Partitioning Shared Resources for Access Between Multiple Agencies

Myron Davis

Programmer, State of Alaska

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.

Personal Introduction

- Myron Davis, State of Alaska
- Member of ETS (Enterprise Technology Services) - provide support for the separate operating units in the state
- Logging and analysis for the State of Alaska
- One of the managers of the Splunk system

Background

State of Alaska has 15 major operating units running through multiple shared firewalls and intrusion detection/prevention systems.

Our problem... share our common infrastructure in order to provide additional feeds of information to business units.

Agenda

- **Goal:** Provide access to business units for shared hardware such as IDS/firewall and other event logs tailored to their specific unit
- Pre-process data with syslog-ng
- Deploy package for synchronizing syslog-ng tagging with Splunk tagging
- Configuring rolemap with authentication.conf and authorize.conf to link search filters and tags applied by syslog-ng to LDAP groups

Syslog-ng Config

splunk>

Syslog-ng Configuration

- A default Linux kernel is not built for the behavior of a dedicated high usage syslog machine, and changes to sysctl.conf are required to get decent throughput
- Syslog-ng is multi-threaded
  - Need to prep your syslog-ng.conf file
- Regular expressions are CPU intensive
  - Default configuration does not take full advantage of multiple CPUs.

Syslog-ng Configuration System Prep

- **Increase buffers**... because syslog-ng has issues processing bursts of records with complex regex's quickly

- Make following change to your sysctl.conf file:
  - Add lines to your /etc/sysctl.conf

net.core.rmem_max=1073741824

net.ipv4.udp_rmem_min = 16384

net.ipv4.tcp_rmem=32768 2097152 134217728

Syslog-ng Configuration

- **Verify data.** Are you getting all your syslog data?
- netstat -su
- Look for the line “packet receive errors”: should be zero
- Logs are lost (sometimes on UDP)
  - Compare the UDP Packets received line to the packet receive errors and the RcvBufErrors. Make sure those numbers are low...
  - If not low, diagnose your buffers: CPU usage and tweaks may be required
  - Check **ALL** syslog servers. Are you losing data?

Syslog-ng Configuration: syslog-ng.conf

- Set some decent defaults:

```
options { chain_hostnames(off); flush_lines(500);
use_dns(no); use_fqdn(no);
log_fifo_size(536870912);
owner("root"); group("adm"); perm(0640);
stats_freq(0);
bad_hostname("^gconfd$"); threaded(yes);
log_msg_size(8192);
};
```

Syslog-ng: Multiple Sources

Every Major Systems need to come into separate buffers to be processed.
We have 16 total inputs across multiple machines for load leveling regex processing.

- Syslog-ng (as of the version we are using), regex processing is SINGLE thread per pipe.
- If you want to use the multi-threading capabilities with UDP you MUST multi-pipe the inputs.
- TCP (both ssl and non-ssl) does NOT have this problem, but does have more overhead.

```
#inside firewall
source s_ext_udp_15140 {
        udp(so_rcvbuf(268435456)
log_fetch_limit(10000) port(15140));
};
#outside firewall
source s_ext_udp_15141 {
        udp(so_rcvbuf(268435456)
log_fetch_limit(10000) port(15141));
};
```

Create a Processing Pipe

```
log {
        source(s_ext_udp_15140); #choose one source data
        filter(f_pix); #verify this pipe is PIX data
        filter(f_noisy_asa_events); #filter any specific events you DON'T want
        rewrite(r_add_dot); #add all of your rewrite statements
        rewrite(r_add_doa);
        rewrite(r_add_dol);
# add location TAGS
rewrite(r_add_anc);
rewrite(r_add_jnu);
rewrite(r_add_fai);
        destination(pixhosts); #send to a directory structure with cisco gear
        destination(d_ciscolog); #send to a temporary log file
};
```

Sample Filter to Find Matching Networks

```
filter f_sam {
#Sample range IPS

#Use http://www.analyticsmarket.com/freetools/ipregex for assistance in generating a start address

#This filter finds all SAMPLE networks

# matching 10.even.64.y – 10.even.95.y that don't match 230-255 in the second octet
message(".*10\\.(\\d*[02468]|[02468])(?<!2[3-5][0-9])\\.(6[4-9]|[7-8][0-9]|9[0-5])\\.([0-9]|[1-9][0-9]|
1([0-9][0-9])|2([0-4][0-9]|5[0-5])).*" type(pcre))

#You can add additional subnets if you wish!
or message(".*10\\.247\\.(3[2-9]|[4-5][0-9]|6[0-3])(?<!2[3-5][0-9]>)\\.([0-9]|[1-9][0-9]|1([0-9][0-9])|2([0-4]
[0-9]|5[0-5])).*" type(pcre));
};
```

If a Sample Dept Has LOTS of Small Subnets

```
#Be Smart, don't duplicate regexs.. for example!
filter f_sam {
message(".*146\\.63\\.[0-9].*" type(pcre));
and
(
#Split the search into TWO major ranges 1's and 2's, if those don't match then hit the rest
message(".*146\\.63\\.1.*" type(pcre))
and
(
#insert all of the search for individual subnets under the 1* octet here
)
or message(“.*146\\63\\2.* type(pcre)
and
(
#insert all of the small individual subnets under the 2* octet here
)
)
};
```

To Add Dept Identifiers

```
# add department identifiers

rewrite r_add_doa { set("$MSGONLY ,DOA=1", value("MSGONLY") condition(filter(f_doa))); };

rewrite r_add_dol { set("$MSGONLY ,DOL=1", value("MSGONLY") condition(filter(f_dol))); };

rewrite r_add_dot { set("$MSGONLY ,DOT=1", value("MSGONLY") condition(filter(f_dot))); };

rewrite r_add_laa { set("$MSGONLY ,LAA=1", value("MSGONLY") condition(filter(f_laa))); };
```

To Add Internal Location Identifiers...

```
# add location identifiers

rewrite r_add_anc { set("$MSGONLY ,GEOLOC=ANC", value("MSGONLY") condition(filter(f_geoanc))); };

rewrite r_add_jnu { set("$MSGONLY ,GEOLOC=JNU", value("MSGONLY") condition(filter(f_geojnu))); };

rewrite r_add_fai { set("$MSGONLY ,GEOLOC=FAI", value("MSGONLY") condition(filter(f_geofai))); };
```

Where Are We At Now?

We have data which is tagged with ownership information. Sometimes individual log entries can be tagged by **multiple agencies.**

For example (fake log file):

2015-08-28T14:24:13-08:00 10.231.8.4 : %ASA-6-302014: Teardown TCP connection 2963008268 for inside-dc:10.247.90.41/53093 to inside-dmz1e:10.4.9.3/80 duration 0:00:00 bytes 522 TCP FINs ,DNR=1 ,DOA=1, HSS=1, ETS=1

Owned by 4 separate entities:

source is Department of Natural Resources (10.247.90.41) user,
network device (10.231.8.4) is owned by DOA, AND ETS,
destination web server is owned by Health and Social services.

All agencies involved have a stake in this individual log entry.

Splunk TA Apply Tags

splunk>

Splunk TA to Apply Tags props.conf

I applied tags at index time using a TA called TA-dept

[default]

REPORT-doa = dept_doa

REPORT-dol = dept_dol

REPORT-dot = dept_dot

Splunk TA to Apply Tags transforms.conf

I applied tags at index time using a TA called TA-dept

```
[dept_doa]
REGEX = DOA\=(1)
FORMAT = DOA::$1

[dept_dol]
REGEX = DOL\=(1)
FORMAT = DOL::$1

[dept_dot]
REGEX = DOT\=(1)
FORMAT = DOT::$1
```

Splunk TA to Apply Tags Meta

This is probably not needed, but here is the default.meta for this TA that we just went over.

```
[]

access = read : [ * ], write : [ admin, power ]

[eventtypes]
    export = system

[props]
    export = system

[transforms]
    export = system

[viewstates]
    access = read : [ * ], write : [ * ]
    export = system

[lookups]
    export = system
```

Summary

By creating a TA with [default] it will allow these rules to be tagged to ALL source types, not just being applied to data types that have key-value pairs loaded.

This TA which has just been outlined is a very simple TA, but required, and must be customized to your environment.

Splunk ACL
authentication.conf
authorize.conf

splunk>

Authentication.conf Base Example Setup

```
[authentication]
authSettings = DOMAIN
authType = LDAP
[DOMAIN]
SSLEnabled = 0
anonymous_referrals = 0
bindDN = CN=splunkserviceaccount,OU=Service
Accounts,OU=SUBAGENCY,OU=AGENCY,OU=State
Departments,DC=DOMAIN,DC=EXAMPLE,DC=COM
bindDNpassword = Tief8ieHOsei9thiEeroaR0fiem6OhBa
charset = utf8
groupBaseDN = OU=Splunk Groups,OU=SUBAGENCY Groups,OU=State Groups,OU=State
Departments,DC=DOMAIN,DC=EXAMPLE,DC=COM
```

Authentication.conf Cont'd.

```
groupMappingAttribute = dn
groupMemberAttribute = member
groupNameAttribute = cn
host = directorycontroller.DOMAIN.EXAMPLE.COM
nestedGroups = 0
network_timeout = 20
port = 389
realNameAttribute = displayname
sizelimit = 1000
timelimit = 15
userBaseDN = DC=DOMAIN,DC=EXAMPLE,DC=COM
userNameAttribute = samaccountname
```

Authentication.conf Sample Rolemap

[roleMap_DOMAIN]

Admin = Splunk Admins
ced = CED Splunk
cdr = CDR Splunk
dnr = DNR Spunk
doa = DOA Splunk
doc = DOC Splunk
dol = DOL Splunk
dot = DOT Splunk
dps = DPS Splunk
doahttpryinbound = DOA Splunk
etshttpryinbound = ETS Splunk
dolhttpryinbound = DOL Splunk
dothttpryinbound = DOT Splunk
eedhttpryinbound = EED Splunk
cedhttpryinbound = CED Splunk
hsshttpryinbound = HSS Splunk
cedtagged = CED Splunk
dectagged = DEC Splunk
dfgtagged = DFG Splunk
mvatagged = DMVA Splunk
dnrtagged = DNR Spunk
doatagged = DOA Splunk

Stanza in authorize.conf on left
LDAP group on right

Authorize.conf Sample Stanza

```
[role_doa]
cumulativeRTSrchJobsQuota = 2
cumulativeSrchJobsQuota = 8
importRoles = power
rtSrchJobsQuota = 12
srchDiskQuota = 1000
srchIndexesAllowed = doa
srchIndexesDefault = doa
srchJobsQuota = 6
srchMaxTime = 0

[role_doatagged]
cumulativeRTSrchJobsQuota = 0
cumulativeSrchJobsQuota = 0
importRoles = power
srchFilter = DOA=1
srchIndexesAllowed = fw;snort;syslog;cisco_acs
srchIndexesDefault = fw;snort;syslog;cisco_acs
srchMaxTime = 0

[role_doahttpryinbound]
cumulativeRTSrchJobsQuota = 0
cumulativeSrchJobsQuota = 0
importRoles = power
srchFilter = DOA=1 AND (dest_as=3724 OR dest_as=0) AND src_as!=0 AND src_as!=3724
srchIndexesAllowed = httpry
srchIndexesDefault = httpry
srchMaxTime = 0
```

Top Stanza a “normal” role mapping 100% access to all logs in one index

2nd Stanza only allow access to data with key-value pair of DOA=1

3rd Stanza only allow access to incoming web logs from sniffer tagged DOA=1

Sample Log httpry

Flash player version 18.0.0.232 accessing an external document. This would NOT be allowed by the previous rule as the previous rule only allowed incoming web logs. (ASN rules are also in place)
This is an example of a PIPE delimited log format being merged with a targeted key-value search.

[2015-09-04T10:00:19-08:00 10.230.8.34 httpry |10.0.225.141|64410|204.2.145.163|80|POST|fnurtmp-f.akamaihd.net|/control/FNCPREV_1_300@143121?cmd=throttle,82&v=3.6.0.50&r=VPSJF&g=ZJMXWSOQNPLH&lvl1=11.571,11,17.432,15.26,0,3.295,1638,0,1,300,35908.361,1441389343.969,35902.527,141.081,135.247,35902.527,5959,0.8,1.872,0,0,145466,u,false|http://foxnewsplayer-a.akamaihd.net/player/7.25.0.0000/amp.foxnews/AkamaiPremierPlayer.swf|Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)|200|14|text/plain|-|AkamaiGHost|18,0,0,232|-|-|-|0.12|| ,DPS=1 ,GEOLOC=ANC

Questions?

THANK YOU

splunk>