Multi-Tenant Splunk

Pitfalls from the Pit

Steven Bochniewicz

October 2019 | Version 1.1

splunk>

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. 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. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> listen to your data

Speakers

Steven Bochniewicz
Sr. Security Architect

UMD Splunk Story in 30 Seconds

2013: Started with 100GB/day
- Internal IT Security use

2014: Moved to 200GB/day
- Expanded Security use

2017: Upgraded to 1.7TB/day
- Expanding to campus

2018 and beyond: 3TB+/day
- Expanding to all of campus

Multi-site cluster (2 data centers), per data center:
- 12 x Indexers
- 2 x Search Heads
- Cluster Master
- Deployment Server

Splunk as Service

Centralized managed splunk for
Decentralized IT services
Potential 122 department it customers sharing one splunk enterprise infrastructure

SplunkBase

Friend or Foe?

SplunkBase just like Splunk not designed for Multi-Tenant

Many apps have searches and other lookups that will run as system which will run with access to all indexes

These apps are a great starting point for views and dashboards

- Watch out for lookups and static index definitions

Apps that can provide CIM mapping and other field extractions are a great

Building on the work of others

Daniel Daily, Splunk Administrator, Indiana University

Allen Tucker, Manager, HELPnet Technology Services, Indiana University

http://conf.splunk.com/sessions/2016-sessions.html#search=Daniel%20Daily

Cary Peterborg, Sr. Monitoring Engineer, LDS Church

https://conf.splunk.com/session/2015/conf2015_CPetterborg_LDS_Church_SimplifiedForwarderDeploymentand_FINAL.pdf

What we have done

Automate all the things:

- AWS through CloudFormation/Puppet - Index creation
- Templated search head apps - new departments
- ClientName (deploymentclient.conf) - department specific technical addons

What we have done continued...

Naming Convention:

- All Indexes, Apps, Roles…
- Division_Department_(SubDepartment)_

Not managing role mappings in splunk (too much configure too little time)

Custom apps for each department

- Separate Read and Write Roles
- Separate Index roles
- Index wildcard

Create custom privilege roles for issue access to commands and features

Department Apps

Example Settings

Authorize.conf

```
[role_a_vpaf_ps_r]
### This is an role placeholder for application permissions ###

[role_a_vpaf_ps_rw]
### This is an role placeholder for application permissions ###

[role_i_vpaf_ps]
### This is an role defines index permissions ###
srchIndexesAllowed = vpaf_ps_*
srchIndexesDefault = vpaf_ps_*
```

user-prefs.conf

```
[role_a_vpaf_ps_r]
default_namespace = UMD_vpaf_ps

[role_a_vpaf_ps_rw]
default_namespace = UMD_vpaf_ps
```

default.meta

```
# Application-level permissions

[]
access = read : [ a_vpaf_ps_r, a_vpaf_ps_rw, a_mgmt ], write : [ a_vpaf_ps_rw, a_mgmt ]
export = app
```

default.xml - data/ui/nav

```
<nav search_view="search" color="#65A637">
  <view name="search"/>
  <view name="datasummary" default='true' />
  <view name="reports" />
  <view name="alerts" />
  <view name="dashboards" />
  <collection label="Windows">
    <view name="eventid" />
    <view name="audit_events" />
    <view name="users_and_groups" />
    <view name="interesting_events" />
    <view name="interesting_processes" />
    <view name="windows_event_sources" />
    <view name="windows_events_xml_source" />
  </collection>
</nav>
```

splunk> listen to your data

Dashboards

Use the work of other apps

Include dashboard and views from a Managed App for quick changes to common dashboards
Directly include another view(dashboard) in an apps nav menu
Add import to default.meta to be able to import knowledge objects from other apps

```
[]
access = read : [ * ], write : [ admin ]
export = app
import = anotherapp
```

Lookups

can be your friend or you data breacher

transforms.conf

```
1. [default]
2. check_permission = true
3.
4. [mylookuptable]
5. filename = yourLookupName.csv
6. check_permission = true
7.
```

limits.conf

```
1. [outputlookup]
2. outputlookup_check_permission = true
3.
```

Lookups can be great source to summarize data and provide quick access

However many apps have savedsearches that build lookups will run as system and thus collect more data then potentially intended.

- Create service account or assign these savedsearches to run as a user with the limited access, or modify the base search to only include the restricted indexes.

By default Splunk does not verify permissions before writing to a lookup file, and power users have the write to create and modify lookups.

Scheduled / Saved Searches

Saved searches by default run as Owner
- Configurable per search
- Can be forced to always run as user

Allows you to override a user limitation with search indexes, however if left unmonitored could lead to data loss

Scheduled Searches always run as owner

Remove the ability to schedule searches from the default User Role

Limit rights of users to perform scheduled searches
- Since these searches only run as owner.

savedsearches.conf - [default] or per search stanza

dispatchAs = [user|owner]

* This setting is only meaningful for shared saved searches.

* When dispatched as "user", the search is run as if the requesting user owned

  the search.

* When dispatched as "owner", the search is run as if the owner of the search

  dispatched the search, no matter which user requested it.

* If the 'force_saved_search_dispatch_as_user' setting, in the limits.conf

  file, is set to "true", then the 'dispatchAs' setting is reset to "user" while

  the saved search is dispatching.

Search App

is not your friend but you can’t disable it

Since by default all users need access to search app for Splunk functions/commands this is a common path where users can breach data

Hide the Search app

- If you don’t users might accidently store searches/alerts or dashboards in them

Forwarders

You can’t control what you can’t control

Create templates of common apps and collections

- Windows App
- *NIX App

Limit the amount of ability users configure forwarders

Cannot control what the forwarders will send

- If the users manage their own inputs they can send whatever they want to any index.

Questions?

Steven Bochniewicz
@Stboch Slack User Group