Traffic Visualization with Arista sFlow and Splunk

Preface

The need for real time traffic information is becoming a growing requirement within a majority of data centers today. Source and destination information, top talkers, top web sites, packet discards, security checks, and packet latency information are being requested, on demand, from network, application and security administrators. The changes in traffic patterns, the sheer volume of traffic and the exponential increase in the data rates are making it difficult to capture, analyze, and troubleshoot traffic, as well as to provide meaningful reports to the application and business constituents.

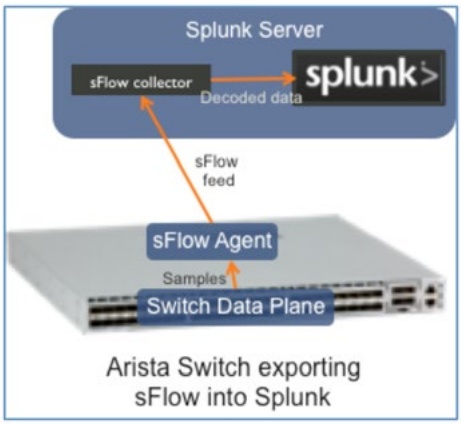
Arista EOS provides ways to gain visibility into network traffic, mirror or steer, and supports triggers/actions based on events but provides no historical reporting/searching or graphical visualization. The extensibility of EOS, with off-the-shelf software from Splunk, provides this. By leveraging Splunk within their network operations, administrators can have real time traffic visibility, reporting and visualization from Arista switching platforms. This is offered without the need for costly hardware switch upgrades or specialized traffic analysis modules. Moreover, this highlights the benefits of an open and extensible platform on Arista switches.

This Arista Solution Brief discusses how to deploy Splunk Enterprise and teach it how to parse sFlow from Arista switches.

Solution Overview

Splunk Enterprise (http://www.splunk.com/) is software that processes machine-generated data in real-time and provides real-time visibility, insight and intelligence via powerful search capabilities.

Splunk ingests machine-generated data from data center infrastructure platforms including server, networking, and storage devices. To date many Splunk use cases have been server application and storage centric or simply based on control-plane log (e.g. syslog) output. This solution brief demonstrates the use of Splunk for capturing distributed traffic data from Arista switches via sFlow and with simple search routines, shows how to generate real time reports on traffic behaviors.

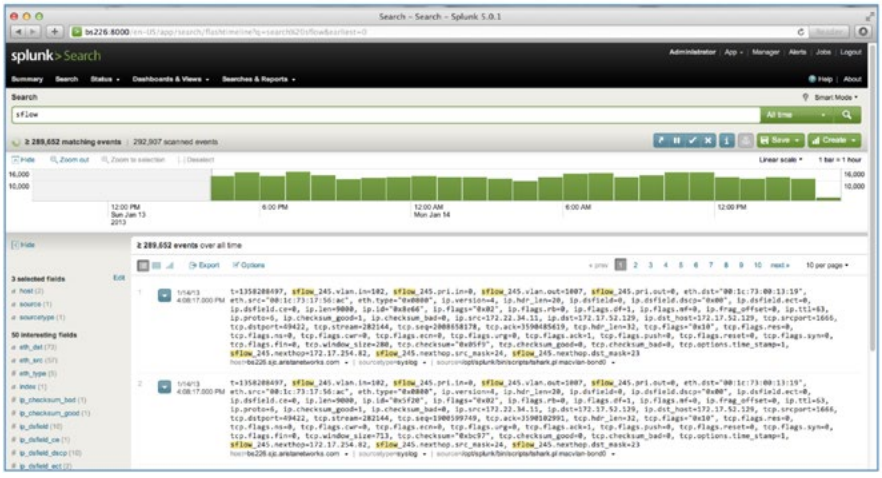


This solution involves taking samples of data-plane traffic via standards-based sFlow (RFC 3176), parsing/extracting L2-L7 fields out of the sampled packets and ingesting that into Splunk software running centralized in the environment which then enables Splunk's extensive reporting and visualization tools to provide visibility into traffic in real-time.

Splunk Output, Search Queries and Reports

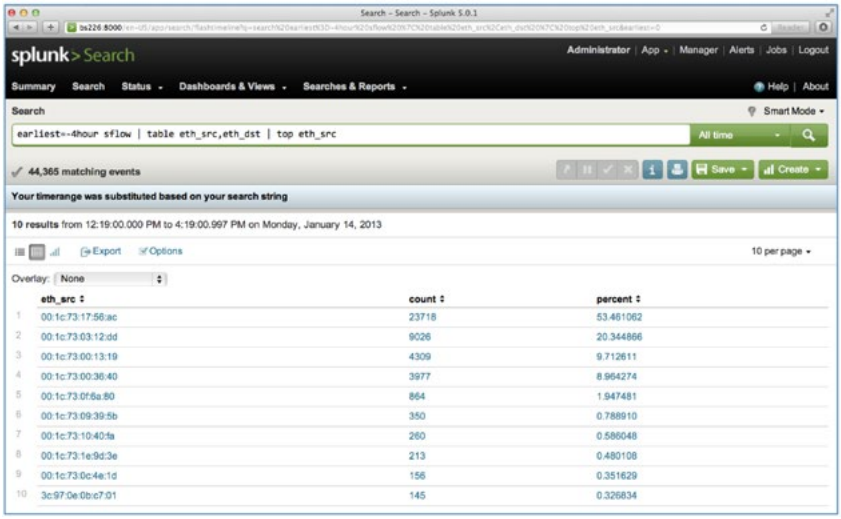
Ingesting decoded sFlow sampled data into Splunk enables search queries based on frame/packet fields. Since sFlow provides the first 128 bytes of each sampled packet, there is not just L2-L4 fields but also the start of packet payloads and things like HTTP, SIP and other protocol headers ready for searching.

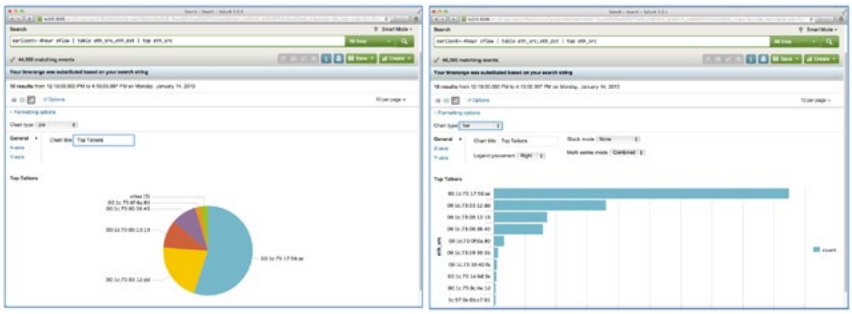
The display below from Splunk's web interface shows frames/packets streaming into Splunk with it waiting for search terms based on field headers:



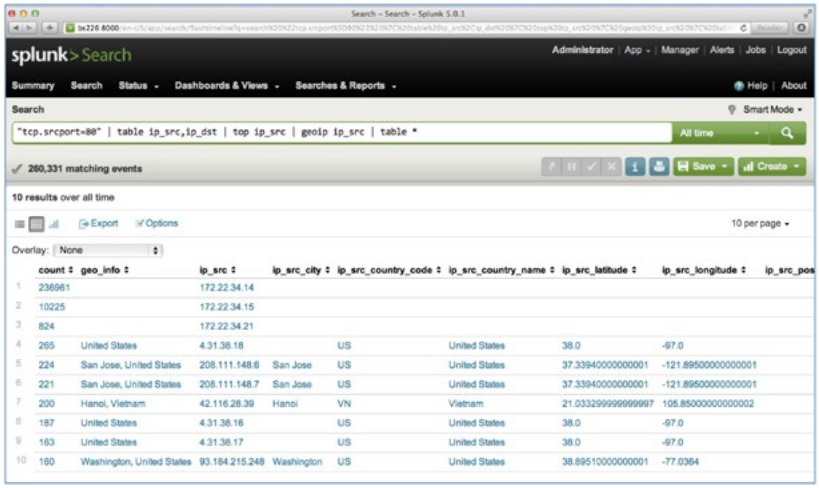
Splunk supports advanced ad-hoc queries of real-time data. For example, if there is more traffic within a VLAN than what is projected this could be analyzed by

constructing a search query for 'Top Talkers' in a given VLAN at layer 2 (eth_src, eth_dst fields) in the last 4 hours. Such a query could be "earliest=-4hour sflow | table eth_src,eth_dst | top eth_src" with the results displayed as a table or graph as shown below:

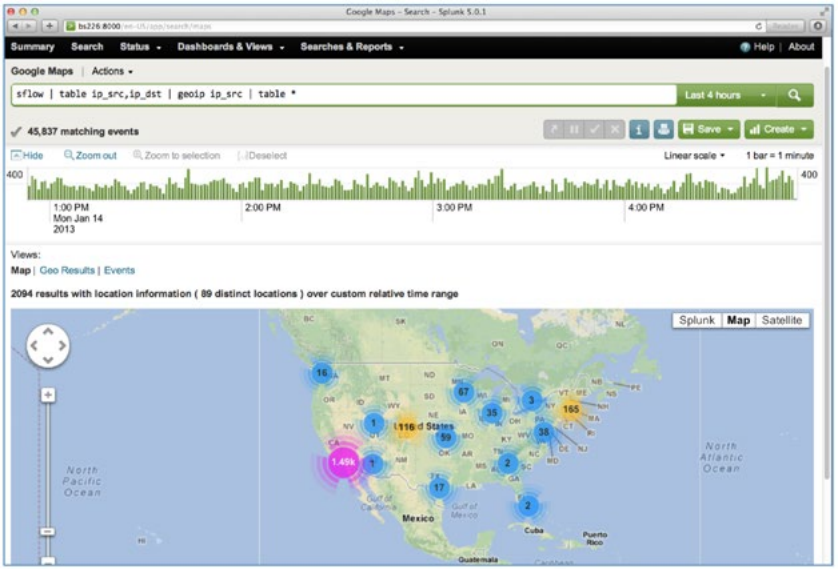




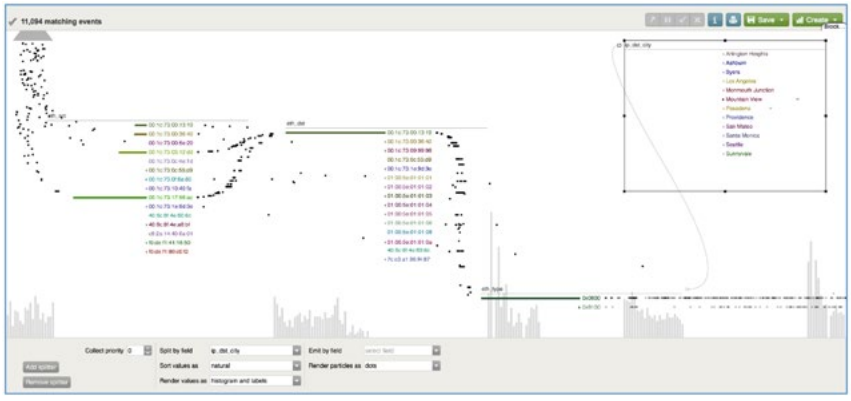
Splunk can post-process things like IP address fields and augment it with additional information such as perform a geo-location lookup. For example, to perform analysis on what countries you have most HTTP traffic originating from using a query in Splunk such as "tcp.srcport=80 | table ip_src,ip_dst | top ip_src | geoip ip_src | table *":



Things like geolocation data can be overlayed on to a map. For example, to show where traffic is originating from a search query such as "sflow | table ip_src,ip_dst | geoip ip_src | table *" with the 'view' set to Google Maps produces results like below:



Splunk Particle provides additional advanced real-time visualization and tools for looking at sFlow sampled traffic and this can be used for real time data forensics. The video at http://www.youtube.com/watch?v=-q-Ue603vKc shows this in action, below is a screenshot of Splunk Particle showing live traffic streaming in, grouped by eth_src initially then eth_dst, eth_type and finally ip_dst_city for any packets that resolve to a geolocation:



There is almost no limit to the possible queries and use cases:

- Application performance and SLA monitoring
- Both packet and flow based traffic analysis, recording and monitoring
- Real time and historical Analytics
- Standard API access to packet capture data
- Access from anywhere

Configuration Recipe: Switch setup

Enable sFlow on the switch with traffic sourced from whatever interfaces and a destination that points towards the server running Splunk. The example below shows sFlow enabled sampling 1 in 20,000 packets with Splunk running on the host 172.16.44.82:

```
switch# config terminal
switch(config)# sflow sample 20000 s
witch(config)# sflow polling-interval 3600
witch(config)# sflow destination 172.16.44.82
switch(config)# sflow source 172.16.44.1
switch(config)# sflow run
```

Configuration Recipe: Splunk server setup

Install Splunk Enterprise software on a server (if not already installed). If this is a fresh Splunk install just download the RPM and install it using RPM:

```
[root@splunk-server ~]# rpm –i splunk-5.0.1-143156.
i386.rpm
-------------------------------------------------
Splunk has been installed in:
      /opt/splunk
To start Splunk, run the command:
      /opt/splunk/bin/splunk start
To use the Splunk Web interface, point your browser
at: http://splunk-server:8000
Complete documentation is at http://docs.splunk.com/
Documentation/Splunk
-------------------------------------------------
[root@splunk-server ~]# /opt/splunk/bin/splunk start
```

Splunk itself doesn't understand sFlow packet sampling data, so a script needs to be installed which can take sFlow traffic, decode it into a format that Splunk can parse and index and use that as a data-source.

We have a sample script on EOS Central that provides this functionality. It takes sFlow packet-sample data, decodes it using Wireshark and presents it in a text format suitable for being ingested by Splunk. Install this script onto the Splunk server with:

```
[root@splunk-server ~]# sudo wget 'https://eos.
aristanetworks.com/wiki/index.php?action=raw&title=
Configuration:Monitoring:arista_sflow_decoder' -O /
opt/splunk/bin/scripts/arista_sflow
Resolving eos.aristanetworks.com... 50.19.101.37
Connecting to eos.aristanetworks.
com|50.19.101.37|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
Length: 1760 (1.7K) [text/x-wiki]
Saving to: `/opt/splunk/bin/scripts/arista_sflow'
```

As this script makes use of the open-source Wireshark network protocol for the packet decoding. Install that on the Splunk server. If the server runs Redhat/Fedora Linux then use yum:

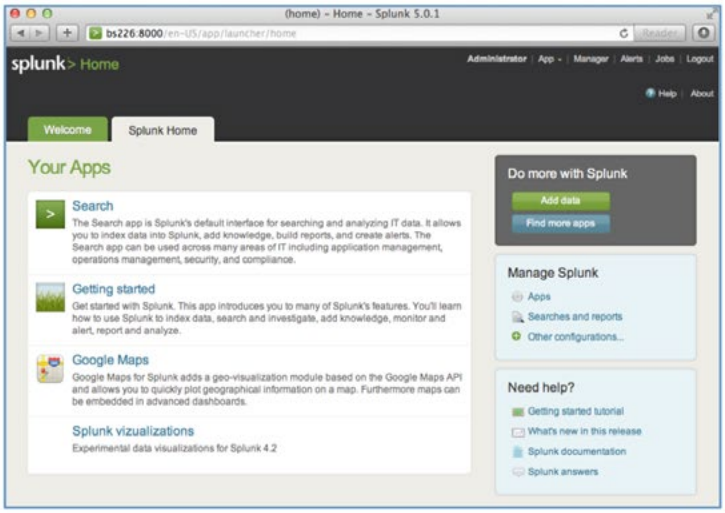
```
[root@splunk-server ~]# yum –y install wireshark*
```

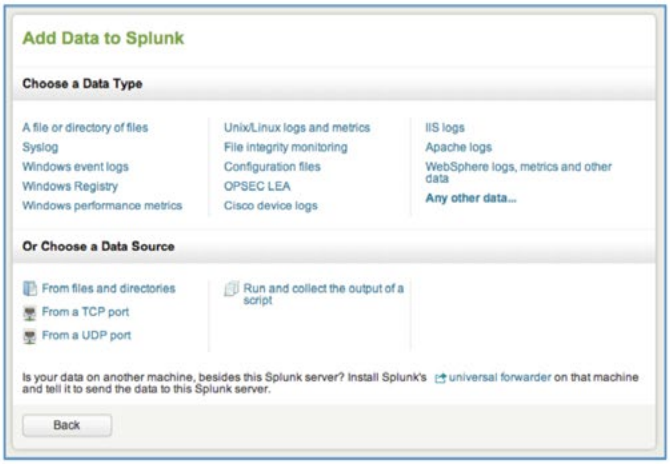
Once sFlow is enabled on the switch you should see the sampled packets arriving on the Splunk server. You can validate this by running the /opt/splunk/bin/scripts/arista_sflow helper script and validating that sampled packets are appearing:

```
[root@splunk-server ~]# /opt/splunk/bin/scripts/
arista_sflow eth1
Running as user “root” and group “root”. This could
be dangerous.
Capturing on eth1

t=1357542076 sflow_245.vlan.in=1 sflow_245.pri.
in=0 sflow_245.vlan.out=1 sflow_245.pri.out=0 eth.
dst=”00:12:bf:1a:59:b2” eth.src=”b8:ff:61:c0:74:59”
eth.type=”0x0800” ip.version=4 ip.hdr_len=20
ip.dsfield=0 ip.dsfield.dscp=”0x00” ip.dsfield.
ect=0 ip.dsfield.ce=0 ip.len=64 ip.id=”0x43c5”
ip.flags=”0x02” ip.flags.rb=0 ip.flags.df=1 ip.flags.
mf=0 ip.frag_offset=0 ip.ttl=64 ip.proto=6
ip.checksum_good=1 ip.checksum_bad=0 [...]
```

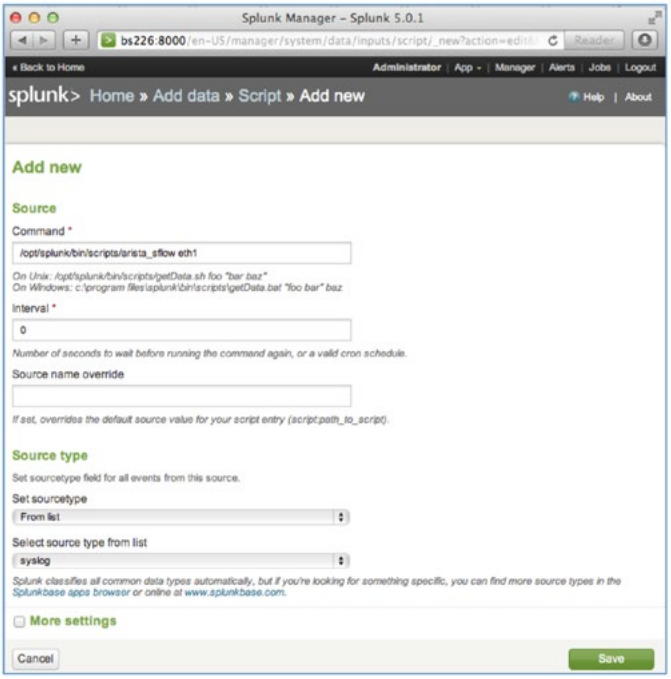
The final step is to enable this helper script as a data source within Splunk. In the Splunk management interface, select ‘Add data’ (green button below) then ‘Run and collect the output of a script’:





The extensibility of Arista EOS combined with the reporting and visualization capabilities of Splunk Enterprise provides a distributed, powerful and cost effective way to gain visibility into network traffic, with ad-hoc queries, graphs and reports capable of being generated in real time.

Configure the new script to call /opt/splunk/bin/scripts/arista_sflow as a script data source on the 'eth1' interface, an interval of '0' and a data format of 'syslog':



At this point, Splunk is operational and data is being collected.

Summary

The extensibility of Arista EOS combined with the reporting and visualization capabilities of Splunk Enterprise provides a distributed, powerful and cost effective way to gain visibility into network traffic, with ad-hoc queries, graphs and reports capable of being generated in real time. This solution provides powerful network analysis without the need for costly hardware upgrades or proprietary network analysis modules.

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office
1390 Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062