To HEC with syslog!

Scalable Aggregated Data Collection in Splunk

Mark Bonsack, CISSP | Staff Sales Engineer
Ryan Faircloth | PS Security Consultant
September 28, 2017 | Washington, DC
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, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
Who are we?

▶ Mark: Staff Systems Engineer, Southwest Majors
  6 years @ Splunk
  Focus: Data Onboarding, Security, IT Operations

▶ Ryan: Senior Security Consultant
  3 years @ Splunk
  Focus: Security, Data Onboarding, Search Performance
We Will Discuss:

1. Syslog and Splunk Best Practices
2. Traditional Syslog/UF Architecture
3. New! HEC with Syslog
4. Python HEC Interface to Syslog
5. Wrap-up/Resources
Syslog and Splunk: Best Practices

Section subtitle goes here
What can Splunk Ingest?
Agent-Less and Forwarder Approach for Flexibility and Optimization

Aggregated/API Data Sources
- Pre-filtering, API subscriptions
  - Heavy Forwarder

Event Logs, Active Directory, OS Stats
  - *Unix, Linux and Windows hosts
    - Universal Forwarder

Wire Data
  - Splunk Stream
    - Universal Forwarder or HTTP Event Collector

Local File Monitoring
- Universal Forwarder
  - *nix
  - Windows
  - Mainframes

DevOps, IoT, Containers
- HTTP Event Collector (Agentless)
  - Syslog server

syslog hosts and network devices

Shell API

Aggregation host

Syslog server
If You Take Only One Thing From This Session...

Do *not* send syslog traffic (on any port) directly to Splunk indexers

(Except in the smallest of installations. Or other corner cases. There are always corner cases.)

TCP/UDP 514
Here’s Why...

▶ Even data distribution on indexers required for search performance at scale
  • Sending "514" traffic to just one indexer works in only the smallest of deployments
  • UDP load balancing typically trickier than TCP

▶ Syslog is a *protocol* – not a sourcetype
  • Syslog typically carries multiple sourcetypes
  • Sourcetypes are essential for “Schema on the Fly”

▶ Best Practice: pre-filter syslog traffic using syslog-ng or rsyslog
  • Provides for a separate *sourcetype* for each technology in the syslog stream of events
  • Use a UF (good) or HEC (best!) back end for proper sourcetyping and data distribution

▶ The rest of this session will show you how to do that!
Ramifications of doing it wrong
Improper sourcetyping

- Can’t find my events when everything is just syslog; no fields to help
  - Yes we can search by IP but we have to look only by key words (“uber-grep”).
  - No "Schema on the Fly" – the key to 99% of the power of Splunk!
Ramifications of doing it **wrong**

Uneven data distribution

- Each indexer takes a turn processing all events for a given block of time, it's just like having 1 indexer
  - $\mid\text{tstats count where index=pan_logs by span=1s \_time splunk_server} | \text{timechart sum(count) as count by splunk_server useother=false}$
Solution: Use a UF or HEC to transport data to Splunk
Benefits of doing it right

- Indexers share even load for all time spans
And at scale...

- Even better distribution (real customer data; 1 TB/day ingest)
Syslog-ng or rsyslog?
Which syslog server to choose?

**syslog-ng**
- Very rich filtering syntax
- High familiarity
- Open Source or fully supported from Balabit
  - Becoming less prevalent on recent Linux distros

**rsyslog**
- Default on almost all Linux distros
- Somewhat difficult filtering syntax
  - Though getting better
- Some distros (Red Hat) may use old versions unsupported by the upstream

**Both Equally at Home with Splunk!**
Traditional UF Architecture

Time-tested performance
Syslog/UF Architecture

Traditional Approach

- Time-tested
- Scales – to a point.
- Complicated Architecture at Scale
- Two configuration tasks
  - Configuration of Syslog server and UF
- So – Let’s dig in!
Syslog-ng Config File Structure

You will see variations on this theme

- Global Options
- Log Sources
- Log Destinations
- Log Filters
- Log Declarations (Source, Dest, Filter)
Syslog-ng Configuration

Global Options and Sources

# Global Options
options {
    sync (40);
    time_reopen (10);
    time_reap(5);
    long_hostnames (off);
    use_dns (no);
}

# Log Sources
source s_syslog {
    udp(ip(0.0.0.0) port(514));
    tcp(ip(0.0.0.0) port(514));
};
Syslog-ng Configuration
Destinations, Filters, and Log Directives

# Destinations
destination d_checkpoint { file("/var/splunk/syslog-${LOGHOST}/chpt/${HOST}.log" create_dirs(yes));};
destination d_asa { file("/var/splunk/syslog-${LOGHOST}/asa/${HOST}.log" create_dirs(yes));};
destination d_all { file("/var/splunk/syslog-${LOGHOST}/data/all.log" create_dirs(yes));};

# Filters for Sourcetypes
filter f_checkpoint { host("10\.64\.8\.79") and match("kernel" value("PROGRAM"));};
filter f_asa { match("%ASA" value("MESSAGE"));};

# Log directives
log { source(s_syslog); filter(f_checkpoint); destination(d_checkpoint); };;
log { source(s_syslog); filter(f_asa); destination(d_asa); };;
Rsyslog Config File Structure

You will see variations on this theme too!

- Global Options
- Log Sources
- Log Filters
- Log Actions
- Log Outputs
rsyslog Configuration

Global Options and Sources

#load modules only once
module(load="imudp")
module(load="imptcp")
module(load="omprog")

#Accept both tcp and udp; some sources use both
input(type="imudp" port="514" ruleset="splunk_file")
input(type="imptcp" port="514" ruleset="splunk_file")
rsyslog Configuration

Destinations, Filters, and Log Directives

#Filters and Actions for Splunk UF

```
ruleset(name="splunk_file") {
    if $msg contains '\%ASA\' then {
        action(type="omfile"
       File="/var/splunk/syslog-%myhostname%/asa/%hostname%.log")
    } if fromhost-ip == "10.64.8.79" then {
        action(type="omfile"
       File="/var/splunk/syslog-%myhostname%/checkpoint/%hostname%.log")
    }
}
```
UF inputs.conf Configuration

Uses structure created by syslog filtering

[monitor://var/splunk/syslog-*/asa/*.log]
  disabled = 0
  index = network_firewall
  host_regex=\/var\/splunk\/syslog[^\/]*/[^\/]*/[^\/]*/([^\.]*)
  sourcetype = cisco:asa

[monitor://var/splunk/syslog-*/chpt/*.log]
  disabled=0
  index=network_firewall
  host_regex=\/var\/splunk\/syslog[^\/]*/[^\/]*/([^\.]*)
  sourcetype = chpt:next_gen
New! HEC with Syslog

Scalable and Simple!
What Drove the Need?
This is where the subtitle goes

- Data distribution
- Search performance
- Ease of Configuration
- OPEX cost reduction
Syslog/HEC Architecture

A New Approach to Scale

- Scales significantly beyond standard UF Architectures
- Allows use of standard TCP load balancers in data path
- Simpler to configure and administer at scale
- Utilizes most of syslog config from UF-based architecture
What causes the indexer imbalance

- Each source (file) is assigned to a pipeline
- Each pipeline will (based on time) rotate to the next indexer at random
  - Most customers choose (default) 30s
- Therefore each pipeline may only load 2 indexers per minute or 10 over 5 min.
- The problem becomes more pronounced as the rate of events from a source increases and the number of indexers increase
Proper load balancing makes search faster!

- The goal is to minimize the separation of the lines in the graph below
  - All indexers receive an equal distribution of data
- Solution: Balance the indexer by events – not time or size
Check your own environment

How even is your indexed data?

| tstats count where index=* sourcetype=<largest syslog type by volume> by span=10s _time splunk_server
| timechart sum(count) as eps by splunk_server
To HEC with Syslog!
Prepare the indexers for HEC

Enable HTTP Event Collection via inputs.conf on the indexer

```
[http]
disabled=0
port=8088

[http://syslog]
disabled=0
index=main
token=<yourguidhere>
indexes=main,summary
```

Set Up the Load balancer

- Select least connected round robin
- Reuse existing SSL Sessions
syslog-ng Configuration for HEC

Simple change for HEC (Raw endpoint; batch via external script)

```bash
# Raw endpoint, batch mode via “omsplunkhec.py” script.
# Arguments to omsplunkhec.py: token, HEC host, options, payload
# Payload can use full complement of syslog-ng templates and macros
# Note: GUID required by raw endpoint is supplied by omsplunkhec.py

destination d_http3
{} program("/usr/local/bin/omsplunkhec.py 00000000-0000-0000-0000-000000000000
hec_endpoint --sourcetype=syslog_tcp --index=main"
template("original_host=${HOST} <${PRI}>${DATE} ${HOST} ${MSG}\n") ;");
```

`syslog-ng` configuration for HEC.
rsyslog Configuration for HEC

Simple change for HEC (Raw endpoint; batch via external script)

# Raw endpoint, batch mode via "omsplunkhec.py" script.
# Arguments to omsplunkhec.py: token, HEC host, options, payload

ruleset(name="splunk_file") {
    if $msg contains '\%ASA\' then {
        action(type="omprog" binary="/usr/local/rsyslog/bin/omsplunkhec.py DAA61EE1-F8B2-4DB1-9159-6D7AA5220B21 192.168.100.70 --sourcetype=cisco:asa --index=netfw" template="RSYSLOG_TraditionalFileFormat")
    }

    if fromhost-ip == "10.64.8.79" then {
        action(type="omprog" binary="/usr/local/rsyslog/bin/omsplunkhec.py DAA61EE1-F8B2-4DB1-9159-6D7AA5220B21 192.168.100.70 --sourcetype=chpt:next_gen --index=netfw" template="RSYSLOG_TraditionalFileFormat")
    }
}
What does all this look like in Splunk?
Using the previous syslog-ng configuration examples

▶ ...and the same event (other than the timestamp):

\[
\text{<165>1 2017-03-19T23:44:38+00:00 sender.computer.org evententry - ID47 [example iut="3" eventSource="Application" eventID="1011"] Test message}
\]

▶ Looks like this using the `d_http3` syslog-ng destination ("raw" HEC endpoint):

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/19/17</td>
<td>original_host=sender.computer.org &lt;165&gt;Mar 19 23:44:38 sender.computer.org Test message</td>
</tr>
<tr>
<td>4:44:38.000 PM</td>
<td>host = dda38bac0b93 ; original_host = sender.computer.org ; source = hec:syslog.dda38bac0b93 ; sourcetype = syslog_tcp</td>
</tr>
</tbody>
</table>
Python HEC Interface to Syslog

omsplunkhec.py
Yes, a simple Script

It's just that easy!

- Read input from stdin
- Assign event to a connection in pool
- Bundle events into transactions
- Post the events

Where to get it:

https://bitbucket.org/rfaircloth-splunk/rsyslog-omsplunk
Never write data to disk

Keep the process simple
- avoid any processing that could be done in the syslog server or Splunk
- Read one event from the syslog server per line from stdin

Bundle events together in raw mode
- allows effective use of each session “batch size”
- allow tuning if needed

Keep data moving
- use a thread pool allowing the load balancer to manage which indexer needs messages next
- thread pool prevents the time required for session management from impacting latency
Arguments to omsplunkhec.py
Supplied when calling script from syslog server

token: http event collector (HEC) token (required)
server: http event collector (HEC) IP/fqdn (required)
--port: port: (default='8088')
--ssl: use ssl: (action='store_true', default=False)
--ssl_noverify: disable ssl validation: (action='store_false')
--source: Splunk metadata: (default="hec:syslog:" + host)
--sourcetype: Splunk metadata: (default="syslog")
--index: Splunk metadata: (default="main")
--host: Splunk metadata: (default=syslog_host)
--maxBatch: max number of records allowed in one batch of requests for hec: (default=10, type=int)
--maxQueue: max number of records to be read from rsyslog queued for transfer: (default=5000, type=int)
--maxThreads: max number of threads for work: (default=10, type=int)
1. Do not send “514” syslog traffic directly to forwarders or indexers!

2. Use a syslog server with UF or HEC for data fidelity, performance and scale

3. There are many helpful resources, both Splunk and open source
Helpful Resources

This session is fully documented here:

• https://www.rfaircloth.com/2016/05/16/building-high-performance-low-latency-rsyslog-splunk/
• http://www.rfaircloth.com/2017/02/10/building-perfect-syslog-collection-infrastructure/

Additional Resources

• https://bitbucket.org/rfaircloth-splunk/rsyslog-omsplunk (omsplunkhec.py source)
• https://www.splunk.com/blog/2016/05/05/high-performance-syslogging-for-splunk-using-syslog-ng-part-2.html (good overview of syslog-ng server configuration and optimization)
• http://www.rsyslog.com/rsyslog-configuration-builder/ (rsyslog configuration tool (beta))
• http://www.rsyslog.com/doc/v8-stable/ (rsyslog documentation)
Thank You

Don't forget to rate this session in the .conf2017 mobile app

splunk> .conf2017