https://fontmeme.com/straight-outta/

Take Control of Port 514!: Taming the Syslog Beast

Mark Bonsack
Staff Sales Engineer | Splunk

Ryan Faircloth
Splunk Enterprise Architect | Splunk

splunk> .conf19

.conf19

splunk>

Mark Bonsack

Staff Sales Engineer | Splunk

Ryan Faircloth

Splunk Enterprise Architect | Splunk

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. 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, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Agenda

Taming the Syslog Beast with SC4S!

1. History and Problem
2. Syslog Basics
3. Splunk Connect for Syslog
4. SC4S Architecture
5. SC4S Configuration Overview
6. A Look Ahead
7. Resources

History and Problem

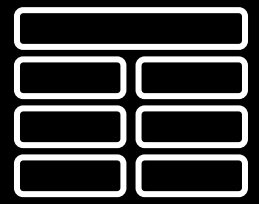
Syslog: Splunk's first data source

.conf19
splunk>

What is the Challenge with syslog?

RFC 3164
RFC 5424

Syslog is a *protocol* –not a sourcetype



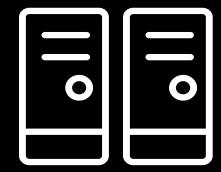
Multiple formats ride on those protocols

Syslog presents unique scale challenges

What have we Historically Done?

Send syslog data directly to Splunk

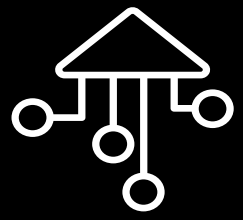
Require App/TA authors to sort it all out



Kept an arm's distance in supporting syslog servers

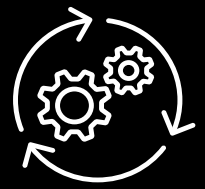
What is the Effect on Customers?

Poor Out-of-Box Experience

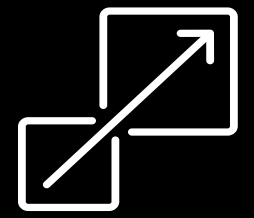


Send directly to Splunk “sourcetype=syslog”

>



Schema on the Fly difficult to apply



Excess Splunk resource usage

Direct syslog ingest does not scale

Syslog Basics

A primer

.conf19

splunk>

Syslog Is:

A Standard for System Logging

- A fundamental part of *nix from the earliest days; 30+ years old!
- An overloaded term and can be defined as:
  - A host-based facility to log local system events
  - A wire protocol for transmission of events between devices and systems
    - Originated with UDP; later expanded to include TCP and TLS
  - All of the data formats used by individual device vendors
- Wire protocol designed to minimize overhead on the sender (device)
  - Favors performance over reliability
  - This fundamental choice means any instability or resource constraint will cause data to be lost in transmission
  - Many devices default to UDP data transmission for this reason

The syslog Wire Protocol

Well-defined by RFCs

- Syslog wire protocol encapsulated in two RFCs
  - RFC 3164 (AKA “BSD syslog”)
    - https://tools.ietf.org/html/rfc3164
    - Ratified in 2001
  - RFC 5424 (AKA “protocol-23”)
    - https://tools.ietf.org/html/rfc5424
    - Ratified in 2009

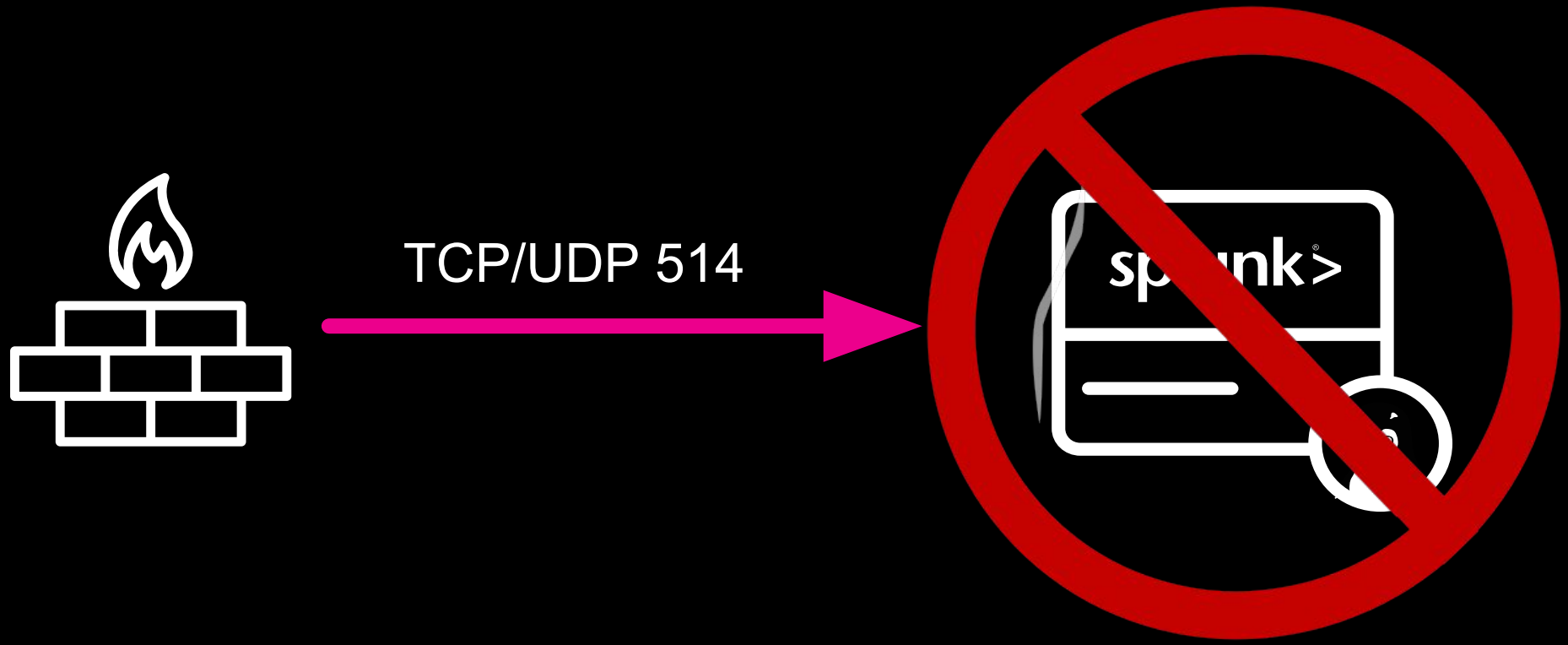
Basics of syslog Data Collection

You *Are* Using a syslog Server, Aren't You?

- Two major Syslog server software packages are in wide use:
  - Syslog-ng, Owned by One Identity
    - https://www.syslog-ng.com/
    - Offers syslog as a hardware appliance
  - rsyslog
    - https://www.rsyslog.com/
- Both very high-quality software that has been developed over *decades*
- Both have fully functional Open Source versions
- Syslog-ng viewed by many as being easier to configure
- Syslog-ng serves as the core of Splunk Connect for Syslog

If You Take Only *One* Thing From This Session...

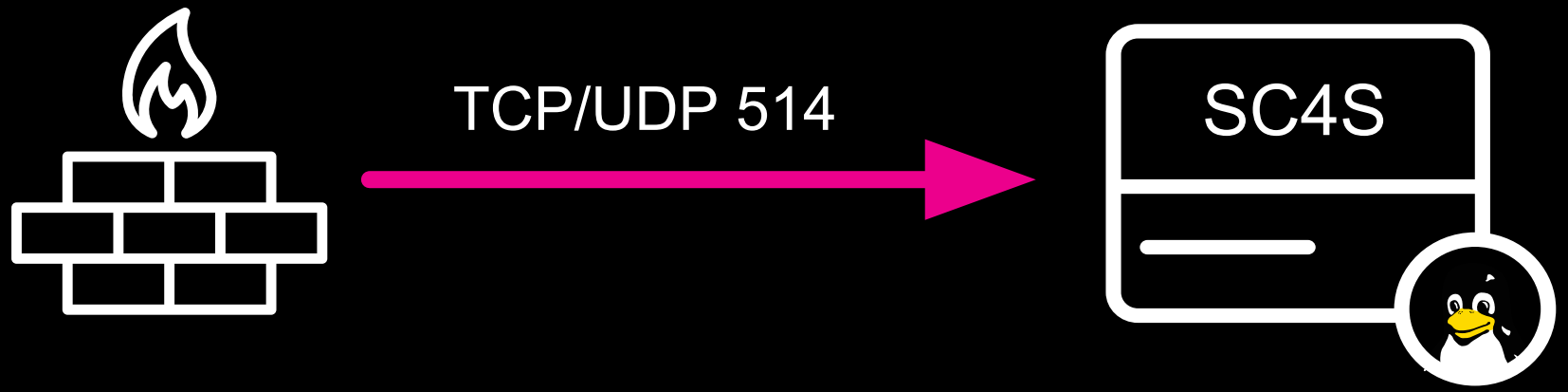
Do *not* send syslog traffic (on any port) directly to Splunk indexers



If You Take Only *One* Thing From *This* Session...

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

But you *can* send syslog traffic (on any port) to Splunk Connect for Syslog!



Splunk Connect for Syslog

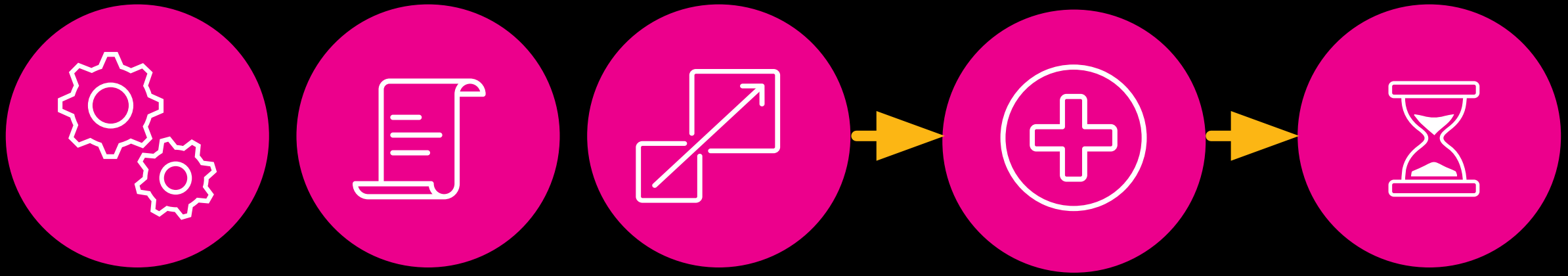
A turnkey solution!

.conf19
splunk>

How do I easily ingest syslog data, at scale, while removing the requirement of up-front design work and syslog-fu?

Introducing Splunk Connect for Syslog!

A Solution for Splunk's Oldest Data Source



Turnkey Container

Consistent Repeatable

Scalable

Data Hygiene Efficient Ops

Time to Value Customer Sat

Introducing Splunk Connect for Syslog!

From this:

```
<165>1 2019-09-13T15:23:34.700Z talent-habitat RT_IDP - IDP_ATTACK_LOG_EVENT
[junos@2636.1.1.1.2.135 epoch-time="1507845354" message-type="SIG"
source-address="183.78.180.27" source-port="45610"
destination-address="118.127.xx.xx" destination-port="80" protocol-name="TCP"
service-name="SERVICE_IDP" application-name="HTTP" rule-name="9"
rulebase-name="IPS" policy-name="Recommended" export-id="15229" repeat-count="0"
action="DROP" threat-severity="HIGH" attack-name="TROJAN:ZMEU-BOT-SCAN"
nat-source-address="0.0.0.0" nat-source-port="0"
nat-destination-address="172.xx.xx.xx" nat-destination-port="0" elapsed-time="0"
inbound-bytes="0" outbound-bytes="0" inbound-packets="0" outbound-packets="0"
source-zone-name="sec-zone-name-internet" source-interface-name="reth0.XXX"
destination-zone-name="dst-sec-zone1-outside"
destination-interface-name="reth1.xxx" packet-log-id="0" alert="no"
username="N/A" roles="N/A" message="-"]
```

Introducing Splunk Connect for Syslog!

To this. No more “sourcetype=syslog”!

i	Time	Event
>	9/13/19 3:31:11.700 PM	RT_IDP: IDP_ATTACK_LOG_EVENT [junos@2636.1.1.1.2.135 epoch-time="1507845354" message-type="SIG" source-address="183.78.180.27" source-port="45610" destination-address="118.127.xx.xx" destination-port="80" protocol-name="TCP" service-name="SERVICE_IDP" application-name="HTTP" rule-name="9" rulebase-name="IPS" policy-name="Recommended" export-id="15229" repeat-count="0" action="DROP" threat-severity="HIGH" attack-name="TROJAN:ZMEU-BOT-SCAN" nat-source-address="0.0.0.0" nat-source-port="0" nat-destination-address="172.xx.xx.xx" nat-destination-port="0" elapsed-time="0" inbound-bytes="0" outbound-bytes="0" inbound-packets="0" outbound-packets="0" source-zone-name="sec-zone-name-internet" source-interface-name="reth0.XXX" destination-zone-name="dst-sec-zone1-outside" destination-interface-name="reth1.xxx" packet-log-id="0" alert="no" username="N/A" roles="N/A" message="-"][meta sequenceId="3"] host = talent-habitat source = sc4s sourcetype = juniper:junos:idp:structured

Introducing Splunk Connect for Syslog!

Or even this – All turnkey!

```
>   9/13/19
    3:28:56.700 PM
{ [-]
  HOST_FROM: 192.168.128.5
  MESSAGE:
  MSGID: IDP_ATTACK_LOG_EVENT
  PROGRAM: RT_IDP
  SOURCE: s_default-ports
  _SDATA: { [-]
    junos@2636.1.1.1.2.135: { [-]
      action: DROP
      alert: no
      application-name: HTTP
      attack-name: TROJAN:ZMEU-BOT-SCAN
      destination-address: 118.127.xx.xx
      destination-interface-name: reth1.xxx
      destination-port: 80
      destination-zone-name: dst-sec-zone1-outside
      elapsed-time: 0
      epoch-time: 1507845354
      export-id: 15229
```

Goals of Splunk Connect for Syslog

Taming the Syslog Beast!

Lower the burden, both on customers and Splunkers, of getting syslog data into the Splunk platform

Provide a consistent, documented, and repeatable syslog collection infrastructure

Provide turnkey data ingestion for 18 top sourcetypes (v1)

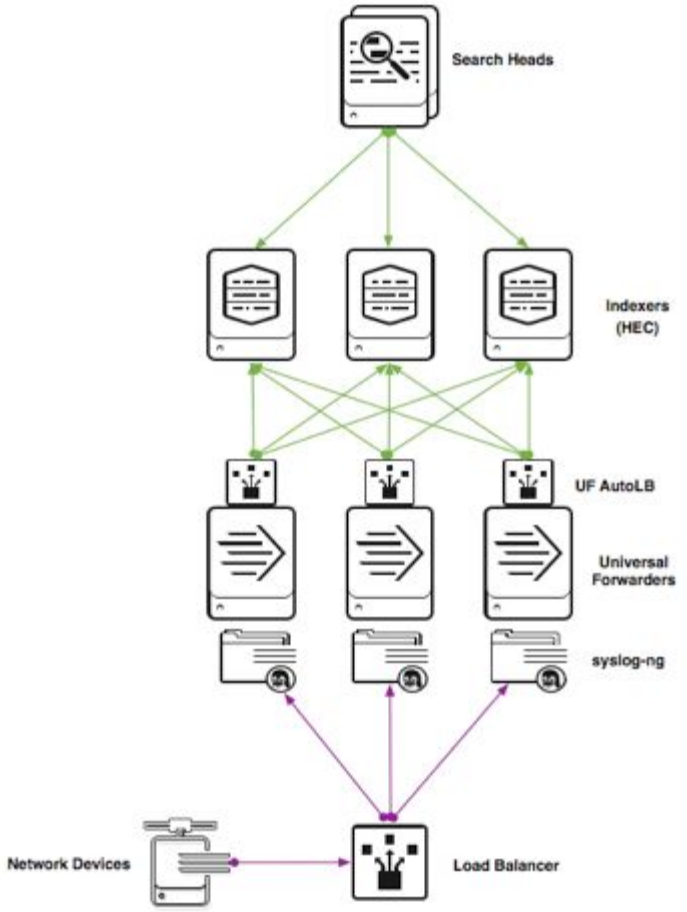
Improve the “data hygiene” of incoming syslog data with proper sourcetyping and enriched metadata; reduces Splunk overhead

Significantly enhance scale and data distribution

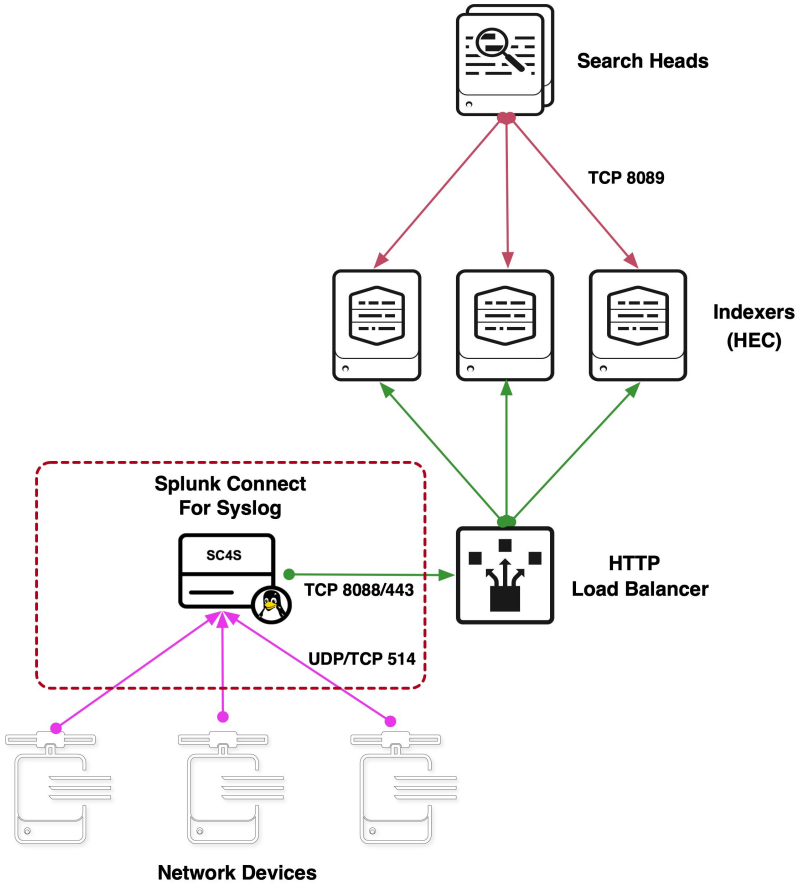
Splunk Connect for Syslog

Turnkey, Performant, and Scalable syslog Data Ingest

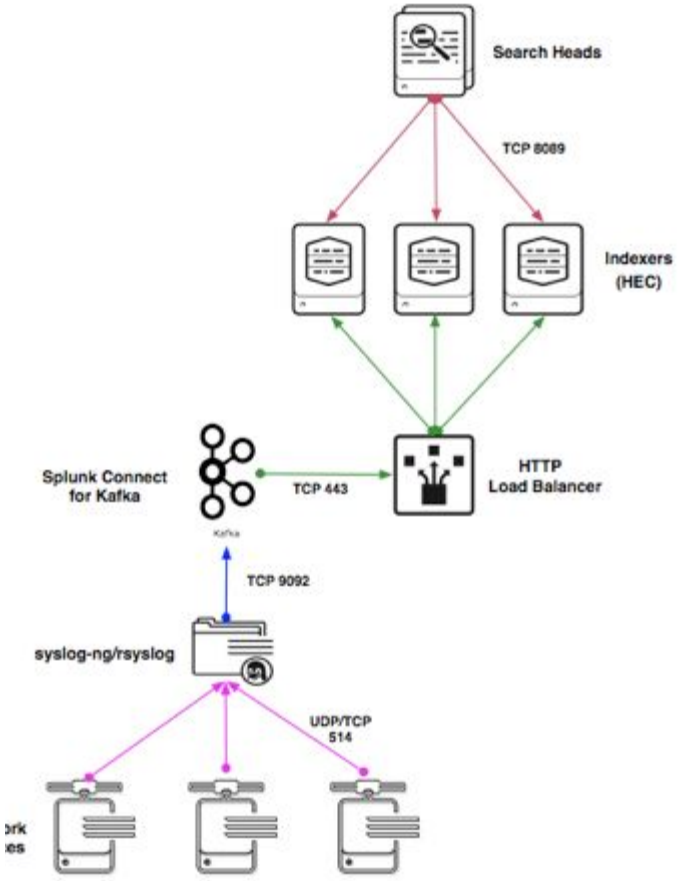
UF



HEC (SC4S v1)



Kafka/HEC (SC4S future)

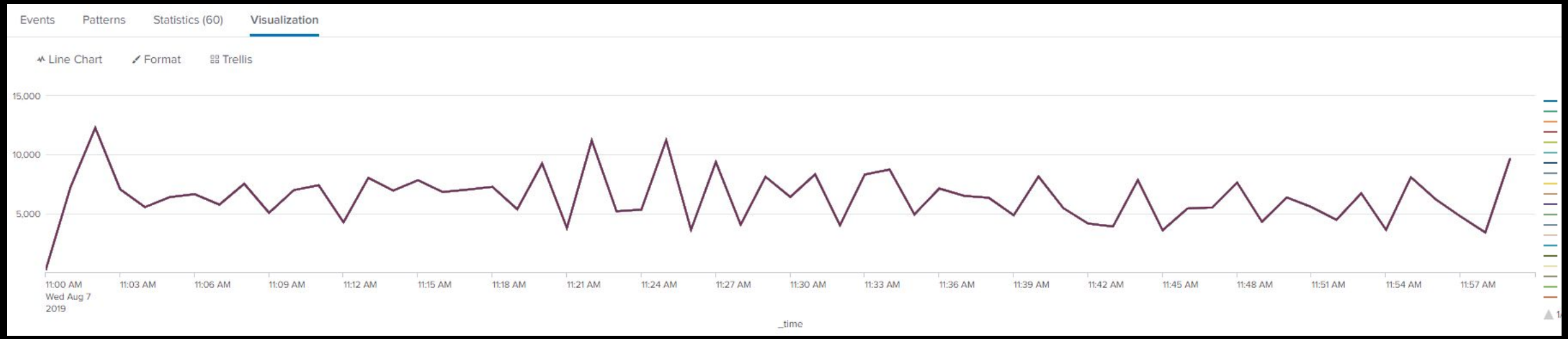


splunk> .conf19

SC4S Data Distribution

A Challenge with Traditional UF-based syslog Designs

- Even data distribution with single-second granularity
- Production customer data; 25 indexers



SC4S Scale

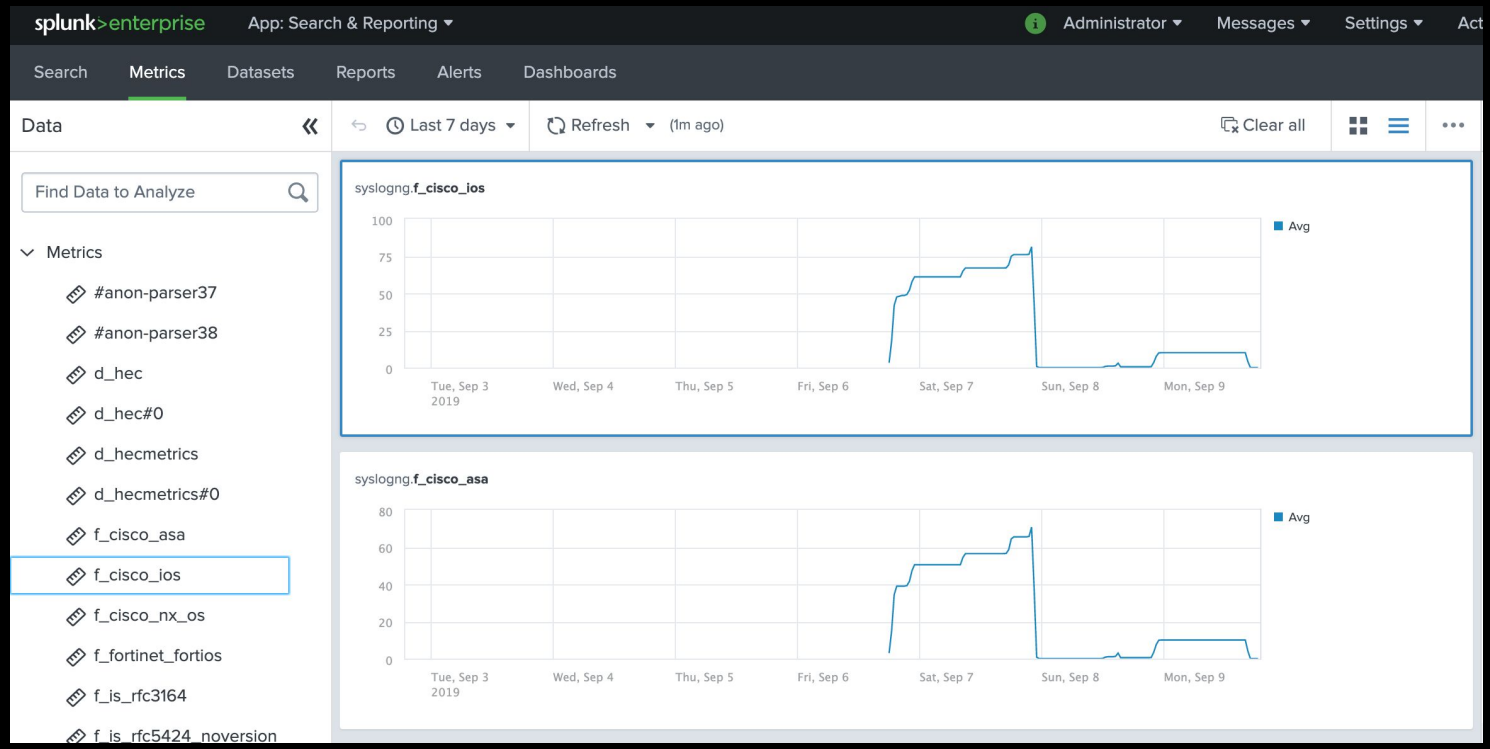
Distributed HEC Provides Unprecedented Scale

- Tested Configuration:
  - SC4S instance requesting 16 cores and 32 GB of memory with K8S scheduler
  - AWS instance type: c5d.4xlarge; NVMe disk buffer
  - Syslog events sent to a 5-indexer single-site cluster
  - Lab conditions with no search
- Command:
  - `loggen -i --rate=5000 --interval=600 -P -F --sdata="[test name=\"stress51\"]" -s 800 --active-connections=40 sc4s.smg.aws  514`
- Results:
  - `average rate = 97267.81 msg/sec, count=58612338, time=602.587, (average) msg size=800, bandwidth=75990.48 kB/sec`
- The single syslog-ng container in this test is able to provide effective balancing and routing of events equivalent to 6.3 TB per day!

SC4S Metrics

Easily Monitor the Health and Wellness of SC4S

- 30s frequency
- Track all destinations, filters, and parsers



SC4S Support

SC4S Will be Released as Open Source

- Full source code on Github
- Splunk will maintain a channel on the splunk-usergroups Slack group
- The SC4S team operates a CI/CD development methodology, ensuring timely updates
- Partner community already contributing to filter development

SC4S Architecture

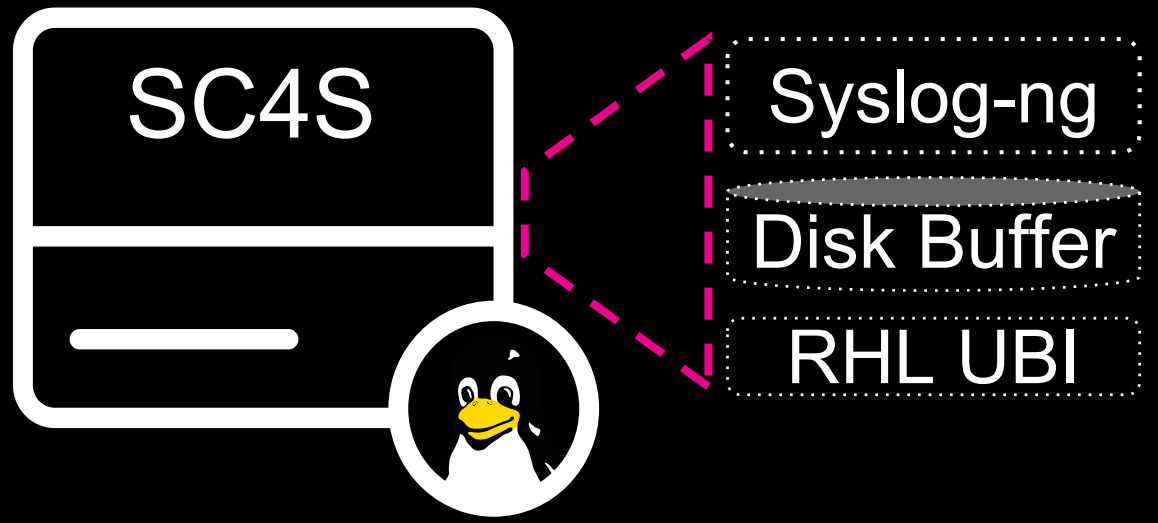
No syslog-ng or rsyslog knowledge needed!

.conf19
splunk>

SC4S Architecture:

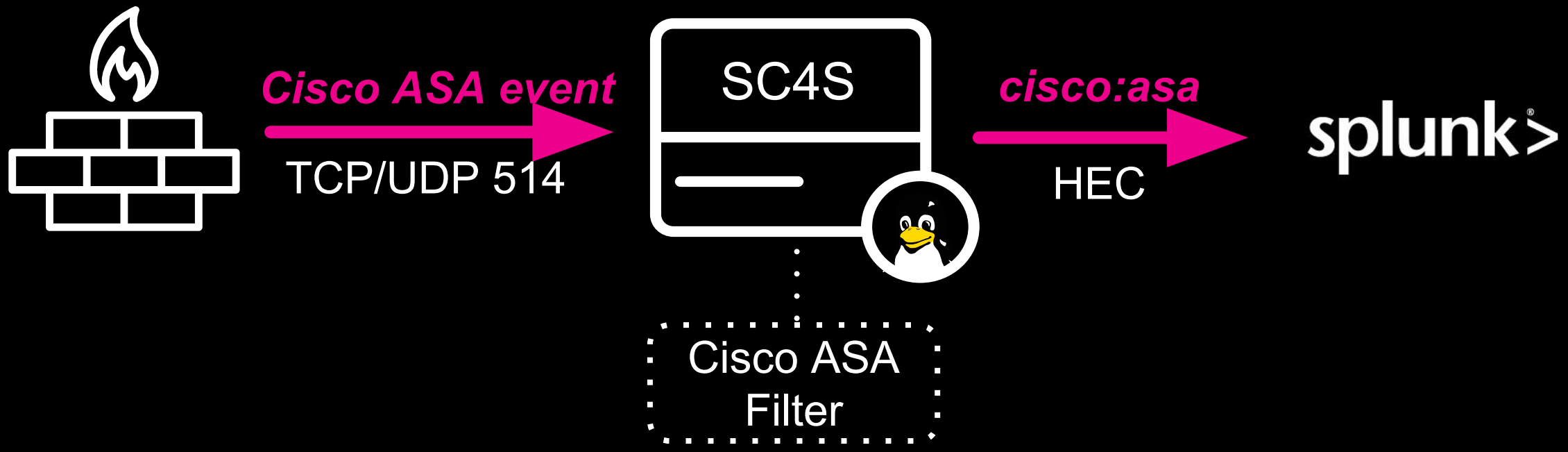
Containers Provide Flexibility and Platform Indpendence

- All syslog-ng configuration and process encapsulated in a container
- Flexible transport choices
  - HEC in v1
  - Kafka/DSP to follow



SC4S Filter Magic

Filter the syslog “soup” with sourcetype auto-identification



Identify > Parse > Format

Design Choices and Constraints

Goal is to Solve the “90%” Problem

- Syslog is a Religion!
  - Syslog is often *way* overengineered
  - SC4S will not solve 100% of the problem, for 100% of the use cases
  - We provide full configs for those who want to adapt to unique circumstances
- Primary goal is to satisfy those who send all of their syslog to the default port (“514 soup”)
- *And* those who need simple customizations such as unique ports and hostname/CIDR blocks
- Solution must require little to no syslog-ng configuration experience
- Solution must be easy to deploy in restricted environments
- Turnkey solution: Container Architecture

Why Did We Choose a Container Approach?

Provides a Turnkey Solution that is Easy to Obtain and Deploy

- Addresses the “Syslog comes with the distro” objection
  - “Cannot use syslog-ng because rsyslog comes with RedHat”
  - Even if syslog-ng comes with the distro, it is *ages* old
  - “Can’t download unapproved software”
  - “Config-only” solution depends on latest version of syslog-ng, which is a prohibitive ask in many environments
- Container allows anyone, regardless of distro, to run SC4S with your container runtime of choice:
  - Docker Compose/Swarm
  - Podman
  - Kubernetes*

Some SC4S Development Specifics

Develop New Source Filters with Confidence!

- Based on RedHat UBI minimal docker based images
- All filter enhancements are fully regression tested before acceptance
- All images are built with a fully automated process
- Documented “IDE” experience for local development of new filters

What the SC4S Container Does *Not* Address

...that the SC4S syslog-ng Config Files (BYOE) on git *Can*

- Custom data formatting
  - Syslog-ng configuration is itself a programming exercise
  - Syslog-ng config syntax is *very* rich and allows for significant processing on its own
  - SC4S limits processing to syslog protocol decode and data preparation for Splunk
    - Index, source, sourcetype, host, and relevant metadata such as priority/severity
- Highly customized existing syslog-ng installations, such as those with (potentially improperly configured) relays
- Sending of data to destinations other than Splunk (or Kafka post-v1)

SC4S Configuration Overview

Not a tutorial

.conf19
splunk>

SC4S Out-of-the-Box Configuration

SC4S: Turnkey for Most Customers

- SC4S ships with pre-defined “filters” for leading security devices
  - SC4S properly sourcetypes data from these devices which send to port 514
- Just a few items are needed from the admin to get going:
  - HEC URL (either a list of endpoints or load balancer VIP)
  - HEC Token
  - Default Data collection port (typically 514)
  - Number of HEC endpoints (needed to properly configure syslog-ng for scale)
  - Disk Buffer Size
- Set as environment variables in the container configuration

SC4S Out-of-the-Box Configuration

There are a Few “Problem Children”

- Certain device types have no defining characteristics. These devices:
  - Are unable to be uniquely identified if mixed in with other traffic (e.g. 514 “soup”)
  - Require an alternative mechanism to describe them
    - Unique receiving port (other than 514)
    - Hostname wildcard(s)
    - Unique receiving CIDR network block

SC4S Environment Customization

SC4S Can be Configured to Match Enterprise Environment

- Two types of customization
- Pre-Instantiation:
  - Requires orchestration/scripting to alter underlying syslog-ng config prior to startup
  - Used to configure:
    - Unique Collection Ports
    - TLS Configuration
- Syslog-ng Configuration File Customization (Runtime):
  - Small snippet of syslog-ng configuration exposed to the admin
  - Used to configure:
    - Hostname wildcards
    - CIDR network blocks

SC4S “Bring Your Own Environment”

For those Requiring Full syslog-ng Config File Access

- Provides the ultimate in flexibility, while retaining benefits of SC4S
- Useful for those with existing, complex syslog-ng environments
- Requires a Linux server and latest (v. 3.22) syslog-ng installation
- Requires significant knowledge of syslog-ng configuration syntax
- Fully documented on GitHub

A Look Ahead

Integration with DSP/Kafka

.conf19

splunk>

SC4S – A Look Ahead

SC4S Will be Continuously Developed

- Additional filters built and vetted by SC4S developers
- “Add your Own” sources
- Additional Destinations to support Splunk future data collection methods
  - Kafka
  - DSP
- Refined configuration/input validation

Resources for SC4S

Links, Blogs, Slack channels...

.conf19
splunk>

SC4S – Resources

SC4S Has a Vibrant Community!

- Main Repository
  - https://github.com/splunk/splunk-connect-for-syslog
- Blog:
  - https://www.splunk.com/blog/search.html?query=SC4S
- Slack Channel:
  - splunk-usergroups.slack.com #splunk-connect-for-syslog

splunk>

Thank You!

Go to the .conf19 mobile app to

RATE THIS SESSION