Real-world Role-based Access Control In A Decentralized Environment

Brandon Lattin

Security Engineer, University of Minnesota

Joshua Buysse

Security Engineer, University of Minnesota

.conf2016

splunk>

Disclaimer

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 the 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.

Goals

- Implement least-privilege access to data ingested by Splunk
- Allow data sharing between groups on a need-to-share basis
- Minimize number of LDAP groups that need to be maintained
- Minimize user LDAP group memberships

Challenges

Institution

- Highly decentralized organization with about 1200 IT staff
- Privacy and data access concerns
- Regulatory issues – FERPA, HIPAA, etc.

Data Specific

- User and Power User roles able to search all indexes
- Creation and usage of default indexes

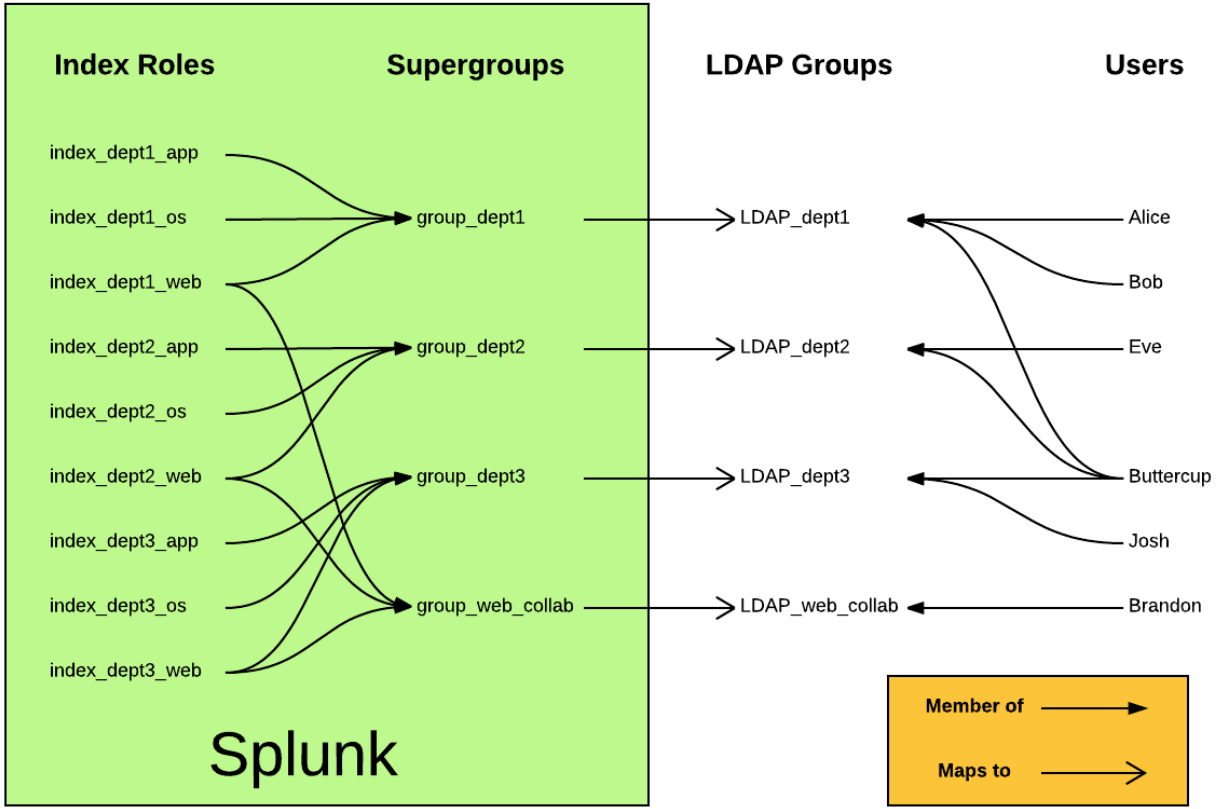
Initial Changes

- Create new roles to replace User and Power User
- Inherit new ”base_user” and “base_power” roles from User and Power User
- Ensure that both of the new roles have these settings:
  - Indexes searched by default is empty
  - Indexes searchable is empty

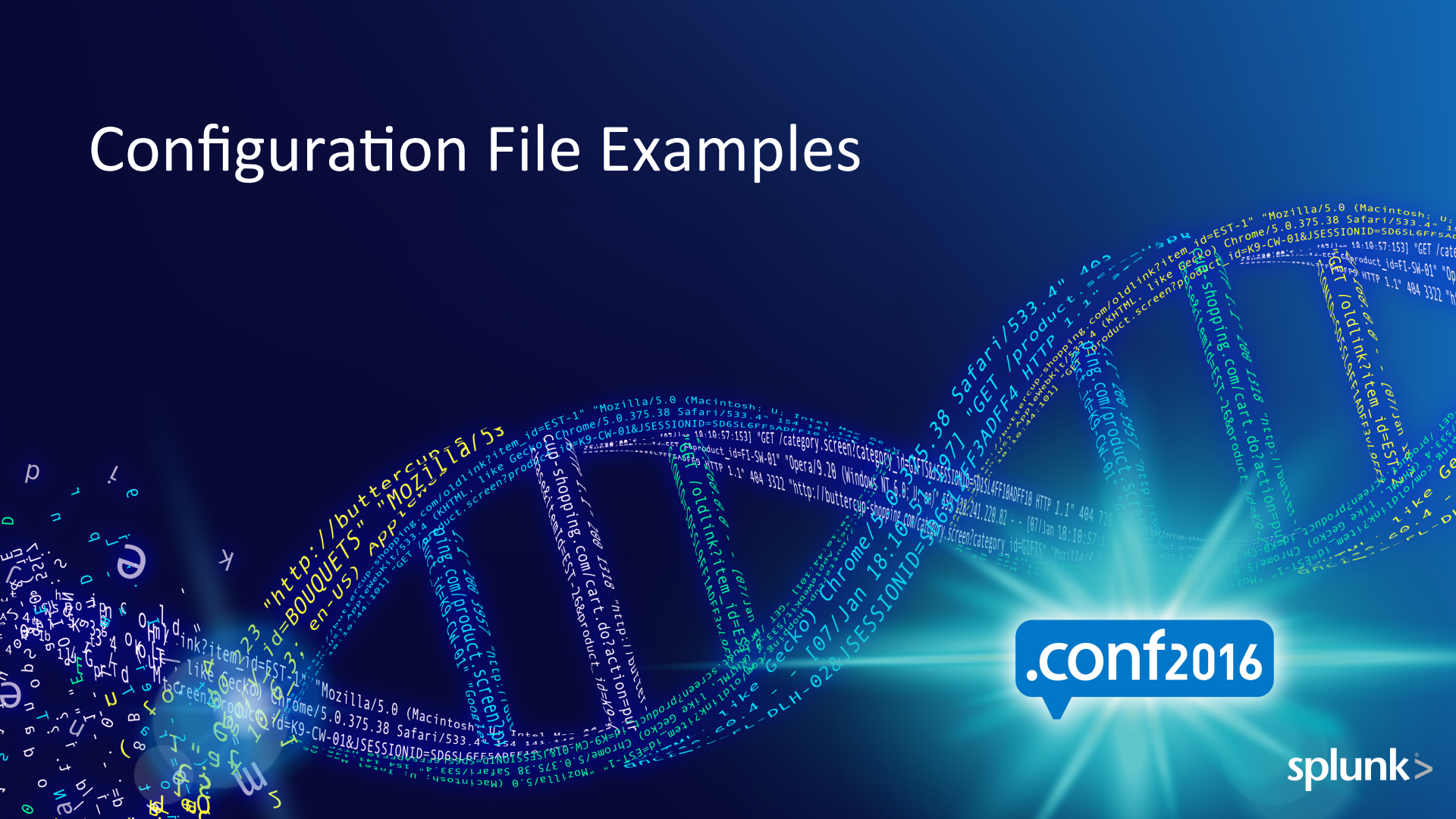
Index Groups and ”Supergroups”

- One group for each index with permission to search that index
- Supergroups contain users
- Each supergroup inherits one or more index groups
  - Index groups can be included by more than one supergroup
- Map supergroups to external directory

Membership and Mapping Relationships



Configuration File Examples



Using Configuration Files

- Make an app
  - Leave $SPLUNK_HOME/etc/system/local alone whenever possible
  - Allows for version control and easy deployment
- authorize.conf
  - Base roles (base_user, base_power)
  - Index roles (index_dept1_web, index_dept1_os)
  - Supergroup roles (group_dept1, group_security)
- authentication.conf
  - Splunk group to LDAP group mapping
  - We will ONLY be mapping supergroups to LDAP groups

authentication.conf

```
# Role LDAP mappings
[roleMap_ldapprd]
# we keep the admin group mapping outside LDAP in case things go sideways
base_user = umn:it:splunk:base_user
base_power = umn:it:splunk:base_power

# Department #1
group_dept1 = umn:it:splunk:group_dept1

# Department #2
group_dept2 = umn:it:splunk:group_dept2

# Cross-functional web collaboration group
group_web_collab = umn:it:splunk:group_web_collab

# Security
group_security = umn:it:splunk:group_security
```

authorize.conf – base roles

```
[role_base_user]
importRoles = user
srchIndexesAllowed =
srchIndexesDefault =

[role_base_power]
importRoles = power
srchIndexesAllowed =
srchIndexesDefault =
srchDiskQuota = 2000
srchJobsQuota = 32
srchMaxTime = 0
...
```

authorize.conf – index roles

```
...
[role_index_dept1_app]
srchIndexesAllowed = dept1_app
srchIndexesDefault = dept1_app

[role_index_dept1_os]
srchIndexesAllowed = dept1_os
srchIndexesDefault = dept1_os

[role_index_dept1_web]
srchIndexesAllowed = dept1_web
srchIndexesDefault = dept1_web

# use with care!
[role_index_all_indexes]
srchIndexesAllowed = _*;*
...
```

authorize.conf – supergroup roles

```
...

[role_group_dept1]
importRoles = index_dept1_app;index_dept1_os;index_dept1_web

[role_group_dept2]
importRoles = index_dept2_app;index_dept2_os;index_dept2_web

[role_group_web_collab]
importRoles = index_dept1_web;index_dept2_web;index_dept3_web

[role_group_security]
importRoles = index_all_indexes
...
```

Wrapping Up

Tips

- Define a default app for each supergroup (user-prefs.conf)
- Keep **admin** role mappings in $SPLUNK_HOME/etc/system/local/ to avoid losing administrator mappings if your role app breaks!
- Require explicit index selection for groups that access all indexes
  - No indexes searched by default

Gotchas

- Default app permissions will almost always need modification
- Some apps and TAs expect to put data in a specific index, or will create indexes for you
- Avoid default indexes, like ‘main’
- If you create an app to manage roles, be aware of configuration file precedence
- Don’t delete or modify user and power - this will break things

What Now?

Related breakout sessions and activities…

THANK YOU

splunk>