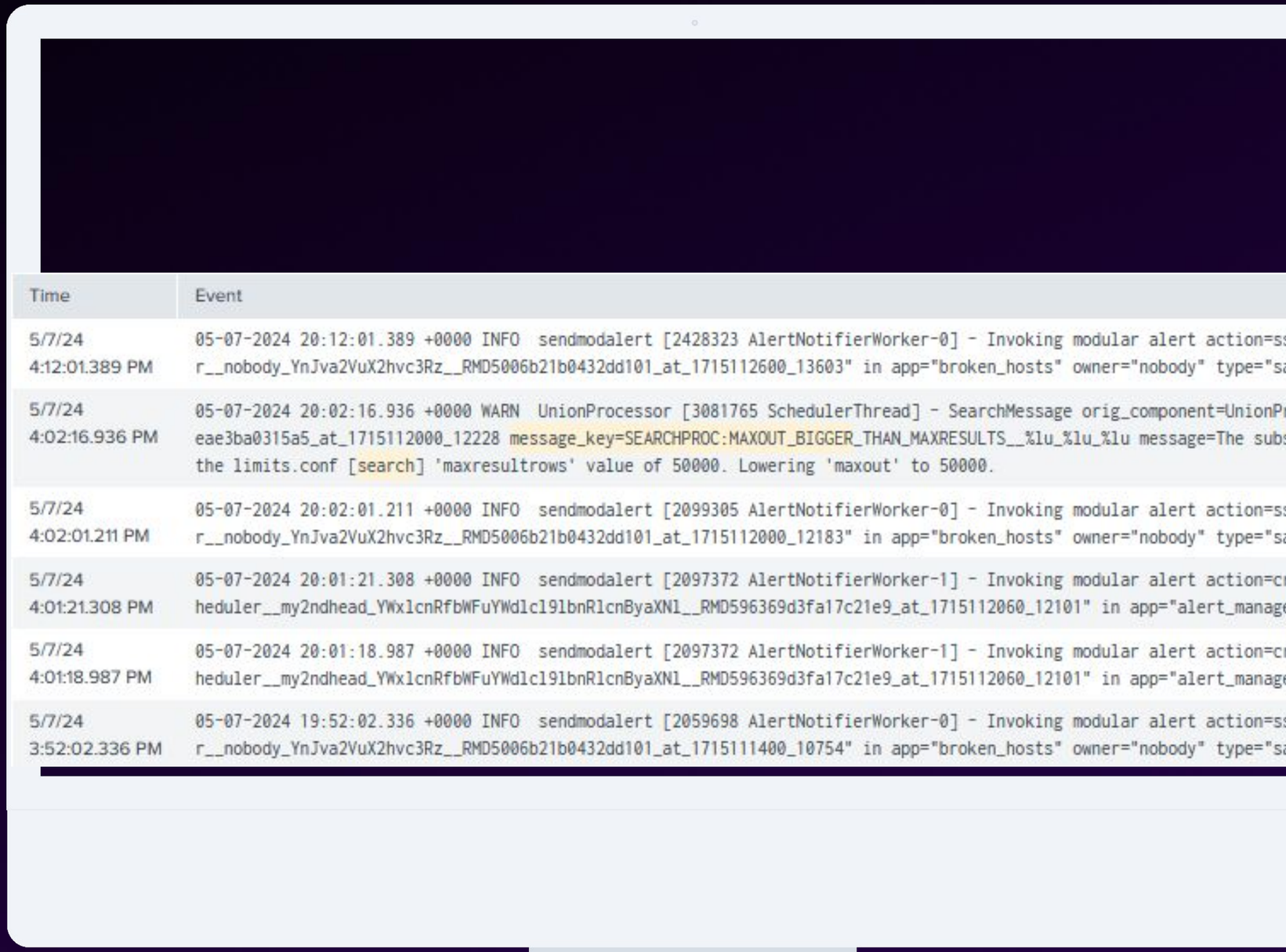


Time	Event
5/6/24 1:50:59.120 PM	2024-05-06 17:50:59,120+0000 process:3434778 thread:MainThread INFO [itsi.bulk_import] [contextlib:112] [__enter__] Invoked tid=3200b2410bd111ef85209b1 ntityUpdater.get_existing_services start_time=1715017859.1204228 owner='None'
5/6/24 1:50:59.109 PM	2024-05-06 17:50:59,109 level=ERROR pid=3943133 tid=Thread-4 logger=splunk_ta_gcp.modinputs.resource_metadata_kubernetes.data_loader pos=data_loader.py ks_response:238 | datainput=b'global:subnetworks' | message="Traceback (most recent call last): File "/opt/splunk/etc/apps/Splunk_TA_google-cloudplatform/bin/splunk_ta_gcp/modinputs/resource_metadata_kubernetes/data_loader.py", line 220, in proc sponse response = request.execute() File "/opt/splunk/etc/apps/Splunk_TA_google-cloudplatform/lib/googleapiclient/_helpers.py", line 130, in positional_wrapper return wrapped(*args, **kwargs) Show all 23 lines
5/6/24 1:50:59.109 PM	2024-05-06 17:50:59,109+0000 process:3434778 thread:MainThread INFO [itsi.bulk_import] [contextlib:119] [__exit__] Finished tid=3200b2410bd111ef85209b1 ntityUpdater._get_existing_entities start_time=1715017859.1093247 end_time=1715017859.1094332 transaction_time=0.00010848045349121094 owner='None'
5/6/24 1:50:59.109 PM	2024-05-06 17:50:59,109+0000 process:3434778 thread:MainThread INFO [itsi.bulk_import] [contextlib:112] [__enter__] Invoked tid=3200b2410bd111ef85209b1 ntityUpdater._get_existing_entities start_time=1715017859.1093247 owner='None'
5/6/24 1:50:59.109 PM	2024-05-06 17:50:59,109+0000 process:3434778 thread:MainThread INFO [itsi.bulk_import] [itoa_bulk_import_entity_updater:64] [update] Invoked tid=3200b2 786c00ed method=EntityUpdater.update start_time=1715017859.1091073 owner='None'
5/6/24 1:50:59.084 PM	05-06-2024 17:50:59.084 +0000 WARN ConfObjectManagerDB [283671 TcpChannelThread] - /opt/splunk/var/run/splunk/noah_tmp/search_head_backup/metadata/def 8: Error parsing setting: export = app
5/6/24 1:50:59.084 PM	05-06-2024 17:50:59.084 +0000 WARN ConfObjectManagerDB [283671 TcpChannelThread] - Ignoring invalid export: app
5/6/24 1:50:59.084 PM	05-06-2024 17:50:59.084 +0000 WARN ConfObjectManagerDB [283671 TcpChannelThread] - /opt/splunk/var/run/splunk/noah_tmp/search_head_backup/metadata/def 1: Error parsing setting: export = app
5/6/24 1:50:59.084 PM	05-06-2024 17:50:59.084 +0000 WARN ConfObjectManagerDB [283671 TcpChannelThread] - Ignoring invalid export: app
5/6/24 1:50:59.067 PM	2024-05-06 17:50:59,067+0000 process:3434778 thread:MainThread INFO [itsi.bulk_import] [itoa_object:830] [get_bulk] Finished tid=3200b2430bd111ef85209b itoa_object.get_bulk start_time=1715017858.980359 end_time=1715017859.0676053 transaction_time=0.08724617958068848 owner='nobody' metric_info.numberOfO
5/6/24 1:50:59.064 PM	2024-05-06 17:50:59,064 level=ERROR pid=3943133 tid=Thread-4 logger=splunk_ta_gcp.modinputs.resource_metadata_kubernetes.data_loader pos=data_loader.py ks_response:238 | datainput=b'global:subnetworks' | message="Traceback (most recent call last): File "/opt/splunk/etc/apps/Splunk_TA_google-cloudplatform/bin/splunk_ta_gcp/modinputs/resource_metadata_kubernetes/data_loader.py", line 220, in proc sponse response = request.execute() File "/opt/splunk/etc/apps/Splunk_TA_google-cloudplatform/lib/googleapiclient/_helpers.py", line 130, in positional_wrapper return wrapped(*args, **kwargs) Show all 23 lines

_internal

The basics

- splunkd components
  - UnionProcessor
    - Reports on subsearches that are too big
    - other uses unknown
  - sendmodalert
    - Modular alert runs

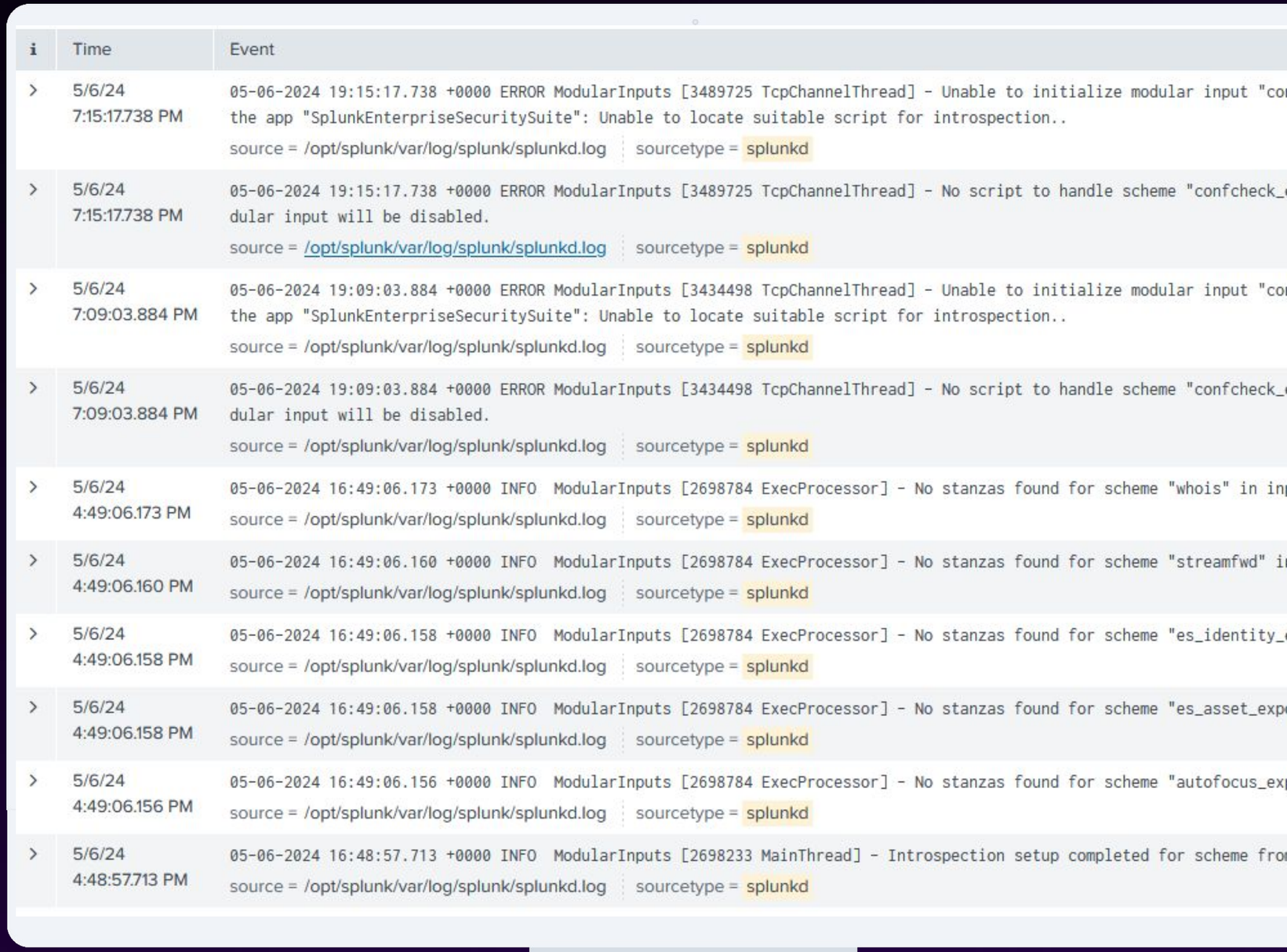


Time	Event
5/7/24 4:12:01.389 PM	05-07-2024 20:12:01.389 +0000 INFO sendmodalert [2428323 AlertNotifierWorker-0] - Invoking modular alert action=ss r__nobody_YnJva2VuX2hvc3Rz__RMD5006b21b0432dd101_at_1715112600_13603" in app="broken_hosts" owner="nobody" type="sa
5/7/24 4:02:16.936 PM	05-07-2024 20:02:16.936 +0000 WARN UnionProcessor [3081765 SchedulerThread] - SearchMessage orig_component=UnionPr eae3ba0315a5_at_1715112000_12228 message_key=SEARCHPROC:MAXOUT_BIGGER_THAN_MAXRESULTS__%lu_%lu_%lu message=The subs the limits.conf [search] 'maxresultrows' value of 50000. Lowering 'maxout' to 50000.
5/7/24 4:02:01.211 PM	05-07-2024 20:02:01.211 +0000 INFO sendmodalert [2099305 AlertNotifierWorker-0] - Invoking modular alert action=ss r__nobody_YnJva2VuX2hvc3Rz__RMD5006b21b0432dd101_at_1715112000_12183" in app="broken_hosts" owner="nobody" type="sa
5/7/24 4:01:21.308 PM	05-07-2024 20:01:21.308 +0000 INFO sendmodalert [2097372 AlertNotifierWorker-1] - Invoking modular alert action=cr heduler__my2ndhead_YWxlcnRfbWFuYWdlcl9lbnRlcnByaXNl__RMD596369d3fa17c21e9_at_1715112060_12101" in app="alert_manage
5/7/24 4:01:18.987 PM	05-07-2024 20:01:18.987 +0000 INFO sendmodalert [2097372 AlertNotifierWorker-1] - Invoking modular alert action=cr heduler__my2ndhead_YWxlcnRfbWFuYWdlcl9lbnRlcnByaXNl__RMD596369d3fa17c21e9_at_1715112060_12101" in app="alert_manage
5/7/24 3:52:02.336 PM	05-07-2024 19:52:02.336 +0000 INFO sendmodalert [2059698 AlertNotifierWorker-0] - Invoking modular alert action=ss r__nobody_YnJva2VuX2hvc3Rz__RMD5006b21b0432dd101_at_1715111400_10754" in app="broken_hosts" owner="nobody" type="sa

_internal

The basics

- modularInputs
  - Messages from modular inputs
  - Errors usually mean the MI isn't running and needs attention
  - index = _internal source = *splunkd.log sourcetype = splunkd component = ModularInputs

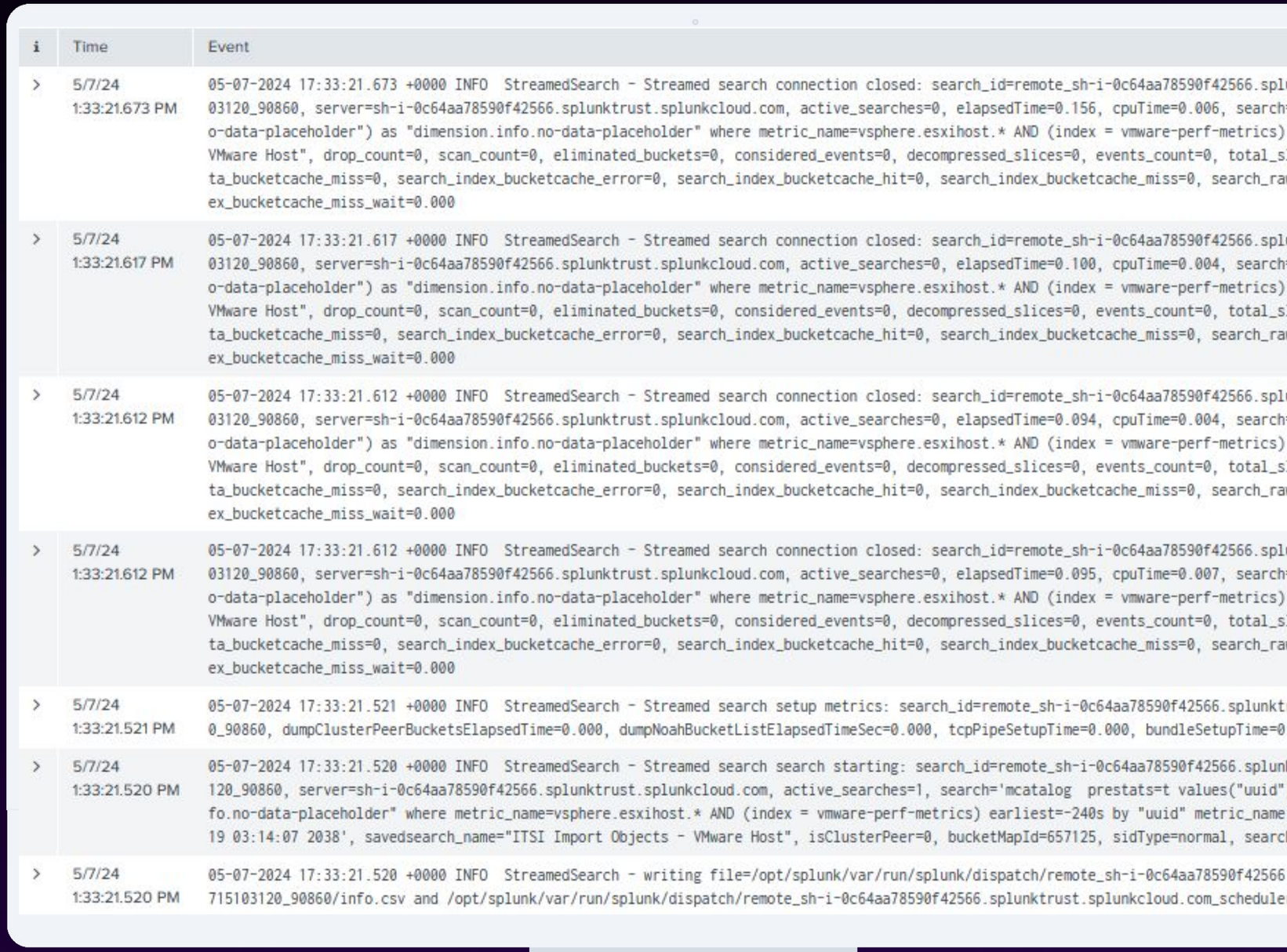


i	Time	Event
>	5/6/24 7:15:17.738 PM	05-06-2024 19:15:17.738 +0000 ERROR ModularInputs [3489725 TcpChannelThread] - Unable to initialize modular input "co the app "SplunkEnterpriseSecuritySuite": Unable to locate suitable script for introspection.. source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 7:15:17.738 PM	05-06-2024 19:15:17.738 +0000 ERROR ModularInputs [3489725 TcpChannelThread] - No script to handle scheme "confcheck_ dular input will be disabled. source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 7:09:03.884 PM	05-06-2024 19:09:03.884 +0000 ERROR ModularInputs [3434498 TcpChannelThread] - Unable to initialize modular input "co the app "SplunkEnterpriseSecuritySuite": Unable to locate suitable script for introspection.. source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 7:09:03.884 PM	05-06-2024 19:09:03.884 +0000 ERROR ModularInputs [3434498 TcpChannelThread] - No script to handle scheme "confcheck_ dular input will be disabled. source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:49:06.173 PM	05-06-2024 16:49:06.173 +0000 INFO ModularInputs [2698784 ExecProcessor] - No stanzas found for scheme "whois" in inp source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:49:06.160 PM	05-06-2024 16:49:06.160 +0000 INFO ModularInputs [2698784 ExecProcessor] - No stanzas found for scheme "streamfwd" i source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:49:06.158 PM	05-06-2024 16:49:06.158 +0000 INFO ModularInputs [2698784 ExecProcessor] - No stanzas found for scheme "es_identity_ source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:49:06.158 PM	05-06-2024 16:49:06.158 +0000 INFO ModularInputs [2698784 ExecProcessor] - No stanzas found for scheme "es_asset_exp source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:49:06.156 PM	05-06-2024 16:49:06.156 +0000 INFO ModularInputs [2698784 ExecProcessor] - No stanzas found for scheme "autofocus_ex source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
>	5/6/24 4:48:57.713 PM	05-06-2024 16:48:57.713 +0000 INFO ModularInputs [2698233 MainThread] - Introspection setup completed for scheme fro source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd

_internal

The basics

- remote_search.log
  - Streamed search activity between SH and indexers
  - index = _internal source = *remote_searches.log sourcetype = splunkd_remote_searches

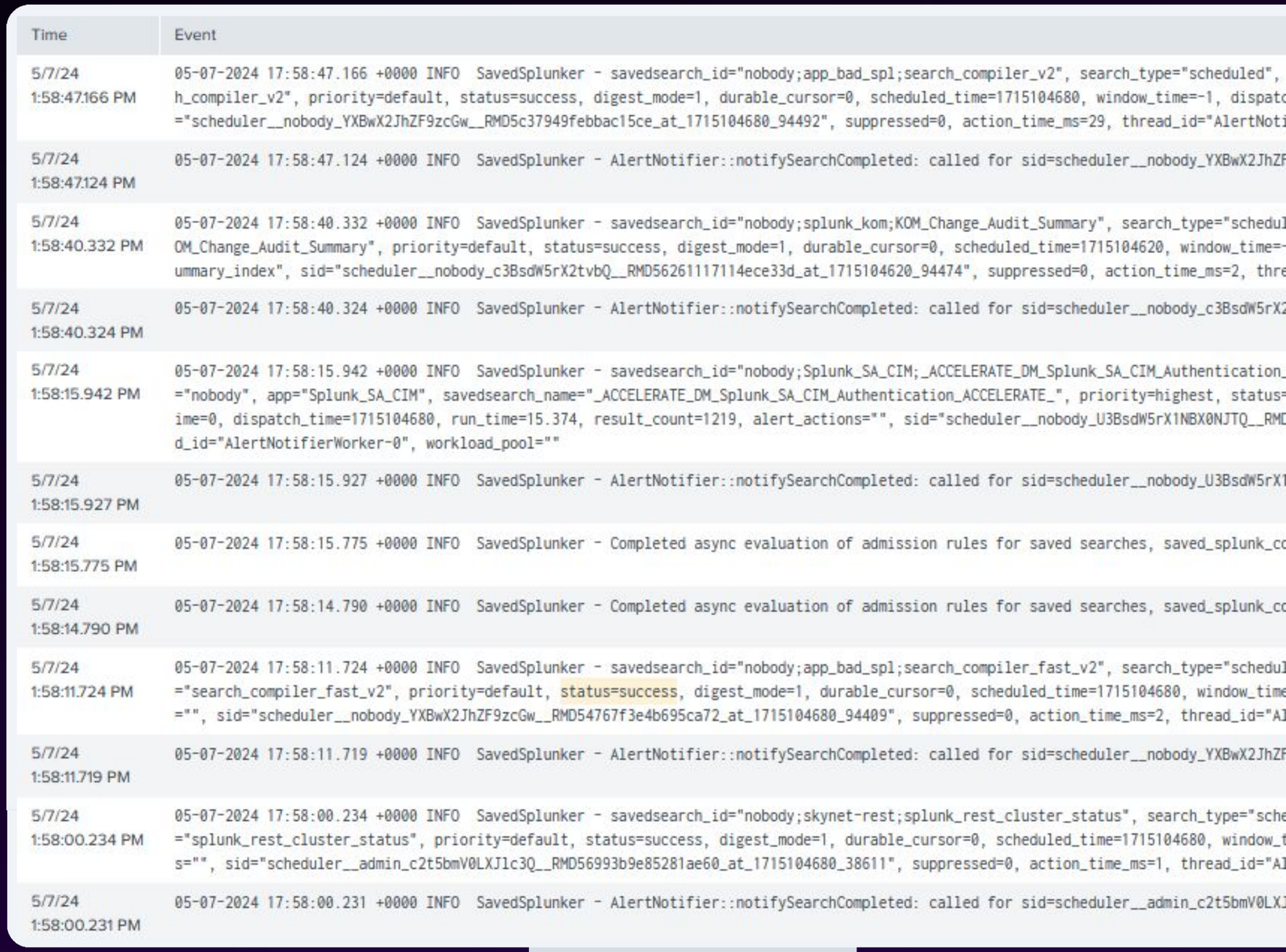


i	Time	Event
>	5/7/24 1:33:21.673 PM	05-07-2024 17:33:21.673 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-0c64aa78590f42566.splu 03120_90860, server=sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com, active_searches=0, elapsedTime=0.156, cpuTime=0.006, search o-data-placeholder") as "dimension.info.no-data-placeholder" where metric_name=vsphere.esxihost.* AND (index = vmware-perf-metrics) VMware Host", drop_count=0, scan_count=0, eliminated_buckets=0, considered_events=0, decompressed_slices=0, events_count=0, total_s ta_bucketcache_miss=0, search_index_bucketcache_error=0, search_index_bucketcache_hit=0, search_index_bucketcache_miss=0, search_ra ex_bucketcache_miss_wait=0.000
>	5/7/24 1:33:21.617 PM	05-07-2024 17:33:21.617 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-0c64aa78590f42566.splu 03120_90860, server=sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com, active_searches=0, elapsedTime=0.100, cpuTime=0.004, search o-data-placeholder") as "dimension.info.no-data-placeholder" where metric_name=vsphere.esxihost.* AND (index = vmware-perf-metrics) VMware Host", drop_count=0, scan_count=0, eliminated_buckets=0, considered_events=0, decompressed_slices=0, events_count=0, total_s ta_bucketcache_miss=0, search_index_bucketcache_error=0, search_index_bucketcache_hit=0, search_index_bucketcache_miss=0, search_ra ex_bucketcache_miss_wait=0.000
>	5/7/24 1:33:21.612 PM	05-07-2024 17:33:21.612 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-0c64aa78590f42566.splu 03120_90860, server=sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com, active_searches=0, elapsedTime=0.094, cpuTime=0.004, search o-data-placeholder") as "dimension.info.no-data-placeholder" where metric_name=vsphere.esxihost.* AND (index = vmware-perf-metrics) VMware Host", drop_count=0, scan_count=0, eliminated_buckets=0, considered_events=0, decompressed_slices=0, events_count=0, total_s ta_bucketcache_miss=0, search_index_bucketcache_error=0, search_index_bucketcache_hit=0, search_index_bucketcache_miss=0, search_ra ex_bucketcache_miss_wait=0.000
>	5/7/24 1:33:21.612 PM	05-07-2024 17:33:21.612 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-0c64aa78590f42566.splu 03120_90860, server=sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com, active_searches=0, elapsedTime=0.095, cpuTime=0.007, search o-data-placeholder") as "dimension.info.no-data-placeholder" where metric_name=vsphere.esxihost.* AND (index = vmware-perf-metrics) VMware Host", drop_count=0, scan_count=0, eliminated_buckets=0, considered_events=0, decompressed_slices=0, events_count=0, total_s ta_bucketcache_miss=0, search_index_bucketcache_error=0, search_index_bucketcache_hit=0, search_index_bucketcache_miss=0, search_ra ex_bucketcache_miss_wait=0.000
>	5/7/24 1:33:21.521 PM	05-07-2024 17:33:21.521 +0000 INFO StreamedSearch - Streamed search setup metrics: search_id=remote_sh-i-0c64aa78590f42566.splunkt 0_90860, dumpClusterPeerBucketsElapsedTime=0.000, dumpNoahBucketListElapsedTimeSec=0.000, tcpPipeSetupTime=0.000, bundleSetupTime=0
>	5/7/24 1:33:21.520 PM	05-07-2024 17:33:21.520 +0000 INFO StreamedSearch - Streamed search search starting: search_id=remote_sh-i-0c64aa78590f42566.splun 120_90860, server=sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com, active_searches=1, search='mcatalog prestats=t values("uuid" fo.no-data-placeholder" where metric_name=vsphere.esxihost.* AND (index = vmware-perf-metrics) earliest=-240s by "uuid" metric_name 19 03:14:07 2038', savedsearch_name="ITSI Import Objects - VMware Host", isClusterPeer=0, bucketMapId=657125, sidType=normal, searc
>	5/7/24 1:33:21.520 PM	05-07-2024 17:33:21.520 +0000 INFO StreamedSearch - writing file=/opt/splunk/var/run/splunk/dispatch/remote_sh-i-0c64aa78590f42566 715103120_90860/info.csv and /opt/splunk/var/run/splunk/dispatch/remote_sh-i-0c64aa78590f42566.splunktrust.splunkcloud.com_schedule

_internal

The basics

- scheduler.log
  - Scheduled search activity
    - Includes alerts and datamodel accelerations
  - Used to populate Scheduler Activity, Skipped Searches panels in CMC
  - index = _internal source = *scheduler.log sourcetype = scheduler

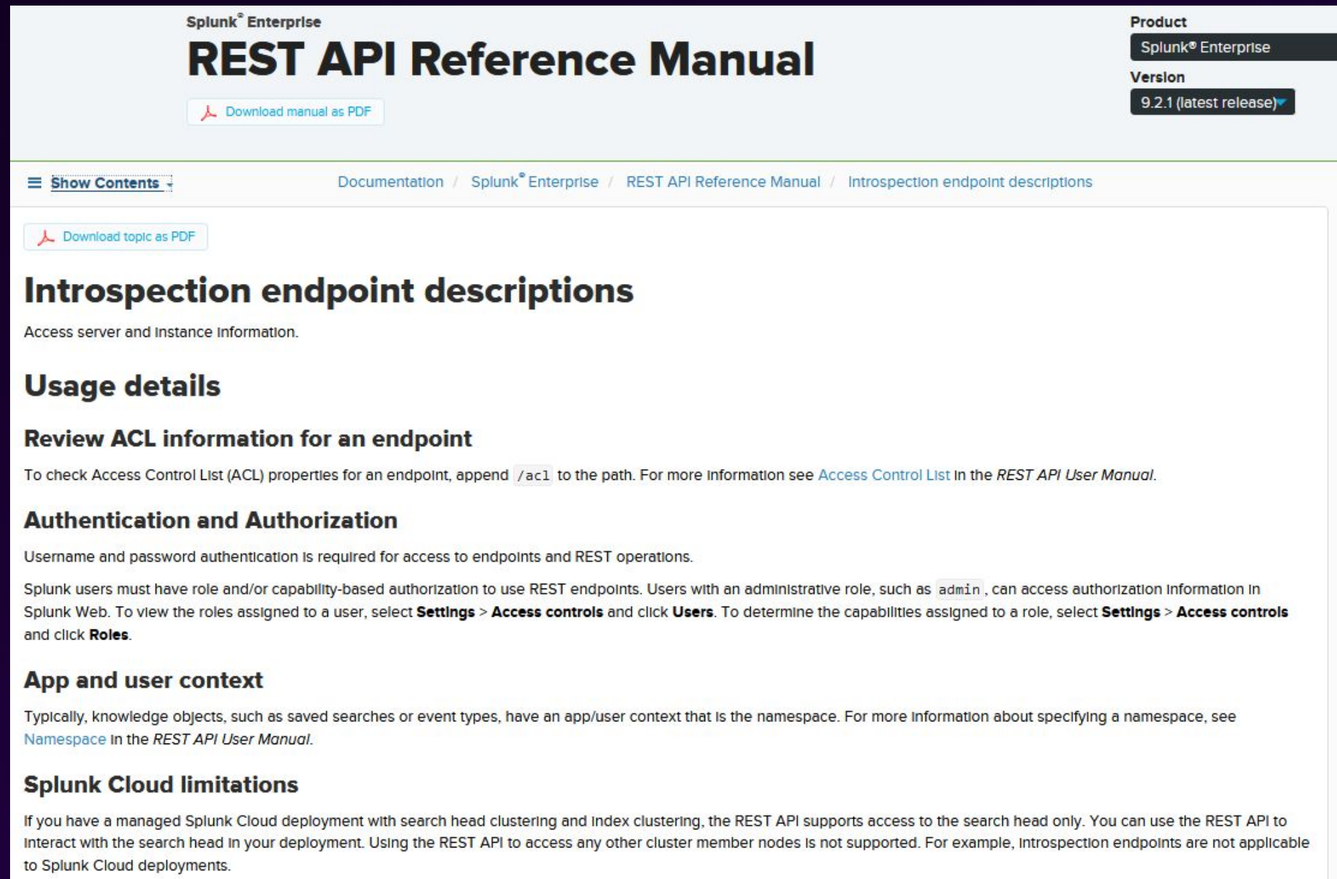


Time	Event
5/7/24 1:58:47.166 PM	05-07-2024 17:58:47.166 +0000 INFO SavedSplunker - savedsearch_id="nobody;app_bad_spl;search_compiler_v2", search_type="scheduled", h_compiler_v2", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715104680, window_time=-1, dispatc ="scheduler__nobody_YXBwX2JhZF9zcGw__RMD5c37949febbac15ce_at_1715104680_94492", suppressed=0, action_time_ms=29, thread_id="AlertNoti
5/7/24 1:58:47.124 PM	05-07-2024 17:58:47.124 +0000 INFO SavedSplunker - AlertNotifier::notifySearchCompleted: called for sid=scheduler__nobody_YXBwX2JhZF
5/7/24 1:58:40.332 PM	05-07-2024 17:58:40.332 +0000 INFO SavedSplunker - savedsearch_id="nobody;splunk_kom;KOM_Change_Audit_Summary", search_type="schedul OM_Change_Audit_Summary", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715104620, window_time= ummary_index", sid="scheduler__nobody_c3BsdW5rX2tvbQ__RMD56261117114ece33d_at_1715104620_94474", suppressed=0, action_time_ms=2, thre
5/7/24 1:58:40.324 PM	05-07-2024 17:58:40.324 +0000 INFO SavedSplunker - AlertNotifier::notifySearchCompleted: called for sid=scheduler__nobody_c3BsdW5rX2
5/7/24 1:58:15.942 PM	05-07-2024 17:58:15.942 +0000 INFO SavedSplunker - savedsearch_id="nobody;Splunk_SA_CIM;_ACCELERATE_DM_Splunk_SA_CIM_Authentication_ ="nobody", app="Splunk_SA_CIM", savedsearch_name="_ACCELERATE_DM_Splunk_SA_CIM_Authentication_ACCELERATE_", priority=highest, status= ime=0, dispatch_time=1715104680, run_time=15.374, result_count=1219, alert_actions="", sid="scheduler__nobody_U3BsdW5rX1NBX0NJTQ__RMD d_id="AlertNotifierWorker-0", workload_pool=""
5/7/24 1:58:15.927 PM	05-07-2024 17:58:15.927 +0000 INFO SavedSplunker - AlertNotifier::notifySearchCompleted: called for sid=scheduler__nobody_U3BsdW5rX1
5/7/24 1:58:15.775 PM	05-07-2024 17:58:15.775 +0000 INFO SavedSplunker - Completed async evaluation of admission rules for saved searches, saved_splunk_co
5/7/24 1:58:14.790 PM	05-07-2024 17:58:14.790 +0000 INFO SavedSplunker - Completed async evaluation of admission rules for saved searches, saved_splunk_co
5/7/24 1:58:11.724 PM	05-07-2024 17:58:11.724 +0000 INFO SavedSplunker - savedsearch_id="nobody;app_bad_spl;search_compiler_fast_v2", search_type="schedul ="search_compiler_fast_v2", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715104680, window_time ="", sid="scheduler__nobody_YXBwX2JhZF9zcGw__RMD54767f3e4b695ca72_at_1715104680_94409", suppressed=0, action_time_ms=2, thread_id="Al
5/7/24 1:58:11.719 PM	05-07-2024 17:58:11.719 +0000 INFO SavedSplunker - AlertNotifier::notifySearchCompleted: called for sid=scheduler__nobody_YXBwX2JhZF
5/7/24 1:58:00.234 PM	05-07-2024 17:58:00.234 +0000 INFO SavedSplunker - savedsearch_id="nobody;skynet-rest;splunk_rest_cluster_status", search_type="sche ="splunk_rest_cluster_status", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715104680, window_t s="", sid="scheduler__admin_c2t5bmV0LXJ1c3Q__RMD56993b9e85281ae60_at_1715104680_38611", suppressed=0, action_time_ms=1, thread_id="Al
5/7/24 1:58:00.231 PM	05-07-2024 17:58:00.231 +0000 INFO SavedSplunker - AlertNotifier::notifySearchCompleted: called for sid=scheduler__admin_c2t5bmV0LXJ

index=_introspection

The basics

- Data about your Splunk instance and environment
  - Resource usage by Splunk processes
  - Host resource usage
  - Disk I/O usage
  - KVStore performance info
- Default retention: 14 days
- Data is collected every 10 seconds (10 minutes on UFs)
- Populates some parts of the Expensive Searches dashboard in CMC

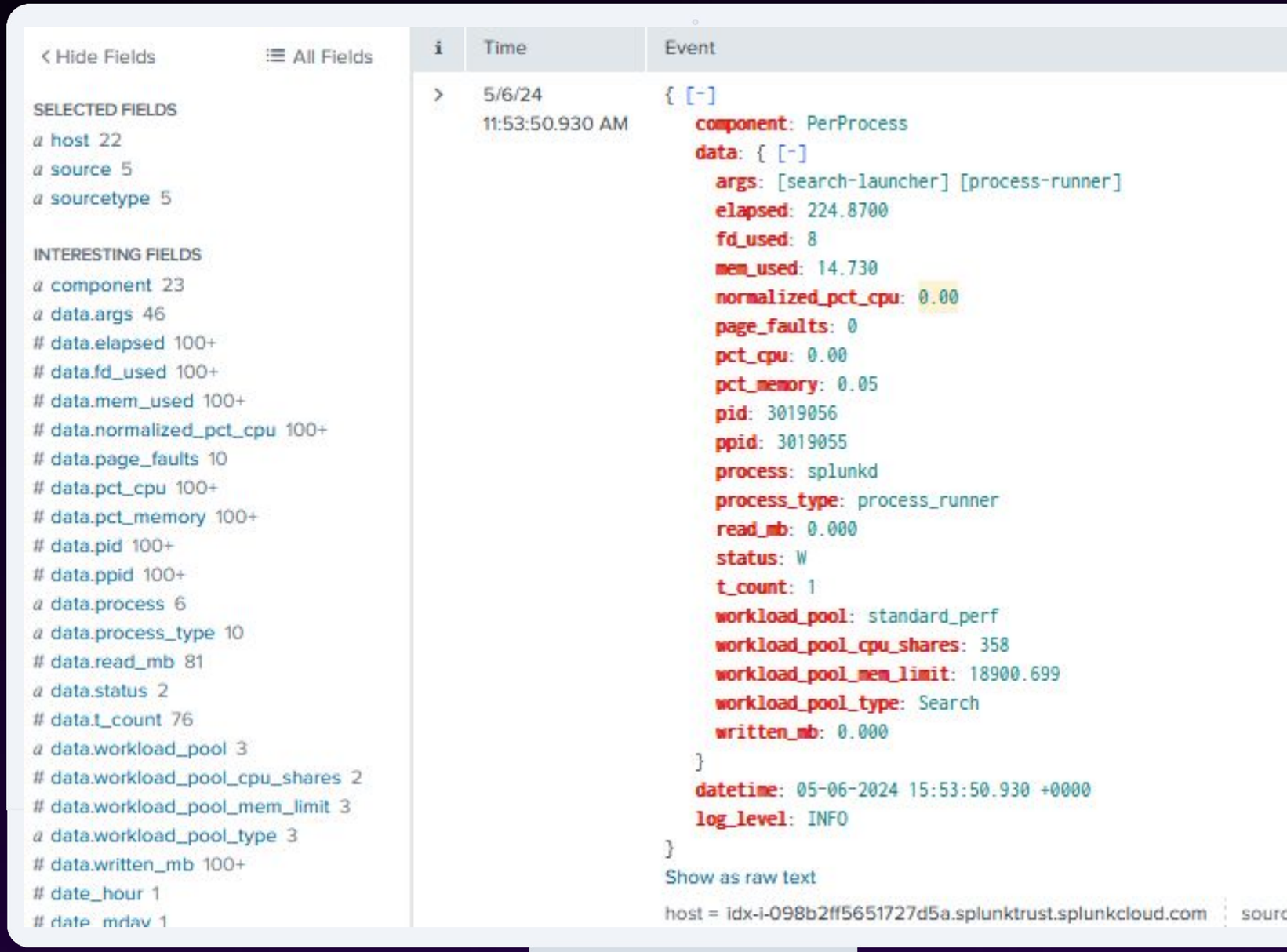


https://docs.splunk.com/Documentation/Splunk/9.2.1/RESTREF/RESTintrospect#

_introspection

The basics

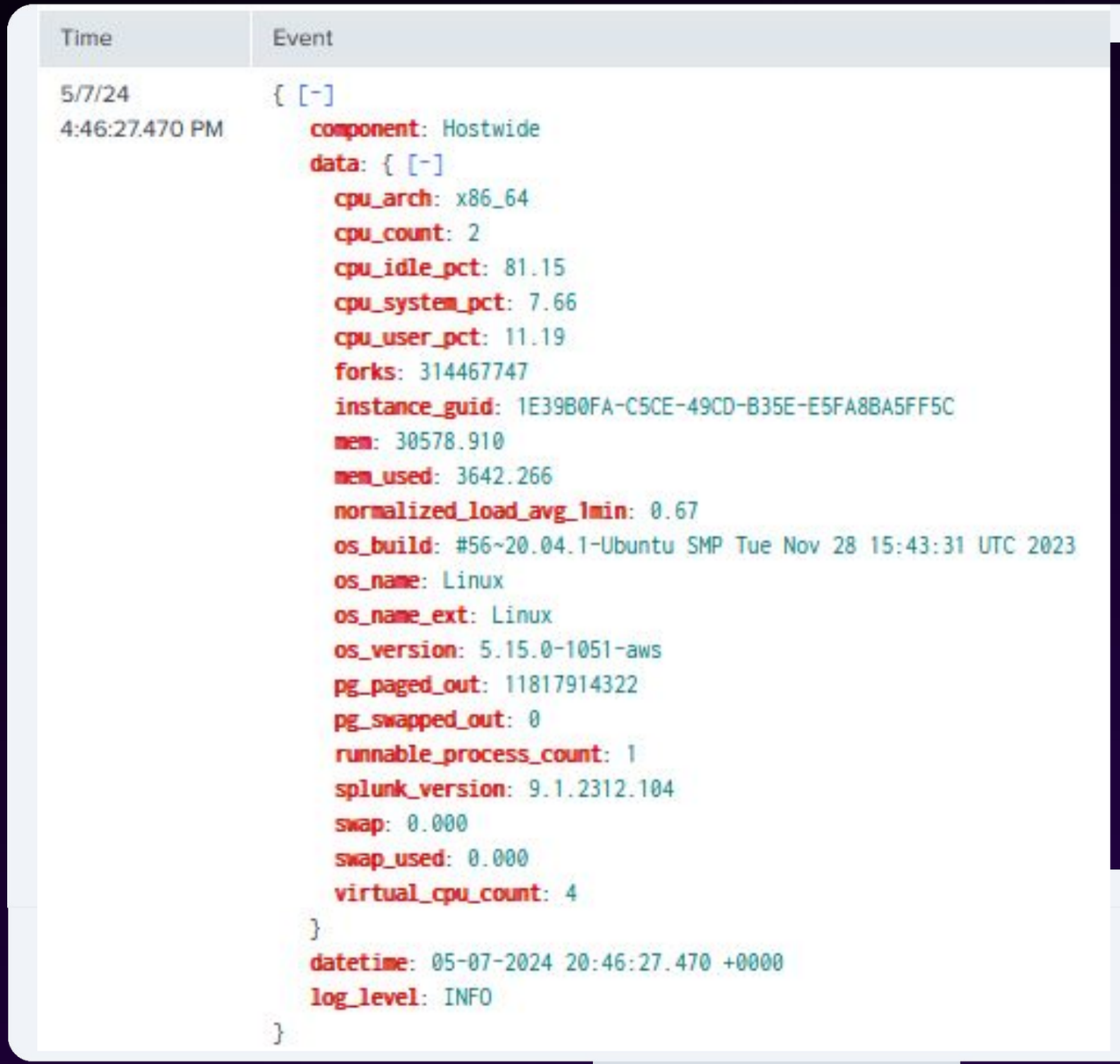
- 20+ components
- Interesting components:
  - PerProcess
  - Indexes
  - Hostwide
  - HttpEventCollector



_introspection

The basics

- 20+ components
- Interesting components:
  - PerProcess
  - Indexes
  - **Hostwide**
  - HttpEventCollector

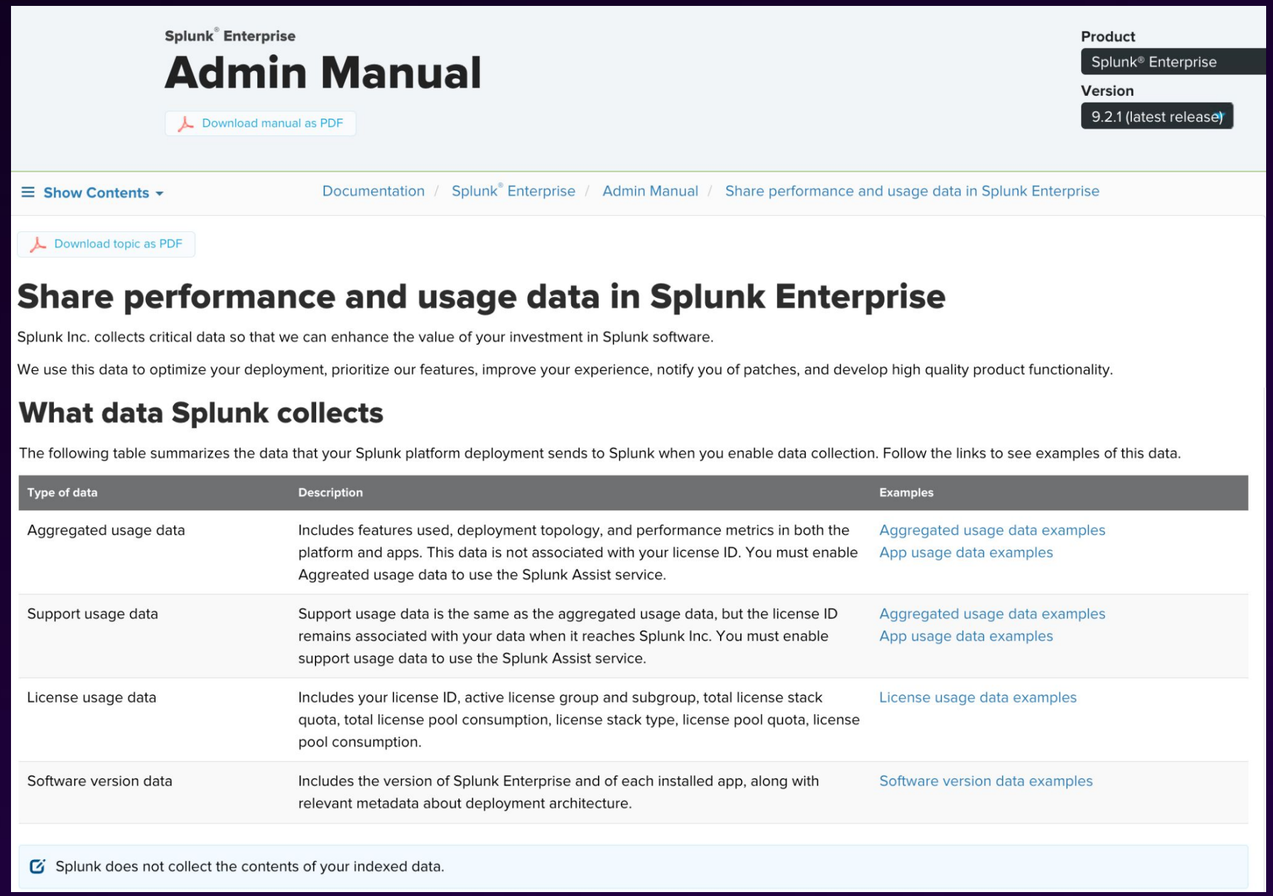


index=_telemetry

The basics

- Instrumentation info collected from internal data sources
  - Features, usage
  - Search types, command usage
  - License usage
- Default retention: 2 years
  - Data size limit: 256mb
- Index contents varies significantly based on Platform
  - Enterprise vs Cloud

https://docs.splunk.com/Documentation/Splunk/9.2.1/Admin/Shareperformancedata#



Splunk® Enterprise

Admin Manual

Download manual as PDF

Product: Splunk® Enterprise

Version: 9.2.1 (latest release)

Show Contents | Documentation / Splunk® Enterprise / Admin Manual / Share performance and usage data in Splunk Enterprise

Download topic as PDF

Share performance and usage data in Splunk Enterprise

Splunk Inc. collects critical data so that we can enhance the value of your investment in Splunk software.

We use this data to optimize your deployment, prioritize our features, improve your experience, notify you of patches, and develop high quality product functionality.

What data Splunk collects

The following table summarizes the data that your Splunk platform deployment sends to Splunk when you enable data collection. Follow the links to see examples of this data.

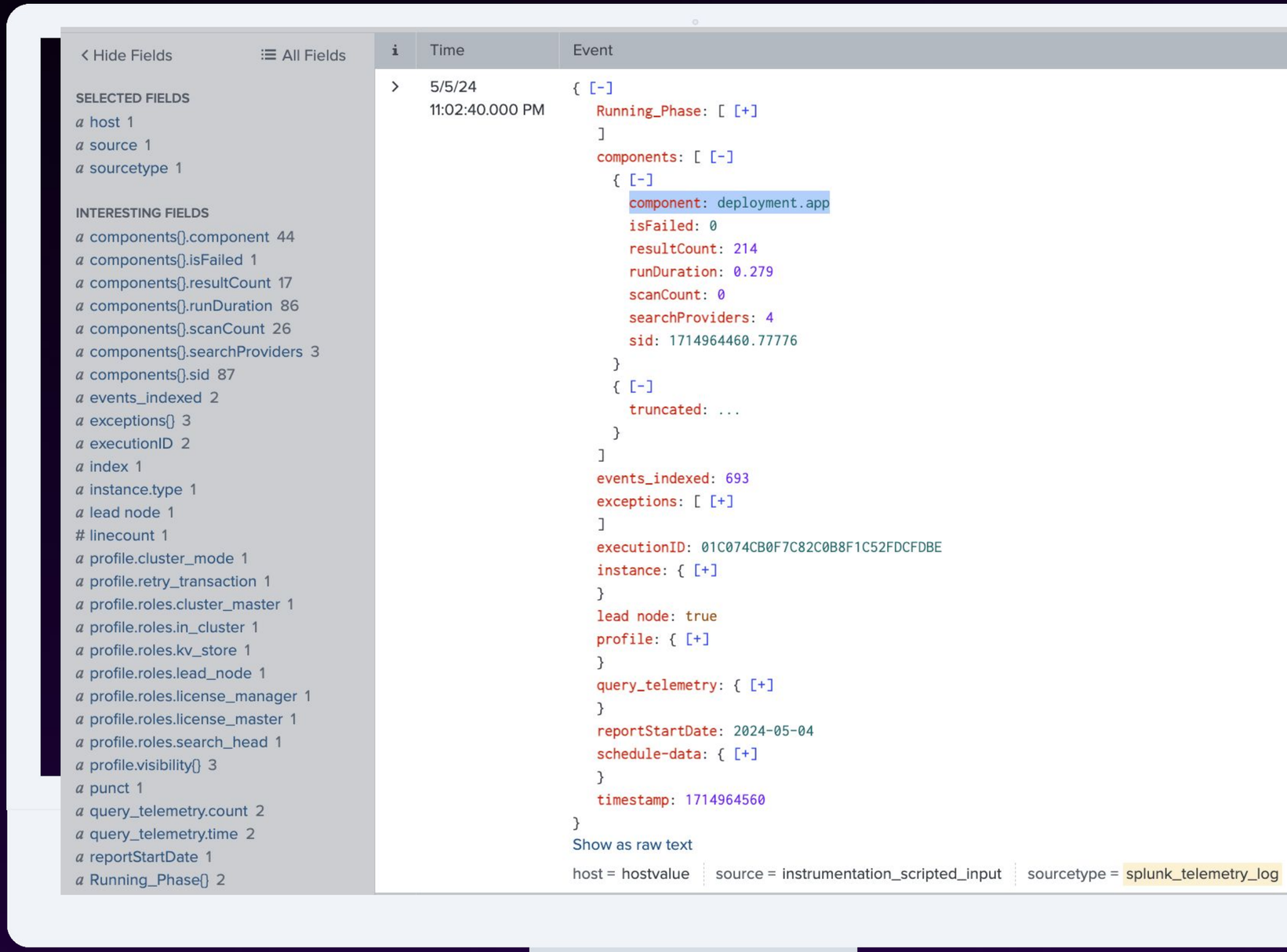
Type of data	Description	Examples
Aggregated usage data	Includes features used, deployment topology, and performance metrics in both the platform and apps. This data is not associated with your license ID. You must enable Aggreated usage data to use the Splunk Assist service.	Aggregated usage data examples App usage data examples
Support usage data	Support usage data is the same as the aggregated usage data, but the license ID remains associated with your data when it reaches Splunk Inc. You must enable support usage data to use the Splunk Assist service.	Aggregated usage data examples App usage data examples
License usage data	Includes your license ID, active license group and subgroup, total license stack quota, total license pool consumption, license stack type, license pool quota, license pool consumption.	License usage data examples
Software version data	Includes the version of Splunk Enterprise and of each installed app, along with relevant metadata about deployment architecture.	Software version data examples

Splunk does not collect the contents of your indexed data.

_telemetry

The basics

- *sourcetype IN (splunk_telemetry, splunk_telemetry_log)*
- Info related to telemetry job execution
- Collection jobs identified by "component" value

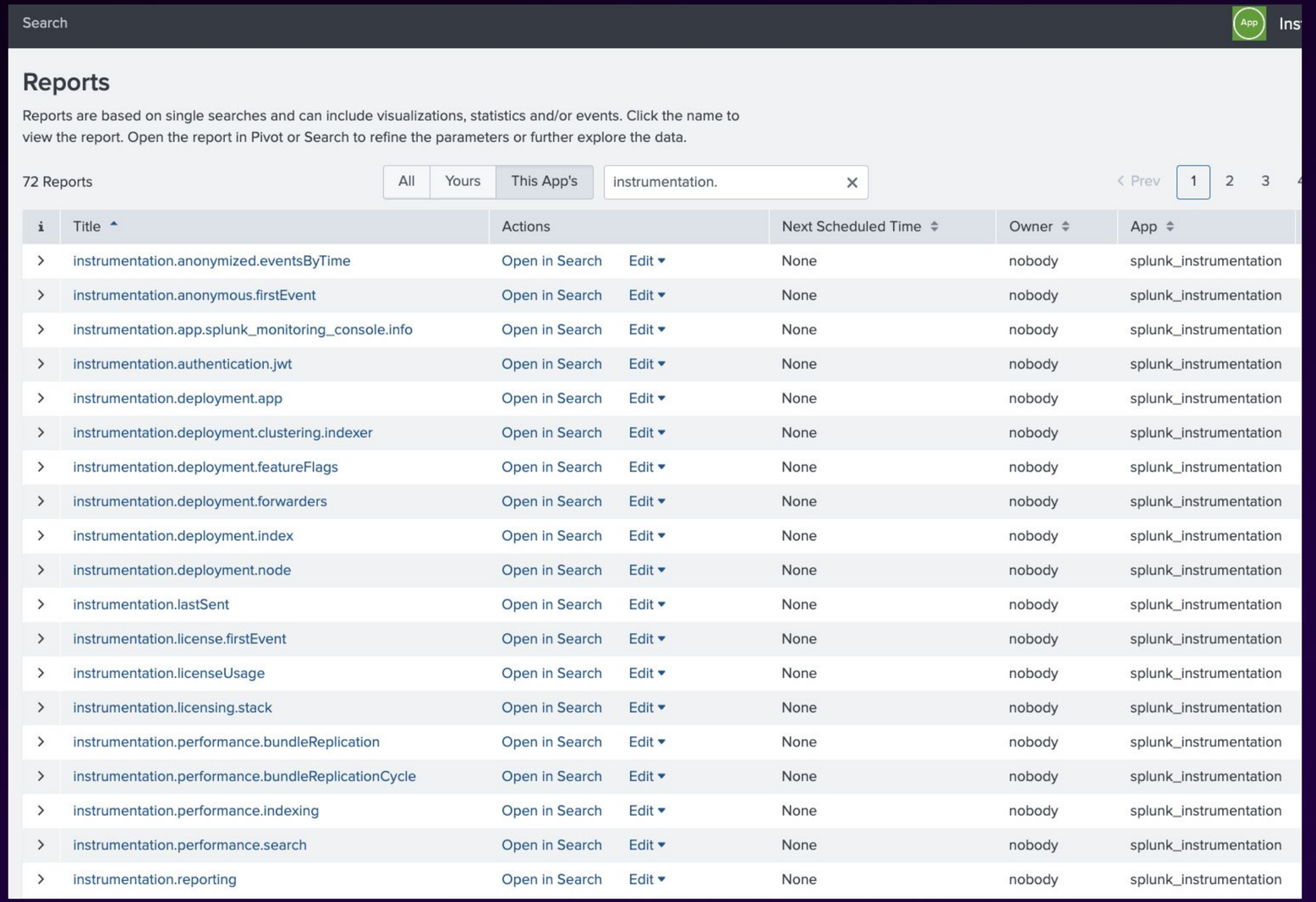


https://docs.splunk.com/Documentation/Splunk/9.2.1/Admin/Shareperformancedata#

_telemetry (splunk_instrumentation)

app=splunk_instrumentation

- Core app default in all Splunk
- Contains savedsearches used to collect telemetry data
  - Jobs run daily at 3am by default
  - Which jobs defined by Server Role
- Runs in both Enterprise & Cloud, but does not write data to index=_telemetry in Enterprise
  - Data written to _introspection



Title	Actions	Next Scheduled Time	Owner	App
instrumentation.anonymized.eventsByTime	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.anonymous.firstEvent	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.app.splunk_monitoring_console.info	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.authentication.jwt	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.app	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.clustering.indexer	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.featureFlags	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.forwarders	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.index	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.deployment.node	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.lastSent	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.license.firstEvent	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.licenseUsage	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.licensing.stack	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.performance.bundleReplication	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.performance.bundleReplicationCycle	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.performance.indexing	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.performance.search	Open in Search Edit	None	nobody	splunk_instrumentation
instrumentation.reporting	Open in Search Edit	None	nobody	splunk_instrumentation

_telemetry (instrumentation)

app=splunk_instrumentation

- Core app default in all Splunk
- Contains savedsearches used to collect telemetry data
  - Jobs run daily at 3am by default
- Runs in both Enterprise & Cloud, but does not write data to index=_telemetry in Enterprise
  - Data written to _introspection

Aggregated usage data examples

The following example demonstrates the data sent to Splunk when sharing of aggregated usage data is enabled.

Component	Description	Example
`usage.search.searchtelemetry.sourcetypeUsage`	Sourcetype usage.	{ [-] sourcetypeUsage: [[-] { [-] http_event_collector_metrics: 1 kvstore: 1 mongod: 3 search_telemetry: 1 splunk_disk_objects: 1 splunk_resource_usage: 1 splunk_web_service: 3 splunkd: 11 splunkd_remote_searches: 3 splunkd_ui_access: 2 }] }

* Docs contain examples of all components.

https://docs.splunk.com/Documentation/Splunk/9.2.1/Admin/Shareperformancedata#Examples_of_data_sent_to_Splunk

Instrumentation Data

Located in different indexes based on your environment type.

Splunk Cloud

index=_telemetry sourcetype=splunk_cloud_telemetry

- Same data as Enterprise, but nested within JSON.
- Much longer retention
- Additional JSON wrapper means rename is needed:

```
| rename data.* AS *
```

Splunk Enterprise

index=_introspection sourcetype=splunk_telemetry

- Much shorter retention
- Must be written to secondary index for longer retention
  - SPL to do this in PDF slides

Instrumentation Data

Located in different indexes based on your environment type.

Splunk Cloud

index=_telemetry sourcetype=splunk_cloud_telemetry

- Same data as Enterprise, but nested within JSON.


```
{ [-]
   component: TelemetryCloudData
   data: { [-]
     component: usage.search.searchtelemetry.sourcetypeUsage
     data: { [-]
       sourcetypeUsage: [ [-]
         { [-]
           audittrail: 350
           search_log_events: 889
           splunk_resource_usage: 206
           truncated: ...
         }
       ]
     }
     date: 2024-05-05
     deploymentID: CLOUD-3e530e68d52f4b3250ebf1ac98bf44b29a48d22f07d70e5b8bb7cee59a1393bf
     executionID: 01C074CB0F7C82C0B8F1C52FDCFDBE
     timestamp: 1714964466
     transactionID: 22D32667-589A-AA93-DB57-0605D9FE94F9
     version: 4
     visibility: anonymous,support
   }
   datetime: 2024-05-06 03:02:40,496
   log_level: INFO
}
```


Show as raw text

host = hostvalue | source = splunk_instrumentation_cloud.log | sourcetype = splunk_cloud_telemetry

Splunk Enterprise

index=_introspection sourcetype=splunk_telemetry


```
{ [-]
   component: usage.search.searchtelemetry.sourcetypeUsage
   data: { [-]
     sourcetypeUsage: [ [-]
       { [-]
         audittrail: 210
         scheduler: 176
         stash: 729
         truncated: ...
       }
     ]
   }
   date: 2024-05-04
   executionID: 0C671A090B3E117CB35D3E885518C3
   timestamp: 1714950102
   visibility: anonymous,support
}
```


Show as raw text

host = hostvalue | source = http-stream | sourcetype = splunk_telemetry

splunk_instrumentation Additional Detail

Additional information related to instrumentation app

- Instrumentation jobs are launched via scheduled python job, which is why savedsearch objects are not scheduled in UI.

- All Instrumentation jobs contain terms:
  - executionID
  - component

- Details for each instrumentation component can be found in docs: https://docs.splunk.com/Documentation/Splunk/9.2.1/Admin/Shareperformancedata#Examples_of_data_sent_to_Splunk

- Scheduling and configurations for telemetry collection are defined in *telemetry.conf* https://docs.splunk.com/Documentation/Splunk/9.2.1/Admin/Telemetryconf
  - Whether to send each type of data

Instrumentation - Saving Job Data for Longer Retention in Splunk Enterprise

Enterprise - Straightforward SPL for populating _telemetry with what's stored in introspection by default

```
rex field=$field$ max_match=100 "`(?<Macro_Reference>\p{Any}+?)`"
| rex field=Macro_Reference mode=sed "s/\"|\s+//g"
| eval Macro_Reference = mvfilter((! match(Macro_Reference,"^\||^\)|^:|^\[|^comment|^ia4s_comment")))
| eval Macro_Reference = if(((Macro_Reference == "") OR isnull(Macro_Reference)), "no-macro-reference", Macro_Reference)
| mvexpand Macro_Reference
| rex field=Macro_Reference max_match=100 "(?<Macro_Name>^[a-zA-Z0-9_-]+)"
| rex field=Macro_Reference max_match=100 "\((?<Macro_Args>.*?)\)"
| makemv delim="," Macro_Args
| eval Macro_Args_Count = mvcount(Macro_Args)
| eval Macro_Title = if (isnull(Macro_Args_Count), Macro_Name, Macro_Name . "(" . Macro_Args_Count . ")")
| eval Macro_Title = if(((Macro_Title == "") OR isnull(Macro_Title)), "no-macro-title", Macro_Title)
| fields - Macro_Reference1 Macro_Name Macro_Args Macro_Args_Count
```

Tips, Common Questions, Etc

Heads ups, Gotchas, Tips, Warnings, Requirements, Etc. related to the internal data sources available.

- Audittrail events automatically extract key=value fields, including those embedded in the search field.
- Don’t assume anything is 100%
- Be aware of index retention times.
- Long search time ranges with SmartStore indexes may be slower because of cache misses.

But how can we use it?

Usecases & Examples

.conf24

splunk>

Bring on the Usecases

Usecase - Identifying Problematic Searches

Introduce first technical walkthrough item - identifying problematic searches/search problems

List search problem examples

- Broken search queries
- Failed or Skipped Searches
- User Searches hitting Quotas
  - Search Concurrency Quotas
  - Disk Usage Quotas
- Long-running Searches
- Resource-intensive Searches
  - Hidden slides - would require introspection which we don't have time for
- Expensive or Inefficient Searches
  - *Maybe not this one here.*

Usecase - Identifying Problematic Searches

_internal

sourcetype=scheduler

- Scheduled search information

*status=**

- …

_audit

sourcetype=audittrail

- Search execution audit events

action=search

- Filter to search events

_introspection

sourcetype=splunk_resource_usage component=PerProcess

- Performance metrics for Splunk processes

*data.search_props.sid::**

- Search process metrics

Usecase - Identifying Problematic Searches

Using _internal sourcetype=scheduler

Scheduler data contains…

Insert picture of scheduler events to right

```
05-05-2024 21:10:30.757 +0000 INFO  SavedSplunker -
savedsearch_id="nobody;splunk_app_stream;_ACCELERATE_C88762F5-BF4D-4B43-9537-
DA44D898C9F2_splunk_app_stream_nobody_615f5f04b93533e7_ACCELERATE_",
search_type="report_acceleration", search_streaming=0, user="nobody", app="splunk_app_stream",
savedsearch_name="_ACCELERATE_C88762F5-BF4D-4B43-9537-
DA44D898C9F2_splunk_app_stream_nobody_615f5f04b93533e7_ACCELERATE_", priority=default,
status=success, digest_mode=1, durable_cursor=0, scheduled_time=1714943400, window_time=0,
dispatch_time=1714943400, run_time=10.929, result_count=324, alert_actions="",
sid="scheduler__nobody_c3BsdW5rX2FwcF9zdHJlYW0__RMD5e27d6c7682f4b855_at_1714943400_10630",
suppressed=0, action_time_ms=1, thread_id="AlertNotifierWorker-0", workload_pool=""

05-05-2024 21:10:19.878 +0000 INFO  SavedSplunker -
savedsearch_id="nobody;itsi;service_health_monitor", search_type="scheduled",
search_streaming=0, user="nobody", app="itsi", savedsearch_name="service_health_monitor",
priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1714943400,
window_time=-1, dispatch_time=1714943414, run_time=5.571, result_count=0,
alert_actions="summary_index",
sid="scheduler__nobody__itsi__RMD507f8603927e905f9_at_1714943400_10658", suppressed=0,
action_time_ms=2, thread_id="AlertNotifierWorker-0", workload_pool=""

05-05-2024 21:10:19.387 +0000 INFO  SavedSplunker -
savedsearch_id="nobody;itsi;disabled_kpis_healthscore_generator_metrics",
search_type="scheduled", search_streaming=0, user="nobody", app="itsi",
savedsearch_name="disabled_kpis_healthscore_generator_metrics", priority=default,
status=success, digest_mode=1, durable_cursor=0, scheduled_time=1714943400, window_time=-1,
dispatch_time=1714943419, run_time=0.084, result_count=0,
alert_actions="itsi_summary_metrics_collect",
sid="scheduler__nobody__itsi__RMD598fe407928707731_at_1714943400_10670", suppressed=0,
action_time_ms=2, thread_id="AlertNotifierWorker-0", workload_pool=""
```

Usecase - Identifying Problematic Searches

Using _internal sourcetype=scheduler

Scheduler data contains…

Edit previous picture to highlight notable fields

Notable fields:

- SID - Search ID
- savedsearch_name
- status - Search status
- Reason - Reason for status

…

Usecase - Identifying Problematic Searches

Fuller Breakout of internal scheduler fields? - **Hidden Slide**

Placeholder - Detailed Info on fields

© 2024 SPLUNK INC.

Usecase - Identifying Problematic Searches

SPL view using internal scheduler to identify search problems

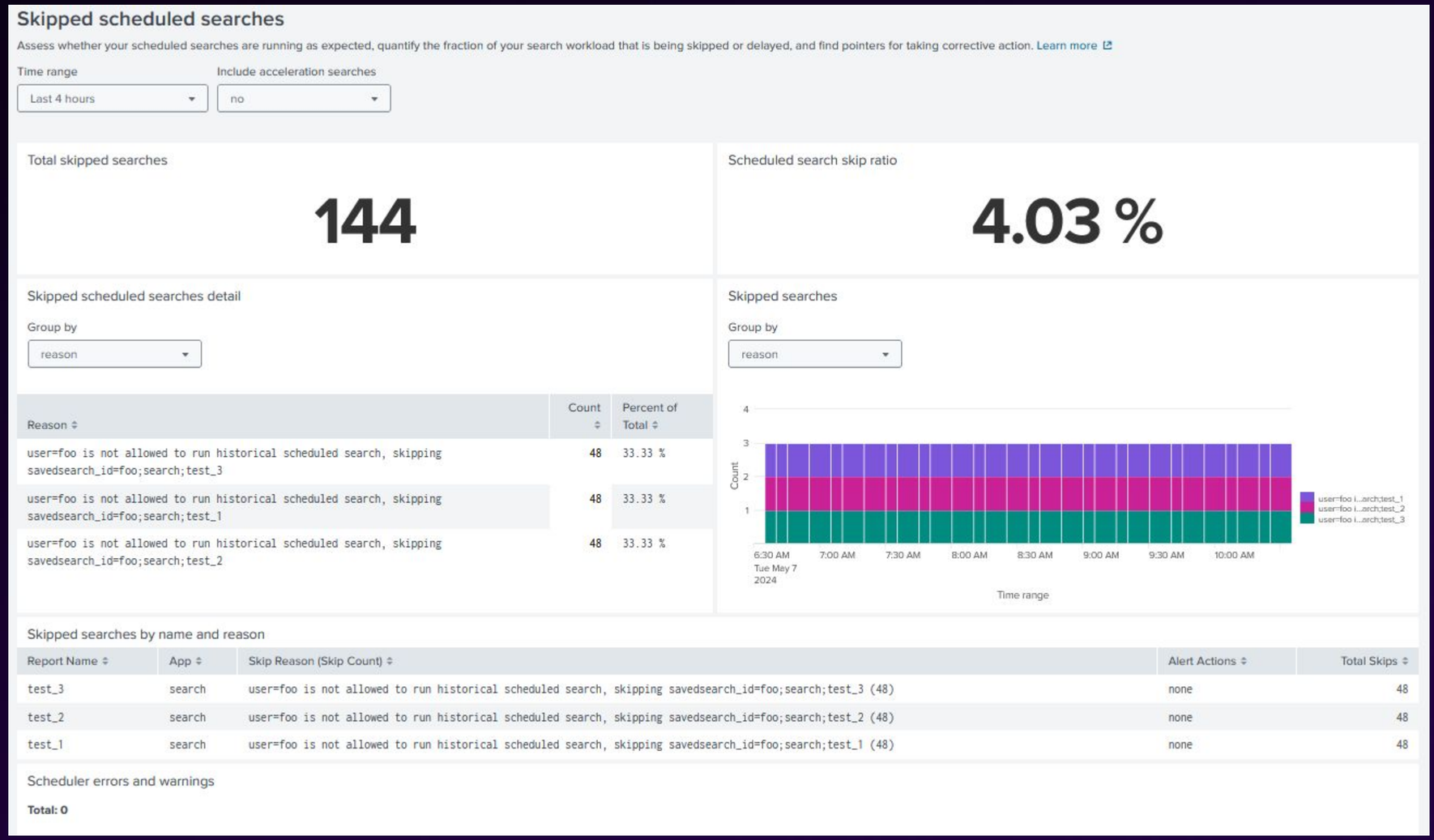
Placeholder - Picture of SPL output report of internal scheduler showing search problems

© 2024 SPLUNK INC.

Usecase - Identifying Problematic Searches

Cloud Monitoring Console

Similar information for Splunk Enterprise users is in the Monitoring Console at Scheduler Activity: Instance

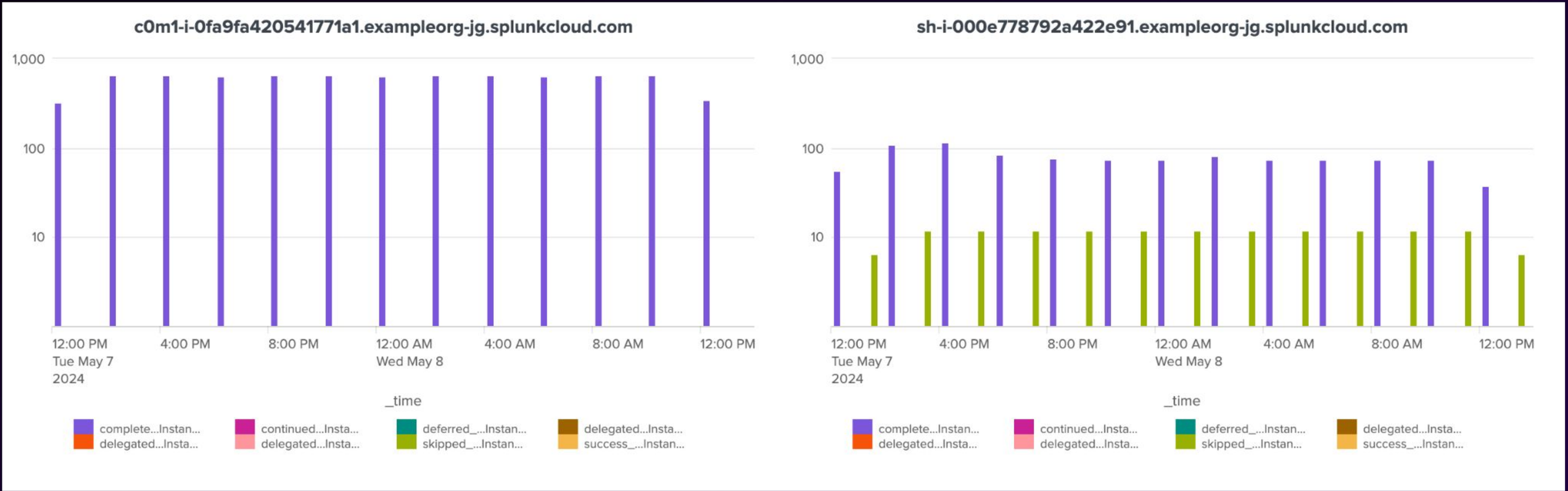


© 2024 SPLUNK INC.

Scheduled Searches Breakout by Status

Timechart view using trellis for detailed breakdown of scheduled search status over time.

Enable Trellis by host



Scheduled Searches Timechart by Status

Timechart view showing granular count by status over time by host. - Enable trellis

```
index=_internal sourcetype=scheduler status=*
| fields _time, savedsearch_id, sid, scheduled_time, status, reason, host
| eval search_id = coalesce(savedsearch_id, sid)
| eval UniqueSearchInstanceIdentifier = search_id."::::".scheduled_time
| eval comment = if( 1==1, null(), "
We treat each unique combination of search_id and scheduled_time as an instance of a search that should execute. We then take the
earliest timestamp for each combination to attribute that status to the timebucket when it began, rather than when it ends or
counting it repeatedly.
This could also be set to take the latest() to attribute the status value to the last event per instance, or not use this logic
at all to take a distinct count in each time bucket regardless of whether that search had already been counted in a previous time
bucket." )
| stats latest(_time) AS _time, latest(*) AS * BY UniqueSearchInstanceIdentifier, status, host
| timechart span=2h dc(eval(if(status="delegated_remote", UniqueSearchInstanceIdentifier, null()))) AS
delegated_remote_dcSearchInstances, dc(eval(if(status="delegated_remote_completion", UniqueSearchInstanceIdentifier, null()))) AS
delegated_remote_completion_dcSearchInstances, dc(eval(if(status="delegated_remote_error", UniqueSearchInstanceIdentifier,
null()))) AS delegated_remote_error_dcSearchInstances, dc(eval(if(status="skipped", UniqueSearchInstanceIdentifier, null()))) AS
skipped_dcSearchInstances, count(eval(if(status="success", UniqueSearchInstanceIdentifier, null()))) AS
success_dcSearchInstances, count(eval(if(status="completed", UniqueSearchInstanceIdentifier, null()))) AS
completed_dcSearchInstances, count(if(eval(status="continued", UniqueSearchInstanceIdentifier, null()))) AS
continued_dcSearchInstances, count(if(eval(status="deferred", UniqueSearchInstanceIdentifier, null()))) AS
deferred_dcSearchInstances BY host
```

Skipped Searches by…

Simple Timechart view of skipped searches by savedsearch_name

```
index=_internal sourcetype=scheduler status=skipped
| timechart count by user, app, savedsearch_name
```

Simple view of skipped searches by reason

```
index=_internal sourcetype=scheduler status=skipped reason=*
| stats count, values(reason) AS reason by user, app, savedsearch_name
```

© 2024 SPLUNK INC.

Usecase - Identifying Problematic Searches

References for Identifying Search Problems - **HIDDEN SLIDE**

HIDDEN SLIDE

Placeholder - Other Resources for Identifying Search Problems

- Monitoring Console Dashboard views
- Instrumentation app searches
- Presentations
- etc

© 2024 SPLUNK INC.

What if I wanted to answer...

- What search SPL query was this SID running?
- Alert actions happening and the roles attached to searches doing them?
- Top 10 searches by memory usage and when it was run?

What if I wanted to answer…

- Searches with highest # scanned buckets and bucket count by indexer?
- Alert actions happening and the roles attached to searches doing them?
- Top 10 searches by memory usage and when it was run?

Search ID!
(SID)

Search - Additional Details

Additional information related to search processes

- Search is a complicated subject with a lot of components, but a good starting place is the Search Manual: https://docs.splunk.com/Documentation/Splunk/latest/Search/GetstartedwithSearch
- Information related to Search Jobs: https://docs.splunk.com/Documentation/SplunkCloud/latest/Search/Aboutjobsandjobmanagement
- On Dispatch directory artifact files, structure, descriptions https://docs.splunk.com/Documentation/SplunkCloud/latest/Search/Dispatchdirectoryandsearchartifacts

Search - Optimizing Searches Resources

Additional information related to optimizing search processes

- Docs on optimizing searches:
  https://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutoptimization
- Splunk Lantern on Optimizing searches:
  Lantern - Optimizing Search
  Lantern - Writing Better Search Queries

- David Paper's Extended Search Reporting Dashboard

Default Data Sources with Search Info

Some of the default data sources containing SIDs & Search Information.

```
index=_internal
sourcetype=splunkd_remote_searches
```

```
index=_introspection
sourcetype=splunk_resource_usage
component=PerProcess
```

```
index=_audit
sourcetype=audittrail
```

```
index=_internal
sourcetype=scheduler
```

© 2024 SPLUNK INC.

Usecase - Unifying SIDs

Showing the effect of normalizing SIDs

index=_internal sourcetype=splunkd_remote_searches

index=_introspection sourcetype=splunk_resource_usage

index=_audit sourcetype=audittrail

index=_internal sourcetype=scheduler

Event

05-07-2024 03:59:58.866 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-000e778792a422e91.examplehost-jg.splunkcloud.com_prd.ph1_1715054398.80946, server=sh-i-000e778792a422e91.examplehost-jg.splunkcloud.com, active_searches=1, elapsedTime=0.732, search='rdin rdout_sid="prd.ph0_1715054398.80946_sh-i-000e778792a422e91.examplehost-jg.splunkcloud.com" intermediaries="idx-i-0cf4e337ebc4c2c83.examplehost-jg.splunkcloud.com,idx-i-0e3bd3088c44ea9db.examplehost-jg.splunkcloud.com" order_sensitive="0" allow_partial_results="0" | tstats summariesonly=false allow_old_summaries=false count WHERE index=_introspection BY splunk_server | prestats count sum(count)', savedsearch_name="", drop_count=0, scan_count=0, eliminated_buckets=0, considered_events=0, decompressed_slices=0, events_count=0, total_slices=0, considered_buckets=0, search_rawdata_bucketcache_error=0, search_rawdata_bucketcache_miss=0, search_index_bucketcache_error=0, search_index_bucketcache_hit=0, search_index_bucketcache_miss=0, search_rawdata_bucketcache_hit=0, search_rawdata_bucketcache_miss_wait=0.000, search_index_bucketcache_miss_wait=0.000

host = examplecloudstack.splunkcloud.com | source = /opt/splunk/var/log/splunk/remote_searches.log | sourcetype = splunkd_remote_searches

{"datetime":"05-07-2024 03:59:57.975 +0000","log_level":"INFO","component":"PerProcess","data":{"pid":"3023884","ppid":"3297339","status":"W","t_count":"11","mem_used":"187.473","pct_memory":"1.20","page_faults":"0","pct_cpu":"106.60","normalized_pct_cpu":"26.65","read_mb":"0.000","written_mb":"6.891","fd_used":"104","elapsed":"2279.7400","process":"splunkd","process_type":"search","workload_pool":"standard_perf","workload_pool_type":"Search","workload_pool_mem_limit":"9218.297","workload_pool_cpu_shares":"358","search_props":{"sid":"remote_sh-i-000e778792a422e91.examplehost-jg.splunkcloud.com_scheduler_cnlhbi5nYXJyQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD54ae33fa6044677a6_at_1715054100_9739","user":"ryan.garr@examplehostsecurity.com","app":"search","label":"new_user_created","provenance":"scheduler","scan_count":"0","delta_scan_count":"0","search_head":"sh-i-000e778792a422e91.examplehost-jg.splunkcloud.com","role":"peer","mode":"historical","type":"scheduled"}}}

Show syntax highlighted

host = examplecloudstack.splunkcloud.com | source = /opt/splunk/var/log/introspection/resource_usage.log | sourcetype = splunk_resource_usage

Audit:[timestamp=05-07-2024 03:59:28.488, user=internal_observability, action=search, info=completed, search_id='1715054338.80943', has_error_warn=false, fully_completed_search=true, total_run_time=0.82, event_count=765, result_count=1, available_count=0, scan_count=765, drop_count=0, exec_time=1715054338, api_et=1715054160.000000000, api_lt=1715054220.000000000, api_index_et=N/A, api_index_lt=N/A, search_et=1715054160.000000000, search_lt=1715054220.000000000, is_realtime=0, savedsearch_name="", search_startup_time="64", is_prjob=true, is_flex_search=false, rate_limit_retry_enabled=false, dispatch_artifact_bytes=299008, status_csv_bytes=4096, is_fss3=false, acceleration_id="D347BC42-6954-4328-9ABE-26B97C9DBF62_search_internal_observability_9e9f02a8f45b44d7", app="search", provenance="N/A", mode="historical_batch", workload_pool=standard_perf, is_proxied=false, searched_buckets=18, eliminated_buckets=0, considered_events=0, total_slices=0, decompressed_slices=0, duration.command.search.index=0, invocations.command.search.index.bucketcache.hit=18, duration.command.search.index.bucketcache.hit=0, invocations.command.search.index.bucketcache.miss=0, duration.command.search.index.bucketcache.miss=0, invocations.command.search.index.bucketcache.error=0, duration.command.search.rawdata=0, invocations.command.search.rawdata.bucketcache.hit=0, duration.command.search.rawdata.bucketcache.hit=0, invocations.command.search.rawdata.bucketcache.miss=0, duration.command.search.rawdata.bucketcache.miss=0, invocations.command.search.rawdata.bucketcache.error=0, roles='observability_role', search='| tstats count where index=_introspection by splunk_server | stats sum(count) AS event_count count AS indexer_count', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]

host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail

05-07-2024 03:59:00.497 +0000 INFO SavedSplunker - savedsearch_id="nobody;skynet-rest;splunk_rest_cluster_status", search_type="scheduled", search_streaming=0, user="admin", app="skynet-rest", savedsearch_name="splunk_rest_cluster_status", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715054340, window_time=0, dispatch_time=1715054340, run_time=0.189, result_count=0, alert_actions="", sid="scheduler__admin_c2t5bmV0LXJlc3Q__RMD56993b9e85281ae60_at_1715054340_20586", suppressed=0, action_time_ms=1, thread_id="AlertNotifierWorker-0", workload_pool=""

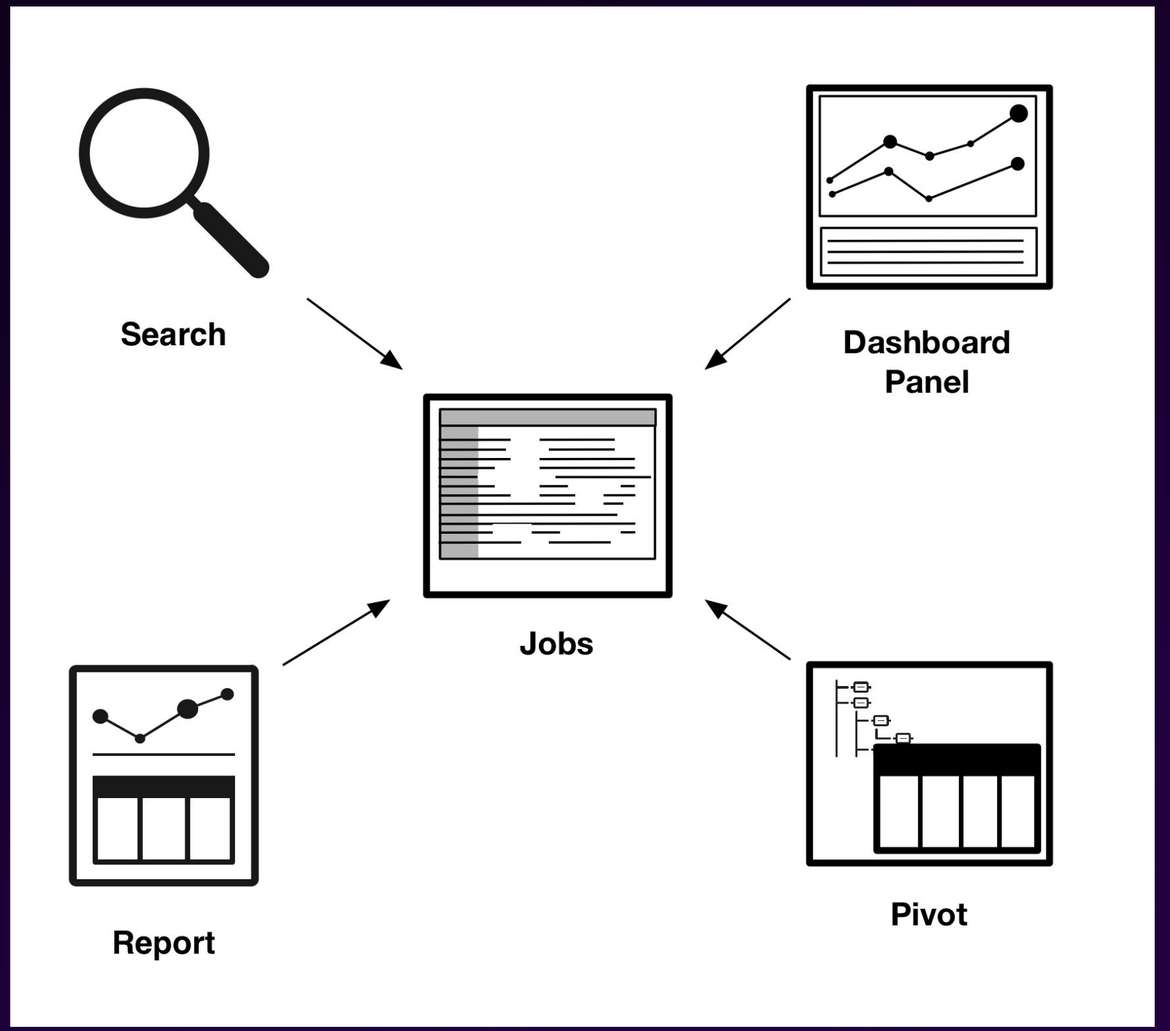
host = examplecloudstack.splunkcloud.com | source = /opt/splunk/var/log/splunk/scheduler.log | sourcetype = scheduler

© 2024 SPLUNK INC.

Level Set - Search Jobs

A quick refresher

- Each time a search is run, Splunk software creates a **search job** in the system.
- Job data is stored within the **dispatch** directory of Splunk
- Search job directories are differentiated using unique ID:
  Search ID.
- **Search ID**s notate the unique directory location of the **artifact files** within the **dispatch** directory



Level Set - Search Types

Types of Searches affect Search ID syntax

Types per Docs:

- Local Ad Hoc search
- Saved search
- Scheduled search
- Remote search (peer)
- Real-time search
- Replicated search
- Report acceleration search

Search Types and Example SID formats with descriptions available in docs:

Dispatch directory naming conventions

The names of the search-specific directories in the dispatch directory are based on the type of search. For saved and scheduled searches, the name of a search-specific directory is determined by the following conditions.

Type of search	Naming convention	Examples

Saved search	The user requesting the search, the user context the search is run as, the app the search came from, the search string, and the UNIX time.	"count" – run by **admin**, in user context **admin**, saved in app **search**
		`admin__admin__search__count_1347454406.2`
		"Errors in the last 24 hours" – run by **somebody**, in user context **somebody**, saved in app **search**
		`somebody__somebody__search_RMD5473cbac83d6c9db7_1347455134.20`

Scheduled search	The user requesting the search, the user context the search is run as, the app the search came from, the search string, UNIX time, and an internal ID added at the end to avoid name collisions.	"foo" – run by the **scheduler**, with no user context, saved in app **unix**
		`scheduler__nobody__unix__foo_at_1347457380_051d958b8354c580`
		"foo2" - remote peer search on search head **sh01**, with **admin** user context, run by the **scheduler**, saved in app **search**
		`remote_sh01_scheduler__admin__search__foo2_at_1347457920_79152a9a8bf33e5e`

https://docs.splunk.com/Documentation/SplunkCloud/latest/Search/Dispatchdirectoryandsearchartifacts#Dispatch_directory_naming_conventions

Level Set - Search Types

Types of Searches affect Search ID syntax

A few search ID prefixes we need to keep in mind for this talk, as they need to be normalized to get accurate reporting:

- Scheduled search
  `scheduler_*`
- Remote peer
  `remote_*`
- Subsearch
  `subsearch_*_1347457148.1`
- Replicated SHC
  `rsa_*`

* Normalizing SPL available in PDF

https://docs.splunk.com/Documentation/SplunkCloud/latest/Search/Dispatchdirectoryandsearchartifacts#Dispatch_directory_naming_conventions

Usecase - Normalizing SIDs

Showing the effect of normalizing SIDs

index=_internal
sourcetype=splunkd_remote_searches

index=_introspection
sourcetype=splunk_resource_usage
component=PerProcess

index=_audit
sourcetype=audittrail

index=_internal
sourcetype=scheduler

```
(index=_audit sourcetype=audittrail search_id="*") OR
(index=_internal sourcetype=scheduler sid="*") OR
(index=_internal sourcetype="splunkd_remote_searches" search_id="*") OR
(index=_introspection sourcetype=splunk_resource_usage data.search_props.sid="*")

| eval unified_sid = case( sourcetype="audittrail", trim(search_id, "'"),
                           sourcetype="scheduler", sid,
                           sourcetype="splunkd_remote_searches", search_id,
                           sourcetype="splunk_resource_usage", 'data.search_props.sid')
| `get_normalized_search_id(unified_sid)`
| rename search_id_normalized AS unified_sid_normalized
| stats dc(unified_sid), dc(unified_sid_normalized) BY index, sourcetype
```

index	sourcetype	dc(unified_sid)	dc(unified_sid_normalized)
_introspection	splunk_resource_usage	9,492	6,029
_audit	audittrail	503,110	110,105
_internal	scheduler	66,117	66,116
_internal	splunkd_remote_searches	36,478	21,851

© 2024 SPLUNK INC.

Normalizing SIDs - SPL

SPL for slide aggregating 4 data sources with SID

```
(index=_audit sourcetype=audittrail search_id="*") OR
(index=_internal sourcetype=scheduler sid="*") OR
(index=_internal sourcetype="splunkd_remote_searches" search_id="*") OR
(index=_introspection sourcetype=splunk_resource_usage data.search_props.sid="*")

| eval unified_sid = case( sourcetype="audittrail", trim(search_id, "'"),
                           sourcetype="scheduler", sid,
                           sourcetype="splunkd_remote_searches", search_id,
                           sourcetype="splunk_resource_usage", 'data.search_props.sid')
| `get_normalized_search_id(unified_sid)`
| rename search_id_normalized AS unified_sid_normalized
| stats dc(unified_sid), dc(unified_sid_normalized) BY index, sourcetype
```

Macro: get_normalized_search_id(1)

SPL Gathered from SCMA App (Splunk Cloud Migration Assessment)

SCMA Macro: *get_normalized_search_id(1)* — inserting "unified_sid" field

```
| rex field=unified_sid "_(?<search_id_normalized1>\d+[._]\d+)_"
| rex field=unified_sid "(?<search_id_normalized2>\d+[._]\d+$)"
| rex field=unified_sid "(?<search_id_normalized3>^\d+[._]\d+)"
| eval search_id_normalized=if(isnull(search_id_normalized1),search_id_normalized2,search_id_normalized1)
| eval search_id_normalized=if(isnull(search_id_normalized),search_id_normalized3,search_id_normalized)
| eval search_id_normalized=if(isnull(search_id_normalized),search_id,search_id_normalized)
| rex field=search_id_normalized mode=sed "s/\./_/g"
| rex field=search_id_normalized mode=sed "s/^\w+;.*;|^_ACCELERATE_DM_|^_ACCELERATE_|_ACCELERATE_$//g"
| fields - search_id_normalized1,search_id_normalized2,search_id_normalized3
```

What if I wanted to answer...

- What search SPL query was this SID running?
- Alert actions happening and the roles attached to searches doing them?
- Top 10 searches by memory usage and when it was run?

Usecase - Normalizing SIDs

What SPL Query did this SID run?

```
SID + _audit audittrail
```

© 2024 SPLUNK INC.

Usecase - Normalizing SIDs

Alert Actions being run and the roles on searches running them.

- _audit audittrail
  - *search_id*
  - *roles*
- _internal scheduler
  - *sid*
  - *alert_actions*

_audit audittrail + _internal scheduler

Event

Audit:[timestamp=05-08-2024 01:16:58.481, user=ryan.wood@exampleorg.com, action=search, info=completed, search_id='scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887', has_error_warn=false, fully_completed_search=true, total_run_time=156.43, event_count=8872255, [...truncated...], savedsearch_name="Sample Data Generation", [...truncated...], search_type=scheduled, roles='gps_admin+power+tokens_auth+user', search='search index=_internal user=*
| stats count by index
| append
[search index=_audit sourcetype=audittrail
| stats count by index]', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]

Collapse

host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail

Event

05-08-2024 01:16:36.614 +0000 INFO SavedSplunker - savedsearch_id="nobody;search;Sample Data Generation", search_type="scheduled", search_streaming=0, user="ryan.wood@exampleorg.com", app="search", savedsearch_name="Sample Data Generation", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715130780, window_time=-1, dispatch_time=1715130839, run_time=156.427, result_count=2, alert_actions="email,logevent,lookup", sid="scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887", suppressed=0, action_time_ms=544, thread_id="AlertNotifierWorker-0", workload_pool="standard_perf"

Usecase - Normalizing SIDs

Alert Actions being run and the roles on searches running them.

- _audit audittrail
  - *search_id*
  - *roles*
- _internal scheduler
  - *sid*
  - *alert_actions*

_audit audittrail + _internal scheduler

Event

```
Audit:[timestamp=05-08-2024 01:16:58.481, user=ryan.wood@exampleorg.com, action=search, info=completed,
search_id='scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887', has_error_warn=false, fully_completed_search=true, total_run_time=156.43, event_count=8872255, [...truncated...], savedsearch_name="Sample Data Generation", [...truncated...], search_type=scheduled, roles='gps_admin+power+tokens_auth+user',
search='search index=_internal user=*
| stats count by index
| append
    [search index=_audit sourcetype=audittrail
    | stats count by index]', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]
Collapse
host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail
```

Event

```
05-08-2024 01:16:36.614 +0000 INFO  SavedSplunker - savedsearch_id="nobody;search;Sample Data Generation", search_type="scheduled", search_streaming=0, user="ryan.wood@exampleorg.com", app="search", savedsearch_name="Sample Data Generation", priority=default, status=success, digest_mode=1, durable_cursor=0, scheduled_time=1715130780, window_time=-1, dispatch_time=1715130839, run_time=156.427, result_count=2, alert_actions="email,logevent,lookup", sid="scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887", suppressed=0, action_time_ms=544, thread_id="AlertNotifierWorker-0", workload_pool="standard_perf"
```

Usecase - Normalizing SIDs

Alert Actions being run and the roles on searches running them.

```
_audit audittrail + _internal scheduler
```

- _audit audittrail
  - *search_id*
  - *roles*
- _internal scheduler
  - *sid*
  - *alert_actions*

sourcetype	search_id_normalized	savedsearch_name	alert_actions	roles	user
audittrail	1715130780_10887	Sample Data Generation		gps_admin power tokens_auth user	ryan.wood@exampleorg.com
scheduler	1715130780_10887	Sample Data Generation	email logevent lookup		ryan.wood@exampleorg.com

Alert Actions & Roles for Searches

Identifying what alert actions are running and roles attached to those search processes.
Also the SPL that ran and original SIDs before normalization.

```
| union
 [ search (index=_internal sourcetype=scheduler sid="*") alert_actions=* NOT alert_actions=""
 | fields sourcetype, alert_actions, sid, savedsearch_name, user, app
 | eval alert_actions = split(alert_actions, ",")
 | `get_normalized_search_id(sid)`]

 [ search (index=_audit sourcetype=audittrail action=search info=completed search_id="scheduler_*")
 | rex field=_raw ",\sroles=('|\")(?<extracted_roles>(\w+\+)*\w+)('|\"),"
 | rex field=_raw
",\ssearch='(?<searchQuery>[\W\w\n]+)'((,\sautojoin=)|(\])|(,\sis_federated_search=)|(,\sincomplete_bucket_maps=)|(,\s[^\s=]+=))"
 | eval searchQuery = replace(searchQuery, "',\s[^\s=]+='[^']+$", "")
 | fields sourcetype, search_id, extracted_roles, savedsearch_name, user, app, searchQuery
 | eval extracted_roles = split(extracted_roles, "+")
 | `get_normalized_search_id(search_id)`]
| stats values(savedsearch_name) AS savedsearch_name, values(alert_actions) AS alert_actions, values(extracted_roles) AS roles,
values(user) AS user, values(app) AS app, latest(searchQuery) AS searchQuery,
values(sid) AS scheduler_sids, values(search_id) AS audittrail_sids BY search_id_normalized
```

Usecase - Normalizing SIDs

Alert Actions being run and the roles on searches running them.

`_internal scheduler + _audit audittrail`

search_id_normalized	values(sourcetype)	savedsearch_name	alert_actions	roles	user	app	searchQuery
1715130780_10887	audittrail scheduler	Sample Data Generation	email logevent lookup	gps_admin power tokens_auth user	ryan.wood@exampleorg.com	search	search index=_internal user=* | stats count by index | eval foo = "bar" | append [search index=_audit sourcetype=audittrail | stats count by index]

Usecase - Normalizing SIDs

Extracting SPL from audittrail

```
| union
 [ search (index=_internal sourcetype=scheduler sid="*") alert_actions=* NOT alert_actions=""
 | fields sourcetype, alert_actions, sid, savedsearch_name, user, app
 | eval alert_actions = split(alert_actions, ",")
 | `get_normalized_search_id(sid)`]

 [ search (index=_audit sourcetype=audittrail action=search info=completed search_id="scheduler_*")
 | rex field=_raw ",\sroles=('|\")(?<extracted_roles>(\w+\+)*\w+)('|\"),"
 | rex field=_raw
",\ssearch='(?<searchQuery>[\W\w\n]+)'((,\sautojoin=)|(\])|(,\sis_federated_search=)|(,\sincomplete_bucket_maps=)|(,\s[^\s=]+=))"
 | eval searchQuery = replace(searchQuery, "',\s[^\s=]+='[^']+$", "")
 | fields sourcetype, search_id, extracted_roles, savedsearch_name, user, app, searchQuery
 | eval extracted_roles = split(extracted_roles, "+")
 | `get_normalized_search_id(search_id)`]
| stats values(savedsearch_name) AS savedsearch_name, values(alert_actions) AS alert_actions, values(extracted_roles) AS roles,
values(user) AS user, values(app) AS app, latest(searchQuery) AS searchQuery, values(sid) AS scheduler_sids, values(search_id) AS
audittrail_sids BY search_id_normalized
```

Highlight - Regex for Audittrail

Extracting useful elements from audittrail events.

```
(index=_audit sourcetype=audittrail source="*audit.log*" action=search)
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
| rex field=_raw ",\ssavedsearch_name=\"(?<savedsearch_name>[^\"]+?)\""
| rex field=_raw ",\sapp=\"(?<app>[^\"]+)\""
| rex field=_raw ",\suser=(?<user>[^,]+)"
| rex field=_raw ",\sinfo=(?<info>[^,]+)"
| rex field=_raw
",\ssearch='(?<searchQuery>[\W\w\n]+)'((,\sautojoin=)|(\])|(,\sis_federated_search=)|(,\sincomplete_bucket_maps=)|(,\s[^\s=]+=))"
| eval searchQuery = replace(searchQuery, "',\s[^\s=]+='[^']+$", "")
```

Ensures these fields are not pulled from auto kv in SPL queries.

Usecase - Normalizing SIDs

Extracting SPL from audittrail

Event

```
Audit:[timestamp=05-08-2024 01:16:58.481, user=ryan.wood@exampleorg.com, action=search, info=completed,
search_id='scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887', has_error_warn=false, fully_c
ompleted_search=true, total_run_time=156.43, event_count=8872255, [...truncated...], savedsearch_name="Sample Data Generation", [...truncated...],
search_type=scheduled, roles='gps_admin+power+tokens_auth+user', search='search index=_internal user=*
| stats count by index
| append
    [search index=_audit sourcetype=audittrail
    | stats count by index]', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]
```

Collapse

host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail

Usecase - Normalizing SIDs

Top 10 Memory Consuming Searches with Execution Time & SPL Query

_introspection splunk_resource_usage + _audit audittrail

exec_time_readable	exec_time	search_id_normalized	savedsearch_name	user	app	total_mem_used	searchQuery
2024-05-07 19:54:59	1715126099	1715126040_10837	Sample Data Generation	ryan.wood@exampleorg.com	search	974.684	search index=_internal user=* | stats count by index | eval foo = "bar" | append [search index=_audit sourcetype=audittrail | stats count by index]

Usecase - Normalizing SIDs

Top 10 Memory Consuming Searches with Execution Time & SPL Query

_introspection splunk_resource_usage + _audit audittrail

exec_time_readable	exec_time	search_id_normalized	savedsearch_name	user	app	total_mem_used	searchQuery
2024-05-07 19:54:59	1715126099	1715126040_10837	Sample Data Generation	ryan.wood@exampleorg.com	search	974.684	search index=_internal user=* | stats count by index | eval foo = "bar" | append [search index=_audit sourcetype=audittrail | stats count by index]

audittrail_sids
scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837
subsearch_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837_1715126099.1

introspection_sids
remote_sh-i-000e778792a422e91.exampleorg-jg.splunkcloud.com_prd.ph0_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837
remote_sh-i-000e778792a422e91.exampleorg-jg.splunkcloud.com_prd.ph1_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837
remote_sh-i-000e778792a422e91.exampleorg-jg.splunkcloud.com_subsearch_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837_1715126099.1
scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715126040_10837

Top 10 Searches by Memory with Exec Time

And SPL Query added to output

```
(index=_introspection sourcetype=splunk_resource_usage data.search_props.sid="*")
| fields data.search_props.sid, data.mem_used, data.search_props.user, data.search_props.app
| rename data.* AS *, search_props.* AS *
``` Gather max value for each SID before normalizing SID and aggregating ```
| stats max(mem_used) AS max_mem_used BY sid, user, app
``` Sum so we get total mem_used including remote SIDs ```
| `get_normalized_search_id(sid)`
| fields - sid
| stats sum(max_mem_used) AS total_mem_used BY search_id_normalized, user, app
``` Filter to top 10 ```
| sort 10 - total_mem_used
``` Add audittrail data, also using normalized SID ```
| join type=outer search_id_normalized
   [search index=_audit sourcetype=audittrail action=search search_id=*
   | rex field=_raw
",\ssearch='(?<searchQuery>[\W\w\n]+)'((,\sautojoin=)|(\])|(,\sis_federated_search=)|(,\sincomplete_bucket_maps=)|(,\s[^\s=]+=))"
   | eval searchQuery = replace(searchQuery, "',\s[^\s=]+='[^']+$", "")
   | fields savedsearch_name, search_id, exec_time, searchQuery
   | `get_normalized_search_id(search_id)`
   | fields - search_id
   | stats latest(*) AS * BY search_id_normalized
   | fields - search_id]
| table exec_time, exec_time_readable, search_id_normalized, savedsearch_name, user, app, total_mem_used, searchQuery
| eval exec_time_readable = strftime(exec_time, "%Y-%m-%d %H:%M:%S")
```

Yes Yes, what about *Data Source Usage?*

Data Source Usage Reporting

.conf24

splunk>

Bring on the Data

Data Source Usage by Searches…

Identifying searches using…

- Indexes
- Sourcetypes
- Datamodels
- Lookups
- Eventtypes
- Macros

Data Source Usage by Searches...

during this talk.

Identifying searches using...

- Indexes
- Sourcetypes
- Datamodels
- Lookups
- Eventtypes
- Macros

Which indexes, sourcetypes are my searches using?

Usecase - Data Source Usage

Approached three ways:

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

search.log

- Ingesting search.log
  OR
- Using REST to:
  - Fetch search.log data from search artifacts

Usecase - Data Source Usage

Approached three ways:

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

Challenges:

- Not all search types

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

Challenges:

- Imprecise due to complexity

search.log

- Ingesting search.log
  OR
- Using REST to:
  - Fetch search.log data from search artifacts

Challenges:

- Ingest vs Visibility
- Point-in-Time

Data Source Usage

Instrumentation

- sourcetype=audittrail

Savedsearch in app=splunk_instrumentation:
instrumentation.usage.search.searchtelemetry.sourcetypeUsage

`usage.search.searchtelemetry.sourcetypeUsage`	Sourcetype usage.	<pre>{ [-]
		  sourcetypeUsage: [[-]
		  { [-]
		  http_event_collector_metrics: 1
		  kvstore: 1
		  mongod: 3
		  search_telemetry: 1
		  splunk_disk_objects: 1
		  splunk_resource_usage: 1
		  splunk_web_service: 3
		  splunkd: 11
		  splunkd_remote_searches: 3
		  splunkd_ui_access: 2
		  }
		  ]
		  }</pre>

Where's it come from?

Data Source Usage

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

```
index=_audit
sourcetype=audittrail
action=search info=completed
"sourcetype_count__*"
```

Default audittrail search info=completed events contain sourcetype usage notation:

Event

```
Audit:[timestamp=05-08-2024 01:16:58.481, user=ryan.wood@exampleorg.com, action=search, info=completed, search_id='sched
uler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887', has_error_warn=fals
e, fully_completed_search=true, total_run_time=156.43, event_count=8872255, result_count=2, available_count=0, scan_coun
t=131267360, drop_count=0, exec_time=1715130839, api_et=1714449600.000000000, api_lt=1715130480.000000000, api_index_et=
N/A, api_index_lt=N/A, search_et=1714449600.000000000, search_lt=1715130480.000000000, is_realtime=0, savedsearch_name
="Sample Data Generation", search_startup_time="85", is_prjob=true, is_flex_search=false, rate_limit_retry_enabled=fals
e, dispatch_artifact_bytes=270336, status_csv_bytes=8192, is_fss3=false, acceleration_id="D347BC42-6954-4328-9ABE-26B97C
9DBF62_search_ryan.wood@exampleorg.com_06508ace7b9522d3", app="search", provenance="scheduler", mode="historical_batch",
workload_pool=standard_perf, is_proxied=false, searched_buckets=45, eliminated_buckets=0, considered_events=131267360, t
otal_slices=365670, decompressed_slices=356464, duration.command.search.index=23852, invocations.command.search.index.bu
cketcache.hit=45, duration.command.search.index.bucketcache.hit=0, invocations.command.search.index.bucketcache.miss=0,
duration.command.search.index.bucketcache.miss=0, invocations.command.search.index.bucketcache.error=0, duration.comman
d.search.rawdata=76906, invocations.command.search.rawdata.bucketcache.hit=27, duration.command.search.rawdata.bucketcac
he.hit=0, invocations.command.search.rawdata.bucketcache.miss=0, duration.command.search.rawdata.bucketcache.miss=0, inv
ocations.command.search.rawdata.bucketcache.error=0,
sourcetype_count__scheduler=66950, sourcetype_count__secure_gateway_app_internal_log=1962, sourcetype_count__splunk_audi
t=317, sourcetype_count__splunk_python=1427, sourcetype_count__splunk_secure_gateway_modular_input.log=476, sourcetype_c
ount__splunk_secure_gateway_modular_input.log-too_small=183, sourcetype_count__splunk_web_access=47722, sourcetype_count
__splunk_web_service=1424, sourcetype_count__splunkd=34076, sourcetype_count__splunkd_access=8233885, sourcetype_count__
splunkd_ui_access=481442, search_type=scheduled, roles='gps_admin+power+tokens_auth+user', search='search index=_interna
l user=*
| stats count by index
| append
    [search index=_audit sourcetype=audittrail
    | stats count by index]', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]
```

Collapse

host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail

Data Source Usage

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

```
index=_audit
sourcetype=audittrail
action=search info=completed
"sourcetype_count__*"
```

Default audittrail search completed events contain sourcetype usage notation:

```
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
```

Data Source Usage

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

```
index=_audit
sourcetype=audittrail
action=search info=completed
"sourcetype_count__*"
```

Default audittrail search completed events contain sourcetype usage notation:

```
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
```

```
index=_audit sourcetype=audittrail action=search info=completed
"*sourcetype_count__*"
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
| rex field=_raw ",\ssavedsearch_name=\"(?<savedsearch_name>[^\"]+?)\""
| eval savedsearch_name = if(savedsearch_name="" OR isnull(savedsearch_name), "None", savedsearch_name)
| rex field=_raw ",\sapp=\"(?<app>[^\"]+)\""
| rex field=_raw ",\suser=(?<user>[^,]+)"
| stats sum(eventCount) AS total_eventCount, values(user) AS user_values, values(app) AS app_values,
values(provenance) AS provenance_values, dc(search_id) AS dc_searches BY savedsearch_name, sourcetype_val
```

Data Source Usage

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

```
index=_audit
sourcetype=audittrail
action=search info=completed
"sourcetype_count__*"
```


Default audittrail search completed events contain sourcetype usage notation:

```
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
```

```
index=_audit sourcetype=audittrail action=search info=completed
"*sourcetype_count__*"
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
| rex field=_raw ",\ssavedsearch_name=\"(?<savedsearch_name>[^\"]+?)\""
| eval savedsearch_name = if(savedsearch_name="" OR isnull(savedsearch_name), "None", savedsearch_name)
| rex field=_raw ",\sapp=\"(?<app>[^\"]+)\""
| rex field=_raw ",\suser=(?<user>[^,]+)"
| stats sum(eventCount) AS total_eventCount, values(user) AS user_values, values(app) AS app_values,
values(provenance) AS provenance_values, dc(search_id) AS dc_searches BY savedsearch_name, sourcetype_val
```

savedsearch_name	sourcetype_val	total_eventCount	user_values	app_values	provenance_values
None	search_log_events	2395727424	ryan.wood@exampleorg.com	search	N/A UI:Search
Sample Data Generation	splunk_python	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_web_service	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	scheduler	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_web_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd_ui_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	secure_gateway_app_internal_log	225360056	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_audit	109909761	ryan.wood@exampleorg.com	search	scheduler

Data Source Usage

Instrumentation

- sourcetype=audittrail

sourcetype_count__<sourcetype>

```
index=_audit
sourcetype=audittrail
action=search info=completed
"sourcetype_count__*"
```

Default audittrail search completed events contain sourcetype usage notation:

```
index=_audit sourcetype=audittrail action=search info=completed
"*sourcetype_count__*"
| rex field=_raw max_match=0 "sourcetype_count__(?<sourcetype_val>\w+)=(?<eventCount>\d+)"
| rex field=_raw ",\ssavedsearch_name=\"(?<savedsearch_name>[^\"]+?)\""
| eval savedsearch_name = if(savedsearch_name="" OR isnull(savedsearch_name), "None", savedsearch_name)
| rex field=_raw ",\sapp=\"(?<app>[^\"]+)\""
| rex field=_raw ",\suser=(?<user>[^,]+)"
| stats sum(eventCount) AS total_eventCount, values(user) AS user_values, values(app) AS app_values,
   values(provenance) AS provenance_values, dc(search_id) AS dc_searches BY savedsearch_name, sourcetype_val
```

savedsearch_name	sourcetype_val	total_eventCount	user_values	app_values	provenance_values
None	search_log_events	2395727424	ryan.wood@exampleorg.com	search	N/A UI:Search
Sample Data Generation	splunk_python	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_web_service	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	scheduler	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_web_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunkd_ui_access	225387831	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	secure_gateway_app_internal_log	225360056	ryan.wood@exampleorg.com	search	scheduler
Sample Data Generation	splunk_audit	109909761	ryan.wood@exampleorg.com	search	scheduler

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

```
index=_audit sourcetype=audittrail action=search
```

Event

Audit:[timestamp=05-08-2024 01:16:58.481, user=ryan.wood@exampleorg.com, action=search, info=completed, search_id='scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887', has_error_warn=false, ompleted_search=true, total_run_time=156.43, event_count=8872255, [...truncated...], savedsearch_name="Sample Data Generation", [...truncat search_type=scheduled, roles='gps_admin+power+tokens_auth+user', search='search index=_internal user=* | stats count by index | append [search index=_audit sourcetype=audittrail | stats count by index]', incomplete_bucket_maps='false', is_federated_search=0, is_fsh_remote_search=0]

Collapse

host = examplecloudstack.splunkcloud.com | source = audittrail | sourcetype = audittrail

```
index=_internal sourcetype=splunkd_remote_searches
```

Event

05-08-2024 01:16:36.026 +0000 INFO StreamedSearch - Streamed search connection closed: search_id=remote_sh-i-000e778792a422e91.exampleorg-jg.splu prd.ph0_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5d4f313530111a124_at_1715130780_10887, server=sh-i-000e778792a422e91.exa plunkcloud.com, active_searches=1, elapsedTime=151.465, search='litsearch (index=_internal ((sourcetype=DhcpSrvLog msdhcp_user="*") OR (sourcetype="WMI:UserAccounts" Name="*") OR (sourcetype="WMI:WinEve ation" User="*") OR (sourcetype=ps USER="*") OR (sourcetype=top USER="*") OR user="*" OR (sourcetype=audittrail (uid="*" OR user_id="*" OR user="* etype=linux_audit OR sourcetype="o365:cas:api" OR sourcetype="o365:management:activity" OR source="WinEventLog:Security" OR source="WinEventLog:Sy rce="XmlWinEventLog:Security")) | litsearch (index=_internal user=*) | addinfo type=count label=prereport_events track_fieldmeta_events=true | f lorder=t "index" "prestats_reserved_*" "psrsvd_*" | prestats count by index | rdout partition_method="hash" num_of_intermediaries="2" hash_keys= r_sensitive="0"', savedsearch_name="Sample Data Generation", drop_count=0, scan_count=43648633, eliminated_buckets=0, considered_events=43648633, _slices=118727, events_count=43648633, total_slices=121260, considered_buckets=14, search_rawdata_bucketcache_error=0, search_rawdata_bucketcache ch_index_bucketcache_error=0, search_index_bucketcache_hit=14, search_index_bucketcache_miss=0, search_rawdata_bucketcache_hit=8, search_rawdata_ iss_wait=0.000, search_index_bucketcache_miss_wait=0.000

host = examplecloudstack.splunkcloud.com | source = /opt/splunk/var/log/splunk/remote_searches.log | sourcetype = splunkd_remote_searches

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

Challenges:

- SPL can be any string format - complex patterns required
- Macros, Eventtypes
- Multiple formats for specifying index/sourcetype
- Handling wildcard references
- Subsearches
- Unexpected bits!

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

Challenges:

- SPL can be any string format - complex patterns required
- Macros, Eventtypes
- Multiple formats for specifying index/sourcetype
- Handling wildcard references
- Subsearches
- Unexpected bits!

IndexerLevel - RemoteSearches Indexes Stats

Save | Save As | View | Creat

```
   index=_internal sourcetype=splunkd_remote_searches source="/opt/splunk/var/log/splunk/remote_searches.log" terminated: OR closed:
| rex "(?s) elapsedTime=(?P<elapsedTime>[0-9\.]+), search='(?P<search>.*?)(', savedsearch_name|\", drop_count=\d+)"
| regex search!="^(pretypeahead|copybuckets)"
| rex "drop_count=[0-9]+, scan_count=(?P<scan_count>[0-9]+)"
| rex "total_slices=[0-9]+, considered_buckets=(?P<considered_count>[0-9]+)"
| rex "(,|}\.\.\.) savedsearch_name=\"(?P<savedsearch_name>[^\"]*)\","
| rex "(terminated|closed): search_id=(?P<search_id>[^,]+)"
| regex search="^(litsearch|mcatalog|mstats|mlitsearch|litmstats|tstats|presummarize)"
| rex field=search max_match=50 "(?s)\|?\s*(mlitsearch)\s+.*?\[(?P<subsearch>.*?)\]\s*(\||$)"
| rex field=search "(?s)(?P<prepipe>\s*\|?([^\|]+))"
| nomv subsearch
| eval subsearch=if(isnull(subsearch),"",subsearch)
| eval prepipe = prepipe . " " . subsearch
| eval search=prepipe
| search `comment("The (index=* OR index=_*) index=<specific index> is a common use case for enterprise security, also some individuals like doing a similar trick so remove the index=*... as this is not a
   wildcard index search")`
| rex field=search "(?P<esstylewildcard>\(\s*index=\*\s+OR\s+index=_\*\s*\))"
| rex mode=sed field=search "s/search index=\s*\S+\s+index\s*=/search index=/"
| search `comment("Extract out index= or index IN (a,b,c) but avoid NOT index in (...) and NOT index=... and also NOT (...anything) statements")`
| rex field=search "(?s)(NOT\s+index(\s*=\s*|::)[^ ]+)|(NOT\s+\([^\)]+\))|(index(\s*=\s*|::)(?P<indexregex>[\*A-Za-z0-9-_]+))" max_match=50
| rex field=search "(?s)(NOT\s+index(\s*=\s*|::)[^ ]+)|(NOT\s+\([^\)]+\))|(index(\s*=\s*|::)\"?(?P<indexregex2>[\*A-Za-z0-9-_]+))" max_match=50
| rex field=search "\s+(?P<skipping>\.\.\.\{skipping \d+ bytes\}\.\.\.)"
| search `comment("If skipping is in the logs as in index=abc- ...{skipping 46464 bytes}..., then drop the last index found in the regex as it is likely invalid")`
| eval indexregex=if(isnotnull(skipping),mvindex(indexregex,0,-2),indexregex)
| eval indexregex2=if(isnotnull(skipping),mvindex(indexregex2,0,-2),indexregex2)
| eval indexes=mvappend(indexregex,indexregex2)
| eval indexes=if(isnotnull(esstylewildcard),mvfilter(NOT match(indexes,"^_?\*$")),indexes)
| eval multi=if(mvcount(mvdedup(indexes))>1,"true","false")
| rex field=search_id "^remote_(?P<sid>.*)"
| rex "search_id=[^,]+,\s+server=(?P<server>[^,]+)"
| eval server_with_underscore = server. "_"
| eval sid=replace(sid, server_with_underscore, "")
| eval search_head=server
```

Data Source Usage - Regex

Using regex to extract index and sourcetypes from SPL queries.

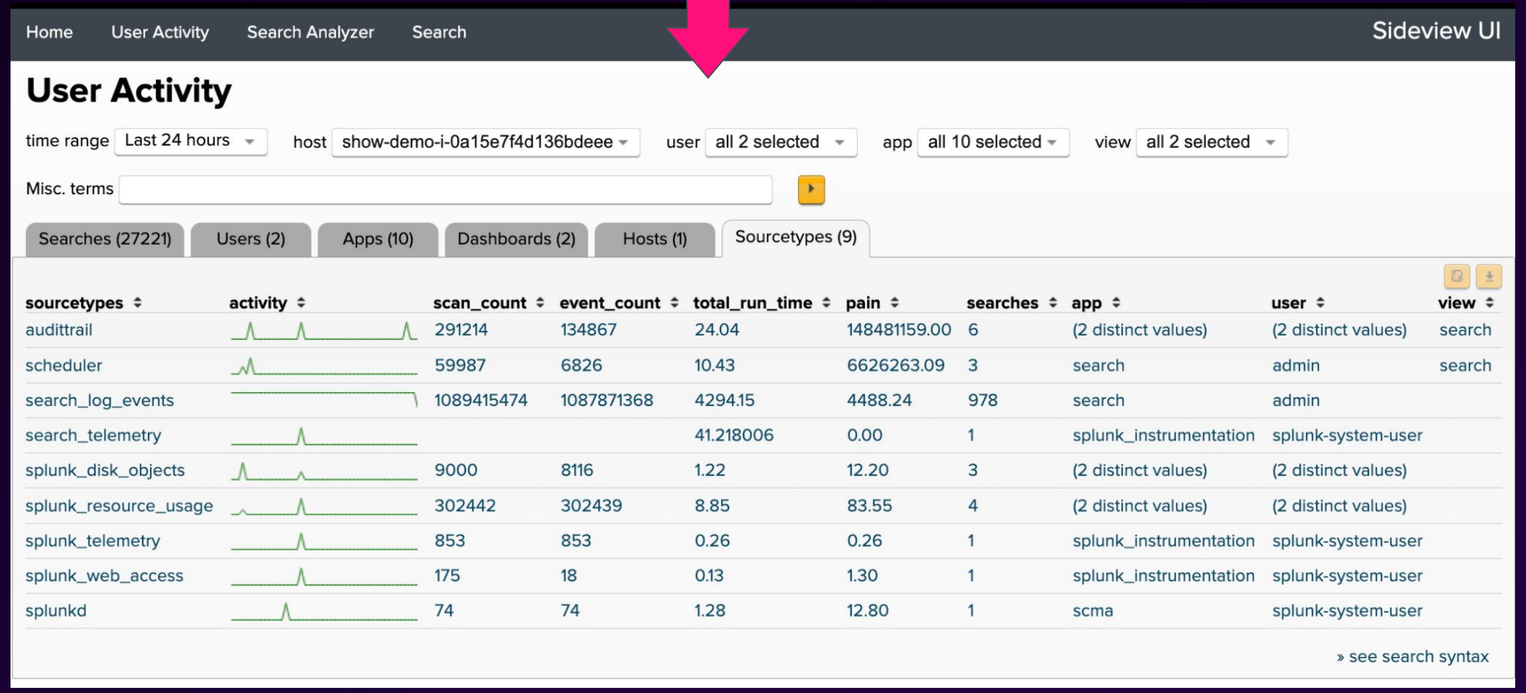
Some great Splunk apps with attempts at doing this:

- **Sideview UI**
  - Dashboard:
    User Activity

- **Splunk Cloud Migration Assessment**
  - Savedsearch:
    search_index_statistics

- **Admin Pilot For Splunk**
  - Macros:
    get_index_reference(1),
    get_sourcetype_reference(1)

- **Alerts for Splunk Admins**
  - Savedsearches:
    IndexerLevel - RemoteSearches Indexes Stats
    SearchHeadLevel - indexes per savedsearch



Data Source Usage - Regex

Using regex to extract index and sourcetypes from SPL queries.

Some great Splunk apps with attempts at doing this:

- **Sideview UI**
  - *https://splunkbase.splunk.com/app/6449*
- **Splunk Cloud Migration Assessment**
  - *https://splunkbase.splunk.com/app/4974*
- **Admin Pilot For Splunk**
  - *https://splunkbase.splunk.com/app/6489*
- **Alerts for Splunk Admins**
  - *https://splunkbase.splunk.com/app/3796*

App Highlights - Admin Pilot for Splunk

Macro: *get_index_reference(1)*

```
rex field=$field$ max_match=100 "index\s*=[\s\"]?(?<Index_Reference1>[a-z0-9-_*]+)[\s\"]"
| rex field=$field$ max_match=100 "index\s*=\s*\"?(?<Index_Reference2>_[a-z]+)[\s\"]"
| rex field=$field$ max_match=100 "index=(?<Index_Reference3>[a-z0-9-_*]+)"
| rex field=$field$ max_match=100 "index=(?<Index_Reference4>[`a-z0-9-_*]+)"
| rex field=$field$ max_match=100 "index=(?<Index_Reference5>\w+)"
| rex field=$field$ max_match=100 "\|\s*collect\s+(?<Index_Reference6>`\S+`)"
| eval Index_Reference =
mvdedup(trim(mvappend(Index_Reference1,Index_Reference2,Index_Reference3,Index_Reference4,Index_Reference5,Index_Reference6)))
| eval Index_Reference = if(match($field$, "index\s*=\s*_\*"), "all-internal-indexes", Index_Reference)
| eval Index_Reference = if(match($field$, "index\s*=\s*\*|index=\"\*\""), "all-indexes", Index_Reference)
| eval Index_Reference = mvfilter((!match(Index_Reference,"^1$")))
| eval Index_Reference = if(isnull(Index_Reference) OR Index_Reference="", "no-index-reference", Index_Reference)
| fields - Index_Reference1 Index_Reference2 Index_Reference3 Index_Reference4 Index_Reference5 Index_Reference6
```

App Highlights - Admin Pilot for Splunk

Macro: *get_sourcetype_reference(1)*

```
rex field=$field$ max_match=100 "sourcetype\s*!?=\s*(?<Sourcetype_Reference>.*?)[\s]"
| rex field=Sourcetype_Reference mode=sed "s/[\s\",=()|]//g"
| eval Sourcetype_Reference = if(Sourcetype_Reference = "" OR match(Sourcetype_Reference, "\$") OR isnull(Sourcetype_Reference),
"no-sourcetype-reference", Sourcetype_Reference)
| eval Sourcetype_Reference = if(match($field$, "sourcetype\s*=\s*\*|sourcetype=\"\*\""), "all-sourcetypes", Sourcetype_Reference)
```

App Highlights - Admin Pilot for Splunk

Macro: *get_source_reference(1)*

```
rex field=$field$ max_match=100 "source\\s*=\\s*(?<Source_Reference1>.*?)[\\s\"\\|]"
| rex field=Source_Reference1 mode=sed "s/^[\\s$?><()\\\\,^=]*//g"
| rex field=$field$ max_match=100 "source\\s+IN\\s*\\((?<Source_Reference2>.*?)\\)"
| makemv delim="," Source_Reference2
| rex field=Source_Reference2 mode=sed "s/^[\\s$?><()\\\\,^=]*//g"
| eval Source_Reference=coalesce(Source_Reference1,Source_Reference2),
   Source_Reference=mvfilter((!
match(Source_Reference,"^source|^\"|^ifisnull|^if\(|\.\*|^Mvindex|^lower|^mvfilter|^mvsort|^spath|^trim"))),
   Source_Reference=mvdedup(mvsort(Source_Reference)), Source_Reference=if(((Source_Reference == "") OR
isnull(Source_Reference)),"no-source-reference",Source_Reference),
   Source_Reference=if(match($field$,"source\\s*=\\s*\\*|source=\"\\*\""),"all-sources", Source_Reference)
| fields - Source_Reference1 Source_Reference2
| fillnull value="no-source-reference" Source_Reference
```

App Highlights - Admin Pilot for Splunk

Macro: *get_eventtype_reference(1)*

```
rex field=$field$ max_match=100 "eventtype\\s*=\\s*(?<Eventtype_Reference1>.*?)[\\s\"\\|]"
| rex field=Eventtype_Reference1 mode=sed "s/^[\\s$?><()\\\\,^=\\]\\[+]*//g"
| rex field=$field$ max_match=100 "eventtype\\s+IN\\s*\\((?<Eventtype_Reference2>.*?)\\)"
| makemv delim="," Eventtype_Reference2
| rex field=Eventtype_Reference2 mode=sed "s/^[\\s$?><()\\\\,^=]*//g"
| eval Eventtype_Reference=coalesce(Eventtype_Reference1,Eventtype_Reference2),
Eventtype_Reference=mvfilter((! match(Eventtype_Reference,"^eventtype|^trim|ifisnull|^\"$"))),
Eventtype_Reference=mvdedup(mvsort(Eventtype_Reference))
| eval Eventtype_Reference=if(((Eventtype_Reference == "") OR isnull(Eventtype_Reference)),"no-eventtype-reference",Eventtype_Reference)
| fields - Eventtype_Reference1 Eventtype_Reference2
| fillnull value="no-eventtype-reference" Eventtype_Reference
```

App Highlights - Admin Pilot for Splunk

Macro: *get_macro_reference(1)*

```
rex field=$field$ max_match=100 "`(?<Macro_Reference>\p{Any}+?)`"
| rex field=Macro_Reference mode=sed "s/\"|\s+//g"
| eval Macro_Reference = mvfilter((! match(Macro_Reference,"^\||^\)|^:|^\[|^comment|^ia4s_comment")))
| eval Macro_Reference = if(((Macro_Reference == "") OR isnull(Macro_Reference)), "no-macro-reference", Macro_Reference)
| mvexpand Macro_Reference
| rex field=Macro_Reference max_match=100 "(?<Macro_Name>^[a-zA-Z0-9_-]+)"
| rex field=Macro_Reference max_match=100 "\((?<Macro_Args>.*?)\)"
| makemv delim="," Macro_Args
| eval Macro_Args_Count = mvcount(Macro_Args)
| eval Macro_Title = if (isnull(Macro_Args_Count), Macro_Name, Macro_Name . "(" . Macro_Args_Count . ")")
| eval Macro_Title = if(((Macro_Title == "") OR isnull(Macro_Title)), "no-macro-title", Macro_Title)
| fields - Macro_Reference1 Macro_Name Macro_Args Macro_Args_Count
```

App Highlights - Admin Pilot for Splunk

Macro: *get_lookup_reference(1)*

```
rex field=$field$ max_match=100 "\|\s*inputlookup\s+(?<Input_Lookup>[^|]+)"
| rex field=$field$ max_match=100 "\|\s*from\s+inputlookup:(?<From_Input_Lookup>[^|]+)"
| rex field=$field$ max_match=100 "\|\s*from\s+lookup:(?<From_Lookup>[^|]+)"
| rex field=$field$ max_match=100 "\|\s*outputlookup\s+(?<Output_Lookup>[^|]+)"
| rex field=$field$ max_match=100 "\|\s*lookup\s+(?<Lookup_Lookup>[^|\s]+)"
| eval Input_Lookup = "Input_Lookup:".Input_Lookup , From_Input_Lookup = "From_Input_Lookup:".From_Input_Lookup, From_Lookup =
"From_Lookup:".From_Lookup, Output_Lookup = "Output_Lookup:".Output_Lookup, Lookup_Lookup = "Lookup_Lookup:".Lookup_Lookup
| eval Lookup_Reference=mvsort(mvdedup(lower(mvappend(Lookup_Lookup,Input_Lookup,From_Lookup,From_Input_Lookup,Output_Lookup))))
| rex field=Lookup_Reference mode=sed
"s/\"|append=\w+|create_empty=\w+|createinapp=\w+|override_if_empty=\w+|event_time_field=\w+|output_format=\w+|local=\w+|update=\w+|key_f
ield=\w+|enabled=\w+|max=\w+|type=\w+|\s+where\s+.*|\$//g"
```| rex field=Lookup_Reference mode=sed "s/(\s|\]).*$//g" ")```
| eval Lookup_Reference=if(((Lookup_Reference == "") OR isnull(Lookup_Reference)),"no-lookup-reference",
mvsort(mvdedup(trim(Lookup_Reference))))
| fields - Input_Lookup,From_Input_Lookup,From_Lookup,Output_Lookup,Lookup_Lookup
| fillnull value="no-lookup-reference" Lookup_Reference
```
```

App Highlights - Admin Pilot for Splunk

Macro: *get_datamodel_reference(1)*

```
| rex field=$field$ max_match=100
"[fF][rR][oO][mM]\s*[dD][aA][tT][aA][mM][oO][dD][eE][lL][:=](?<Datamodel_Reference1>.*?)\s"
| rex field=$field$ max_match=100 "\|\s*(datamodel|datamodelsimple)\s+(?<Datamodel_Reference2>.*?)\s"
| eval Datamodel_Reference=coalesce(Datamodel_Reference1,Datamodel_Reference2)
| rex field=Datamodel_Reference mode=sed "s/\"//g"
| eval Datamodel_Reference = mvfilter( ! match(Datamodel_Reference, "^\$|type=|^\|") )
| eval Datamodel_Reference=if(((Datamodel_Reference == "") OR isnull(Datamodel_Reference)),"no-datamodel-reference",
mvdedup(mvsort(Datamodel_Reference)))
| fields - Datamodel_Reference1 Datamodel_Reference2
| fillnull value="no-datamodel-reference" Datamodel_Reference
```

App Highlights - Admin Pilot for Splunk

Macro: *get_dashboard_reference(1)*

```
rex field=$field$ max_match=100 "href=\"(?<Dashboard_Reference>\w+)\">"
| eval Dashboard_Reference=mvdedup(mvsort(Dashboard_Reference)), Dashboard_Reference=if(((Dashboard_Reference == "")
OR isnull(Dashboard_Reference)),"no-dashboard-reference",mvdedup(mvsort(Dashboard_Reference)))
| fillnull value="no-dashboard-reference" Dashboard_Reference
```

App Highlights - Alerts for Splunk Admins

Using regex to extract index and sourcetypes from SPL queries.

Notable Savedsearches in Alerts for Splunk Admins for data source usage:

- SearchHeadLevel - indexes per savedsearch
- SearchHeadLevel - Search Queries summary exact match
- IndexerLevel - RemoteSearches Indexes Stats
- IndexerLevel - RemoteSearches Indexes Stats Wilcard
- IndexerLevel - RemoteSearches - lookup usage
- IndexerLevel - DataModel Acceleration - Indexes in use
- SearchHeadLevel - Scheduled searches not specifying an index
- SearchHeadLevel - Scheduled searches not specifying an index macro version
- SearchHeadLevel - Users exceeding the disk quota
- SearchHeadLevel - Splunk Users Violating the Search Quota

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

```
| rest
/servicesNS/-/-/search/jobs
```

Using REST:

`/search/jobs`

Benefits:

- Very useful information not in default indexes
- *keywords* field contains only the info we care about

Challenges:

- Only available while the job artifact is alive
- Subsearch artifacts do not receive the parent artifact's TTL

https://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTsearch#search.2Fjobs

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

```
| rest
/servicesNS/-/-/search/jobs
```

Using REST:
/search/jobs

search/jobs

```
https://<host>:<mPort>/services/search/jobs
```

List search jobs.

For more information about this and other search endpoints, see Creating searches using the REST API in the *REST API Tutorial*.

keywords	All positive keywords used by this search. A positive keyword is a keyword that is not in a NOT clause.

https://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTsearch#search.2Fjobs

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

```
| rest
/servicesNS/-/-/search/jobs
```

Using REST:
/search/jobs

search/jobs

```
https://<host>:<mPort>/services/search/jobs
```

List search jobs.

For more information about this and other search endpoints, see Creating searches using the REST API in the *REST API Tutorial*.

keywords	All positive keywords used by this search. A positive keyword is a keyword that is not in a NOT clause.

eventSearch	keywords
search (sourcetype="scheduler" (NOT index="foo" OR NOT sourcetype=bar) (index="_internal" OR index="aws" OR index="botsv4"))	index::_internal index::aws index::botsv4 sourcetype::scheduler

https://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTsearch#search.2Fjobs

Data Source Usage

Regex on SPL

- Using regex to extract index and sourcetype from SPL queries in:
  - audittrail
  - remote_searches.log
  - /search/jobs REST API

```
| rest
/servicesNS/-/-/search/jobs
```

Using REST:

/search/jobs

```
| rest /servicesNS/-/-/search/jobs splunk_server=*
| rename title AS searchQuery
| table label, sid, searchQuery, eventSearch, keywords
```

label	sid	eventSearch	keywords
Sample Data Generation - Add NOT	scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5ca9bbe8a8c6fca50_at_1715209380_11793	search (index=_internal user=* NOT sourcetype=splunkd_ui_access NOT sourcetype="*modular_input*" NOT index=_introspection)	index::_internal user::*
	subsearch_scheduler_cnlhbi53b29kQGd1aWRlcG9pbnRzZWN1cml0eS5jb20__search__RMD5ca9bbe8a8c6fca50_at_1715209380_11793_1715209410.13430	search (index=_audit sourcetype=audittrail)	index::_audit sourcetype::audittrail

https://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTsearch#search.2Fjobs

Utility SPL - search/jobs Regex Identification

Identifying index & sourcetype usage via search/jobs *keywords* field

```
rex field=$field$ max_match=100 "href=\"(?<Dashboard_Reference>\w+)\">"
| eval Dashboard_Reference=mvdedup(mvsort(Dashboard_Reference)), Dashboard_Reference=if(((Dashboard_Reference == "")
OR isnull(Dashboard_Reference)),"no-dashboard-reference",mvdedup(mvsort(Dashboard_Reference)))
| fillnull value="no-dashboard-reference" Dashboard_Reference
```

What about search.log?

Data Source Usage

search.log

- Ingesting search.log

  OR

- Using REST to:
  - Fetch search.log data from search artifacts

search.log

Benefits:

- Accurate - works for searches without index reference
  - Avoids abstraction problem with default indexes, eventtypes, macros
- Consistent - as of v9.0 all search artifacts contain these lines

Challenges:

- Takes time to build "proper" insight
  - Users aren't always consistent:
    - Investigation into Incidents/Issues
    - App validation/QA Processes
  - Infrequent-use data, like Compliance
- Ingest Size of search.log

Data Source Usage

search.log

- Ingesting search.log

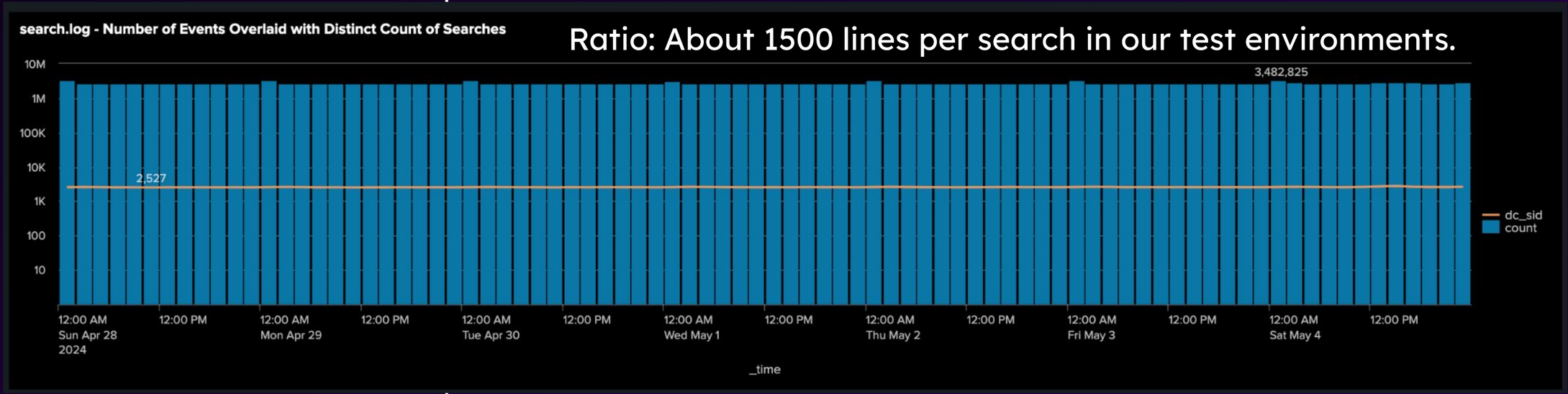
  OR

- Using REST to:
  - Fetch search.log data from search artifacts

search.log

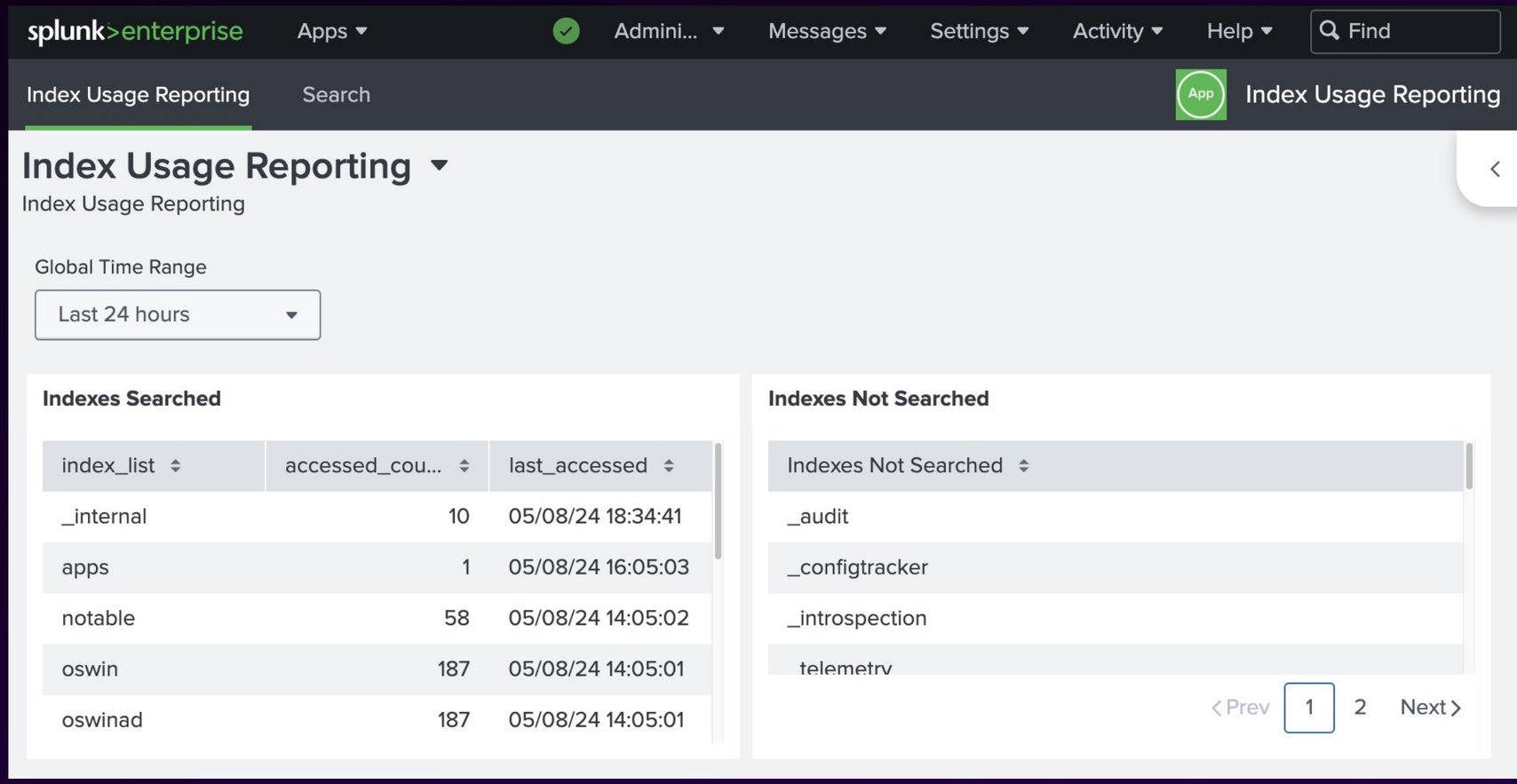
search.log contains a *ton* of useful information about searches

… but it's also a ton of data to ingest.



App Highlight: Index Usage Reporting

**Splunk Enterprise Only - Located on Github



*Credit to David Paper

https://github.com/dpaper-splunk/public/tree/master/apps/IUR

App Highlight: Index Usage Reporting

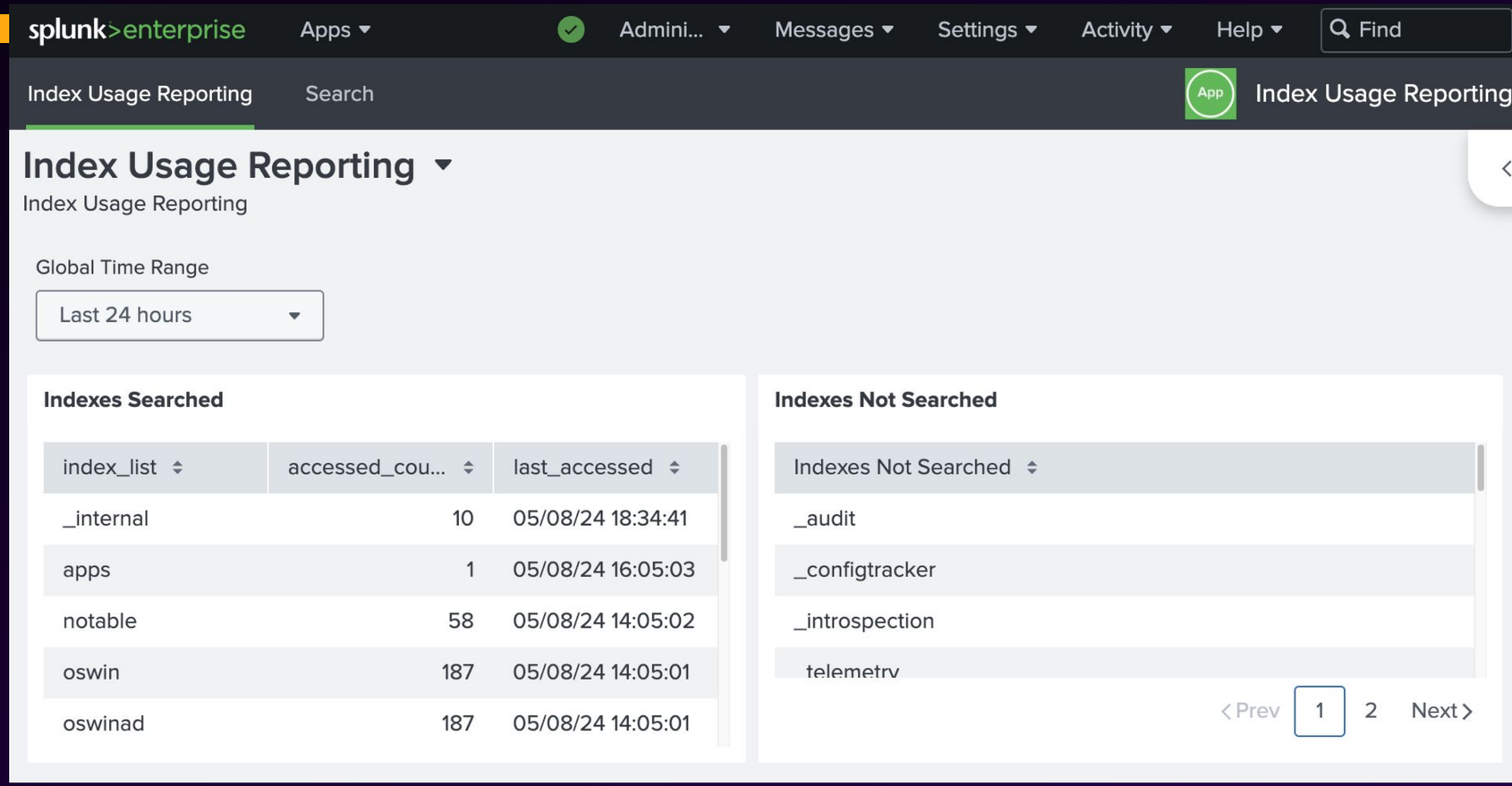
**Splunk Enterprise Only - Located on Github

Note: this is not "feature complete"

- Ingests search.log from dispatch directory
  - Populates *index=_internal*

- Uses search.log line to identify index:

`"IndexScopedSearch is called"`



*Credit to David Paper

https://github.com/dpaper-splunk/public/tree/master/apps/IUR

Data Source Usage

search.log

- Ingesting search.log

  OR

- Using REST to:
  - Fetch search.log data from search artifacts
  - Parse search/jobs keywords

search.log

"IndexScopedSearch" event lines

Event

05-02-2024 23:55:02.623 INFO IndexScopedSearch [1319756 localCollectorThread] - IndexScopedSearch is called for index = notable, et = 1714532400.000000000, lt = 1714708500.000000000, index_et = -9223372036854775808.000000000, index_lt = 9223372036854775807.999999000, noRead = FALSE

host = examplehost | source = /opt/splunk/var/run/splunk/dispatch/scheduler__admin__SplunkEnterpriseSecur... | sourcetype = search_logs

05-02-2024 23:54:01.272 INFO IndexScopedSearch [1315741 localCollectorThread] - IndexScopedSearch is called for index = wineventlog_security, et = 1714707540.000000000, lt = 1714708440.000000000, index_et = 1714707540.000000000, index_lt = 1714708440.000000000, noRead = FALSE

host = examplehost | source = /opt/splunk/var/run/splunk/dispatch/scheduler__admin__ThreatHunting__RMD... | sourcetype = search_logs

05-02-2024 23:54:01.272 INFO IndexScopedSearch [1315741 localCollectorThread] - IndexScopedSearch is called for index = wineventlog_perfmon, et = 1714707540.000000000, lt = 1714708440.000000000, index_et = 1714707540.000000000, index_lt = 1714708440.000000000, noRead = FALSE

host = examplehost | source = /opt/splunk/var/run/splunk/dispatch/scheduler__admin__ThreatHunting__RMD... | sourcetype = search_logs

BREAK - End of Technical Section

- *Placeholder to mark section end*

_telemetry additional info

- Enable instrumentation at Settings->Instrumentation
- We found usage.search.searchtelemetry.sourcetypeUsage events that show how many times each sourcetype was used, but they were only present on one of our test environments so that information is not included here.
  - index=_telemetry | spath | search "data.component"="usage.search.searchtelemetry.sourcetypeUsage"
  - This data appears to be collected at midnight each day
  - {"datetime": "2024-05-05 04:16:39,868", "log_level": "INFO", "component": "TelemetryCloudData", "data": {"data": {"sourcetypeUsage": [{"http_event_collector_metrics": 51, "linux_secure": 75, "splunk_resource_usage": 228, "stash": 1252, "splunkd": 256, "splunk_web_access": 97, "splunkd_access": 222, "splunkd_ui_access": 191, "audittrail": 208, "splunk_telemetry": 51, "scheduler": 141, "splunk_disk_objects": 23, "cloud_monitoring_console": 10, "cloudgateway_app_internal_log": 15, "dmc_agent_access": 10, "get_scs_tokens_2": 15, "get_scs_tokens_3": 14, "get_scs_tokens_too_small": 16, "itsi_internal_log": 18, "itsi_license_checker_too_small": 15, "kvstore": 3, "littlehelper_rest_2": 15, "mongod": 16, "phantom_retry_2": 15, "python_upgrade_readiness_app": 8, "sa_itsi_at_recommendations_2": 7, "secure_gateway_app_internal_log": 15, "splunk_app_ar_internal_log": 15, "splunk_app_cloudgateway_modular_input_3": 5, "splunk_app_cloudgateway_subscription_search_requests_2": 15, "splunk_app_stream_2": 15, "splunk_btool": 16, "splunk_cloud_telemetry": 3, "splunk_python": 16, "splunk_search_messages": 16, "splunk_ta_google_cloudplatform_ucc_lib_2": 15, "splunk_ta_google_cloudplatform_util_2": 10, "splunk_web_service": 14, "splunkd_remote_searches": 18, "sup_pkg_identity_stdout_2": 16, "supervisor_2": 16, "victorops_alert_recovery_2": 15, "mlspl_2": 2, "splunk_app_cloudgateway_modular_input_2": 10, "ssg_deep_link_dashboard_modular_input_too_small": 10, "sup_pkg_identity_stdout_too_small": 12, "search_telemetry": 2, "data_manager_global_config_lookup_2": 2, "data_manager_telemetry_lookup_log": 2, "splunk_secure_gateway_modular_input_log_too_small": 2}]}, "timestamp": 1714882568, "component": "usage.search.searchtelemetry.sourcetypeUsage", "date": "2024-05-04", "eventID": "17CADD89-9CDF-4F35-82E3-A8F4751434A3", "batchNum": 7, "visibility": "anonymous,support", "executionID": "A4FF7549DF2E73D7B53707C3E5442F", "transactionID": "B7C35195-EF21-44B0-AD7B-A02C84599763", "deploymentID": "CLOUD-f0589c718a53de653fda94d284ef369eeda659fcea68a57d9230797fe938aed8", "version": "4"}}

Lookups used by searches

Why bother?

- Know which lookup tables are being used in a search
- Clean up unused lookups
- Reduce the size of the knowledge bundle

Lookups used by searches

What the data looks like

Savedsearch Name	Query
CloudTrail Base Search	`aws-cloudtrail((aws_account_id="*"), (region="**") )` | lookup unauthorized_errorCode errorCode OUTPUT Unauthorized
host_role_generator	index="summary" source="splunk-svc-consumer" | stats values(search_head_names) as hrole by usage_source | rename usage_source as host | outputlookup hostrole.csv
Metrics Selectable Lookup	| inputlookup dropdowns.csv | stats values(host) as host by unix_category unix_group
splunk_identities_custom_gap_report	| from lookup:splunk_rest_identities_kv_store_lookup | search NOT [| inputlookup splunk_identities_custom_kv_store_lookup | fields identity] | table …

Lookups used by searches

How?

```
| rest splunk_server=local /servicesNS/-/-/saved/searches
| fields title eai:acl.app search
| append [| rest splunk_server=local /servicesNS/-/-/data/ui/views
    | fields title eai:acl.app eai:data | rename eai:data as search ]
| regex search="\|\s*(?:inputlookup|lookup|outputlookup|from\s+lookup:)"
| rex field=search max_match=0
"\|\s*(?:inputlookup|lookup|outputlookup|from\s+lookup:)\s+(?:append\s*=\s*\w+\s+)?(?<lookup>[\w\.]+)"
| stats values(lookup) as lookups by eai:acl.app
| fields lookups | mvexpand lookups | dedup lookups | sort + lookups
```

Lookups used by searches

How?

| rest splunk_server=local /servicesNS/-/-/saved/searches ``` Fetch scheduled searches ```

| fields title eai:acl.app search

| append [| rest splunk_server=local /servicesNS/-/-/data/ui/views ``` Fetch dashboards ```

    | fields title eai:acl.app eai:data | rename eai:data as search]

``` Look for queries that use lookup commands ```

| regex search="\|\s*(?:inputlookup|lookup|outputlookup|from\s+lookup:)"

``` Extract the lookup name ```

| rex field=search max_match=0 "\|\s*(?:inputlookup|lookup|outputlookup|from\s+lookup:)\s+(?:append\s*=\s*\w+\s+)?(?<lookup>[\w\.]+)"

``` List the lookups used by each app ```

| stats values(lookup) as lookups by eai:acl.app

``` Optional - Show distinct lookup names ```

| fields lookups | mvexpand lookups | dedup lookups | sort + lookups

Datamodels used by searches

Why worry?

- Unused datamodels don’t need to be accelerated, saving search slots and SVCs
- Know which datamodels should remain accelerated

Datamodels used by searches

What the data looks like

Search/Dashboard Name	Query
Critical Severity Intrusion	| tstats local=false summariesonly=true allow_old_summaries=true count from datamodel=Intrusion_Detection.IDS_Attacks where …
DA-ITSI-LB-Upstream_Members	| datamodel LoadBalancer Connections search | search host="$host$"
compliance	<form … <query>| from datamodel:"Malware"."Malware_Operations" …

Datamodels used by searches

How to do it

```
| rest splunk_server=local /servicesNS/-/-/saved/searches
| fields title eai:acl.app search
| append [| rest splunk_server=local /servicesNS/-/-/data/ui/views
| fields title eai:acl.app eai:data | rename eai:data as search ]
| regex search="\|\s*datamodel[\s=]"
| rex field=search max_match=0 "(?:\|\s*|\s+from\s+)datamodel[\s=:]\\\"?(?<dm>[\w]+)\\\"?"
| where isnotnull(dm)
| stats values(dm) as DMs by eai:acl.app
| table eai:acl.app DMs
| rename eai:acl.app as App, DMs as DataModels
```

Datamodels used by searches

How to do it

Some events that we don’t handle include:

- DM name in a token (as in a dashboard or the ‘map’ command) is ignored
- The ‘datamodel’ command with no arguments (which displays all DMs) is ignored
- Commented ‘datamodel’, ‘from datamodel’, or ‘tstats… from datamodel’ commands are treated as un-commented

```
| rest splunk_server=local /servicesNS/-/-/saved/searches  ``` Fetch scheduled searches ```
| fields title eai:acl.app search
| append [| rest splunk_server=local /servicesNS/-/-/data/ui/views ``` Fetch dashboards ```
| fields title eai:acl.app eai:data | rename eai:data as search ]
``` We only care about searches that use the ‘datamodel’ command, the ‘from datamodel’, or the ‘from datamodel’ option to the ‘tstats’  command ```
| regex search="\|\s*datamodel[\s=]"
``` Extract the DM name ```
| rex field=search max_match=0 "(?:\|\s*|\s+from\s+)datamodel[\s=:]\\\"?(?<dm>[\w]+)\\\"?"
``` Ignore events we couldn’t get a DM name from ```
| where isnotnull(dm)
``` List the DM names by app ```
| stats values(dm) as DMs by eai:acl.app
| table eai:acl.app DMs
| rename eai:acl.app as App, DMs as DataModels
```

Unused Sourcetypes

Why do we care?

- If it isn’t used then maybe it doesn’t have to be ingested
- If it isn’t used and has to be ingested then maybe it can be stored elsewhere
  - Ingest Action to S3

Unused Sourcetypes

What does the data look like?

Audit:[timestamp=05-05-2024 00:15:16.205, user=splunk-system-user, action=search, info=completed, search_id='scheduler__nobody_SW5mb1NlY19BcHBfZm9yX1NwbHVuaw__RMD57bc665bd594a1cb5_at_1714868100_31264', has_error_warn=false, fully_completed_search=true, total_run_time=3.05, cpu_time=0.11, event_count=129, result_count=28, available_count=0, scan_count=834, drop_count=0, exec_time=1714868112, …

invocations.command.search.rawdata.bucketcache.error=0, sourcetype_count__linux_secure=4, search_type=scheduled, …

Unused Sourcetypes

How to do it?

```
index=_audit sourcetype=audittrail info=completed action=search
| rex max_match=0 "sourcetype_count__(?<st>\w+)=(?<cnt>\d+)"
| where isnotnull(st)
| append
    [| tstats count where index=* by sourcetype,index | stats values(index) as indexes by sourcetype
    | rename sourcetype as st | eval cnt=0]
| search NOT st IN (audittrail http_event_collector_metrics itsi_internal_log scheduler splunk*)
| stats sum(cnt) as cnt, values(indexes) as indexes by st
| where cnt=0
| fields - cnt
| rename st as sourcetype
```

Unused Sourcetypes

More info

The results of this query are meant to be used to start talks with your users. Find out if they're really not using the sourcetype or if it's one that is needed only under certain circumstances (break glass, audit, etc.). It's important to note that the output of this search improves as the time range is extended.

Please do not feed them into automation that deletes inputs or indexes.

```
index=_audit sourcetype=audittrail info=completed action=search
``` Extract sourcetype names from the audittrail events ```
| rex max_match=0 "sourcetype_count__(?<st>\w+)=(?<cnt>\d+)"
| fields st cnt
``` Discard events without a sourcetype ```
| where isnotnull(st)
| append
    ``` Get a list of all sourcetypes ```
    [| tstats count where index=* by sourcetype,index | stats values(index) as indexes by sourcetype
    | rename sourcetype as st | eval cnt=0]
``` Ignore Splunk default sourcetypes (not an exhaustive list) ```
| search NOT st IN (audittrail http_event_collector_metrics itsi_internal_log scheduler splunk*)
``` Count how many times each sourcetype is used ```
| stats sum(cnt) as cnt, values(indexes) as indexes by st
``` Keep only the sourcetypes with count of zero ```
| where cnt=0
| fields - cnt
| rename st as sourcetype
```

Questions?

Resources

Links to slides, app, etc.

Please complete the survey

Thank you

Appendix

Cool stuff we didn’t have time to talk about

_internal index

Notes about some components

component = Metrics

Caution: Values are based on sampled data so they may be incomplete

component = DC:DeploymentClient

Can help find Deployment Server clients

component = TCPInputProc

Can help find TCP inputs. Watch out for inputs that stop reporting.

sourcetype = splunkd_remote_searches

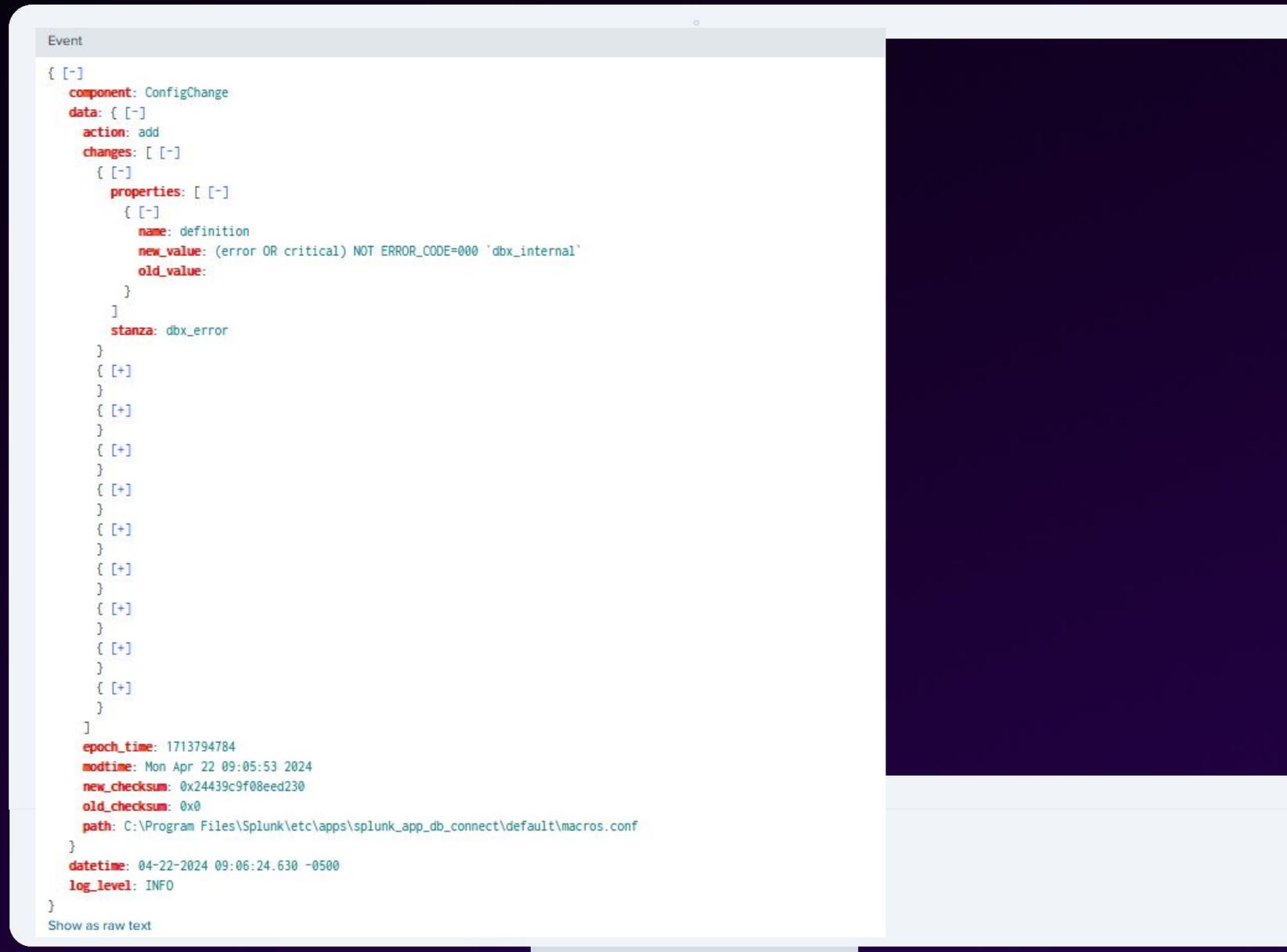
Most stats here apply only to the indexers; don’t forget to add CPU time used by SH

Interesting numbers to watch: index_bucketcache_miss_wait, rawdata_bucketcache_miss_wait

_configtracker index

Track changes to configs

- Reports changes to .conf files made via the UI
- Does not attribute changes to a user
- This example event shows the macro “dbx_error” was created



Find Inactive Users

```
| rest splunk_server=local /servicesNS/-/-/admin/users
| fields title realname last_successful_login
| eval lastLogin = if(isnull(last_successful_login) OR last_successful_login=0, "never", strftime(last_successful_login, "%c"))
| eval idleDays = round((now()-last_successful_login)/86400,0)
| where (idleDays > 90 OR lastLogin = "never")
| table title realname lastLogin idleDays
| rename title as User, realname as Name, lastLogin as "Last Login Time", idleDays as "Days Since Last Login"
```

Objects owned by inactive users

```
| rest splunk_server=local /servicesNS/-/-/admin/directory | fields title eai:acl.app eai:acl.owner eai:type eai:acl.sharing
| rename eai:acl.* as *
| where (owner!="nobody" AND owner!="admin")
| search [| rest splunk_server = local /servicesNS/-/-/admin/users
| fields title realname last_successful_login
| eval lastLogin = if(isnull(last_successful_login) OR last_successful_login=0,"never", strftime(last_successful_login, "%c"))
| eval idleDays = round((now()-last_successful_login)/86400,0)
| where (idleDays > 90 OR lastLogin = "never")
| fields title
| rename title as owner | format]
```

Unused Indexes

Why do we care?

- Don’t pay to store data we don’t use
- Less to manage
- Faster indexer startup

If the data is needed, consider using Ingest Actions to send to S3 instead of an index.

Unused Indexes

How to do it?

Get a list of unused indexes and subtract that from a list of all indexes.

- Extract index names from searches, via audit logs or saved search queries
- Expand macros to see if the contain index names
- Expand tags and eventtypes to see if they contain index names
- If “index=*” is found, fetch the names of all indexes that user can access
- If no index name is found, fetch the user’s default indexes

OR

- Index all search.log files
- Extract index names from ‘INFO IndexScopedSearch [22436 localCollectorThread] - IndexScopedSearch is called for index = history’ events.
- Know this doesn’t work in Splunk Cloud

Unused Indexes

How to do it?

To find out which indexes are not used, you take the list of all indexes and remove those that are used in searches. The second part is the problem. It is not a easy task, which is why we didn't cover it in the main talk. To get a list of used index names, scrape the audit logs, saved searches, and dashboards for queries that use "index=…" or "index IN (…)" construct and extract the index names. While parsing the queries, expand any macros you find in case they contain index references. Do the same for tags and eventtypes. Note that this may be a recursive operation. Any instance of "index=*" should be replaced by all of the indexes the user's role is allowed to access. Similarly, a query with no explicit index name requires retrieving the user's default indexes.

There is an option, covered in a white paper by David Paper, that uses the IndexScopedSearch component in search.log to collect used index names. It requires all search.log files to be indexed ($$) and takes time before enough indexed data is collected to make a useful dataset. This is not available to Splunk Cloud users, however.

Unused Indexes

A possible solution

```
| rest /services/data/indexes splunk_server=local
| search title!="_*" NOT
[ search index=_internal sourcetype=splunkd_remote_searches source=*remote_searches.log "Streamed search connection closed:"
| regex search="^(litsearch|mcatalog|mstats|mlitsearch|litmstats|tstats|presummarize)" ```these lines have search expanded```
| rex field=search max_match=0 "(?s)\|?\s*(mlitsearch)\s+.*?\[(?P<subsearch>.*?)\]\s*(\||$)" ```retrieving subsearch(es)```
| rex field=search "(?s)(?P<prepipe>\s*\|?([^\|]+))" ```retrieving prepipe```
| nomv subsearch ```Converts subsearch from multivalue to single value with values separated by linefeed```
| fillnull value="" subsearch ```eliminate null values of subsearch```
| eval search = prepipe . " " . subsearch ```Concatenates prepipe search with its subsearch(es)```
| rex field=search "(?s)(NOT\s+index(\s*=\s*|::)[^ ]+)|(NOT\s+\([^\)]+\))|(index(\s*=\s*|::)\"?(?P<indexregex1>[\*A-Za-z0-9-_]+))" max_match=50 ```gets value(s) after "index=" or
"index::" into 'indexregex'```
| rex field=search "(?s)(NOT\s+index(\s*IN\s*)[^ ]+)|(NOT\s+\([^\)]+\))|(index(\s*IN\s*\()(?P<indexregex2>[^\)]+))" max_match=50 ```gets indexes specified in 'IN' statement```
| eval indexregex2=trim(split(indexregex2,","),"\s\""), index=mvappend(indexregex1,indexregex2) ```splits comma-delimited indexes from 'IN' statement into mv field and trims
quotes and spaces from name and appends indexes discovered from 'index=' or 'index::'```
| stats values(index) as index ```gets mv list of indexes attempted to search```
| mvexpand index ```places each unique, discovered index pattern on a separate row```
| where !match(index,"^_?\*+$") ```do not include index=* or index=_*```
| where !match(index,"^_\w+") ```do not include internal indexes``` | rename index as title | fields title | format]
| table title
```

Thanks to Tim Pacl of Splunk

More Resources

What Splunk Logs about Itself

Description of Splunk Process Introspection Values

About Splunk Enterprise Platform Instrumentation

Redundant & Inefficient Search Spotter app

Index Usage Reporting - white paper by David Paper

Scheduled Search Management - white paper by David Paper

Even More Resources

Write better searches

Splunk Clara-fication Search Best Practices

Splunk Clara-fication Dashboarding Best Practices

Splunk Clara-fication Job Inspector

TRU1143C - Splunk > Clara-fication: Job Inspector (conf20 session by Martin Müller and Clara Merriman)

Improving dashboard performance and resource usage (15 minutes), Dashboard Studio

TRU1133B - Clara-Fication: More Tstats for Your Buckets (2021), (PDF link), (Video link)

(conf22) Master joining your datasets without using join

How to compare fields over multiple sourcetypes without 'join', 'append' or use of subsearches?

(conf23) I Am Speed! Searching on Your Own TERMs With Simple Techniques That 99% Aren't Using!

(conf22) PLA1466B - Fields, Indexed Tokens and You

Using the Splunk job inspector (youtube)

PLA1089C - TSTATS and PREFIX, How to get the most out of your lexicon with walklex, tstats, indexed fields, PREFIX, TERM

TRU1133B - Clara-Fication: More Tstats for Your Buckets

PLA1162B - Clara-Fication: Finding and Improving Expensive Searches

Disclaimers

Feel free to use the SPL in these slides, but test it before using it in a production environment. Use at your own risk.

The SPL is intended to show what is possible and may not be efficient. Nor is it exhaustive.

The searches that extract information from dashboard code were not tested against SPL2 queries.

Your results may vary.