Collecting Wire Data at *Household Speeds*

James Callahan
(JP)

UScontracting Inc.

https://www.uscontractinginc.com/

Disclaimer

Please read the disclaimer carefully before you continue to view this presentation. I'm only going to give you less than 5 seconds, so I hope you can read fast. By attending this presentation, or viewing the saved version online later, you agree to the Terms of Use when this option is made available to you, you accept and agree to be bound and abide by the disclaimer. If you do not want to agree to this disclaimer, feel free to leave. Although no refunds will be issued, and you might hurt my feelings. All opinions and comments are the opinions and comments of the presenter and the inclusion in this presentation is in no way to be construed as or imply any endorsement or affiliation with his employer or any client organizations past, present or future. The information contained on this presentation and the resources linked through this presentation are for educational and informational purposes only. Your viewing of this presentation – including implementation of any suggestions set out in this presentation and/or use of any resources available on this presentation – does not create a professional-client relationship between you and me or my current past or future employers or client organizations. By using the information in this presentation, you accept personal responsibility for the results of your actions. You agree to take full responsibility for any harm or damage you suffer as a result of the use, or non-use, of the information available to you in this presentation or the resources available for download discussed in this presentation. You agree to use judgment and conduct due diligence before taking any action or implementing any of the items in this presentation. THE PRESENTER IS ABSOLVED OF ANY AND ALL LIABILITY OR LOSS THAT YOU OR ANY PERSON OR ENTITY ASSOCIATED WITH YOU MAY SUFFER OR INCUR AS A RESULT OF USE OF THE INFORMATION CONTAINED IN THIS PRESENTATION AND/OR THE RESOURCES YOU MAY DOWNLOAD FROM THAT ARE MENTIONED HEREIN. THE PRESENTER SHALL NOT BE LIABLE TO YOU FOR ANY TYPE OF DAMAGES, INCLUDING DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EQUITABLE, OR CONSEQUENTIAL LOSS OR DAMAGES FOR USE OF ANY OF THE INFOMATION PRESENTED. THE INFORMATION, SOFTWARE, PRODUCTS, AND SERVICES DISCUSSED MAY INCLUDE INACCURACIES OR TYPOGRAPHICAL ERRORS. YOU CAN EXPECT THIS FROM THE PRESENTER, AS HIS BRAIN MOVES SLOWER THAN HIS FINGERS TYPES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL THE PRESENTER BE LIABLE FOR ANY DIRECT, INDIRECT, PUNITIVE, INCIDENTAL, SPECIAL, CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF USE, DATA OR PROFITS, ARISING OUT OF OR IN ANY WAY CONNECTED WITH THE USE OF THE INFOMRATION IN THIS PRESENTATION, WITH THE DELAY OR INABILITY TO USE INFORMATION, THE PROVISION OF OR FAILURE TO PROVIDE SERVICES, OR FOR ANY INFORMATION, SOFTWARE, PRODUCTS, SERVICES AND RELATED GRAPHICS OBTAINED THROUGH THE PRESENTATION, OR OTHERWISE ARISING OUT OF THE USE OF THE INFOMRATION IN THIS PRESENTATION, WHETHER BASED ON CONTRACT, TORT, NEGLIGENCE, STRICT LIABILITY OR OTHERWISE, EVEN IF THE PRESENTOR HAS BEEN ADVISED OF THE POSSIBILITY OF DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IF YOU ARE DISSATISFIED WITH ANY PORTION OF THE PRESENTATION, OR WITH ANY OF THESE TERMS OF USE, YOUR SOLE AND EXCLUSIVE REMEDY IS TO USE THE MEN IN BLACK'S NEURALIZER (THE FLASHY THING) TO ERASE THIS HORRIBLE MEMORY.

Disclaimer

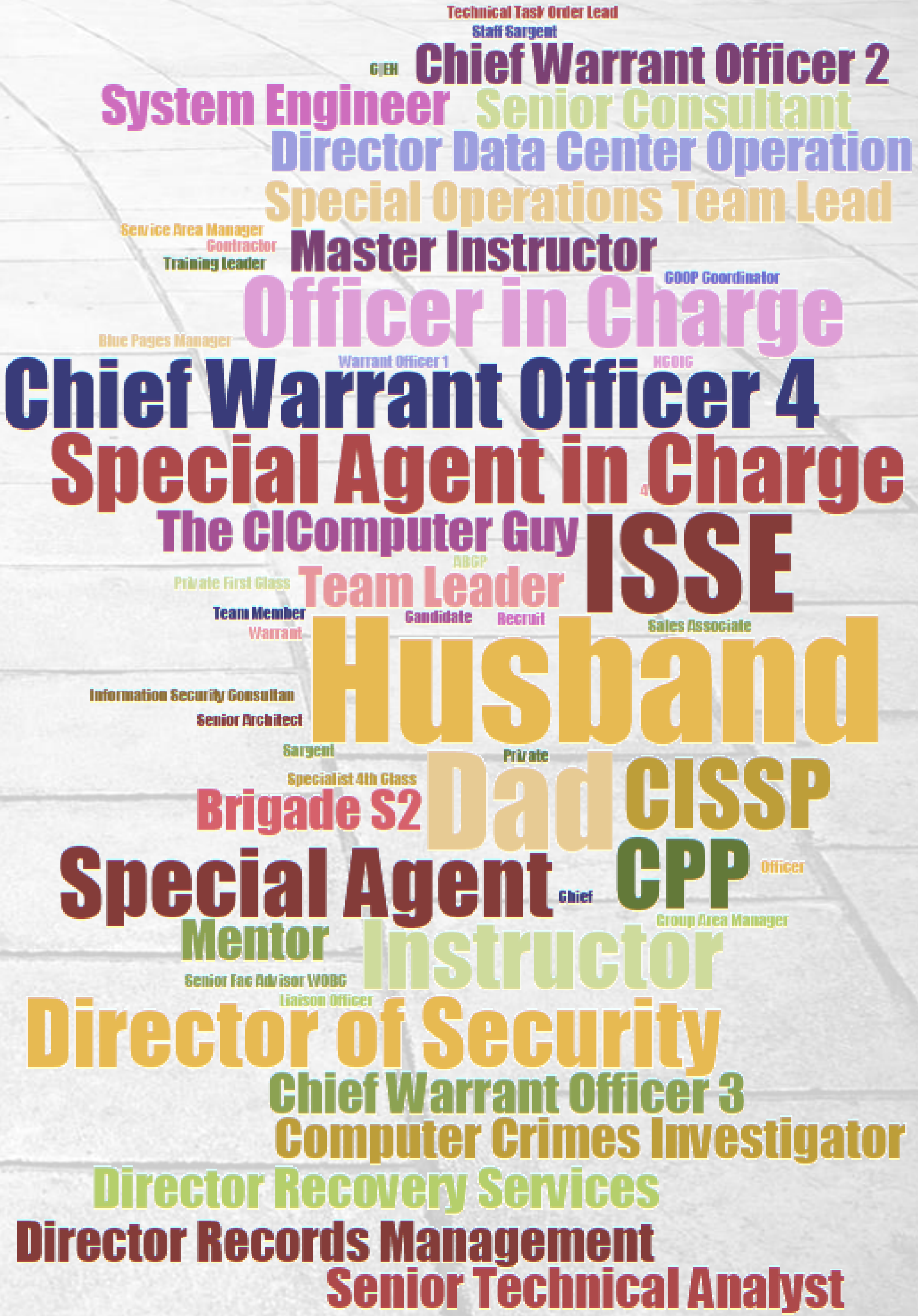
Please read the disclaimer carefully before you continue to view this presentation. I'm only going to give you less than 5 seconds, so I hope you can read fast. By attending this presentation, or viewing the saved version online later, you agree to the Terms of Use when this option is made available to you, you accept and agree to be bound and abide by the disclaimer. If you do not want to agree to this disclaimer, feel free to leave. Although no refunds will be issued, and you might hurt [illegible] ns and comments are [illegible] opinions and comments of the presenter and the inclusion in this presentation is in no way t [illegible] endorsem [illegible] his employer or any client organizations past, present or future. The information containe [illegible] esource [illegible] tation are for educational and informational purposes only. Your viewing of this prese [illegible] ation c [illegible] presentation and/or use of any resources available on this presentation – does not crea [illegible] elations [illegible] ent past or future employers or client organizations. By using the information in this presenta [illegible] ot person [illegible] ur actions. You agree to take full responsibility for any harm or damage you suffer as a result of the use, o [illegible] on-us [illegible] in this presentation or the resources available for download discussed in this presentation. You agree to [illegible] se ju [illegible] efore taking any action or implementing any of the items in this presentation. THE PRESENTER IS ABSOLVED OF ANY A [illegible] U OR ANY PERSON OR ENTITY ASSOCIATED WITH YOU MAY SUFFER OR INCUR AS A RESULT OF USE OF THE INFORMATIO [illegible] N AND/OR THE RESOURCES YOU MAY DOWNLOAD FROM THAT ARE MENTIONED HEREIN. THE PRESENTER SHALL NOT [illegible] F DAMAGES, INCLUDING DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EQUITABLE, OR CONSEQUENTIAL LOSS OR DAMA [illegible] FOMATION PRESENTED. THE INFORMATION, SOFTWARE, PRODUCTS, AND SERVICES DISCUSSED MAY INCLUDE INACC [illegible] ERRORS. YOU CAN EXPECT THIS FROM THE PRESENTER, AS HIS BRAIN MOVES SLOWER THAN HIS FINGERS TYPES. TO THE [illegible] BY APPLICABLE LAW, IN NO EVENT SHALL THE PRESENTER BE LIABLE FOR ANY DIRECT, INDIRECT, PUNITIVE, IN [illegible] ECIAL, [illegible] AL [illegible] ANY DAMAGES WHATSOEVER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF USE, DATA [illegible] RISING OUT OF OR IN A [illegible] CTED WITH THE USE OF THE INFOMRATION IN THIS PRESENTATION, WITH THE DELAY OR INABILITY [illegible] IATION, THE PROVISIO [illegible] TO PROVIDE SERVICES, OR FOR ANY INFORMATION, SOFTWARE, PRODUCTS, SERVIC [illegible] ED GRAPHICS OBTAINED [illegible] PRESENTATION, OR OTHERWISE ARISING OUT OF THE USE OF THE INFOMRATION IN THIS PRESENTATION, WHETHER BASED ON CONTRACT, TORT, NEGLIGENCE, STRICT LIABILITY OR OTHERWISE, EVEN IF THE PRESENTOR H [illegible] USION OR LIMITATION OF LIA [illegible] DISSATISFIED WITH ANY PORTION OF THE PRESENTATION, OR WITH ANY OF THESE TERMS OF USE, YOUR SOLE AND EXCLUSIVE REMEDY IS TO USE THE MEN IN BLACK'S NEURALIZER (THE FLASHY THING) TO ERASE THIS HORRIBLE MEMORY.

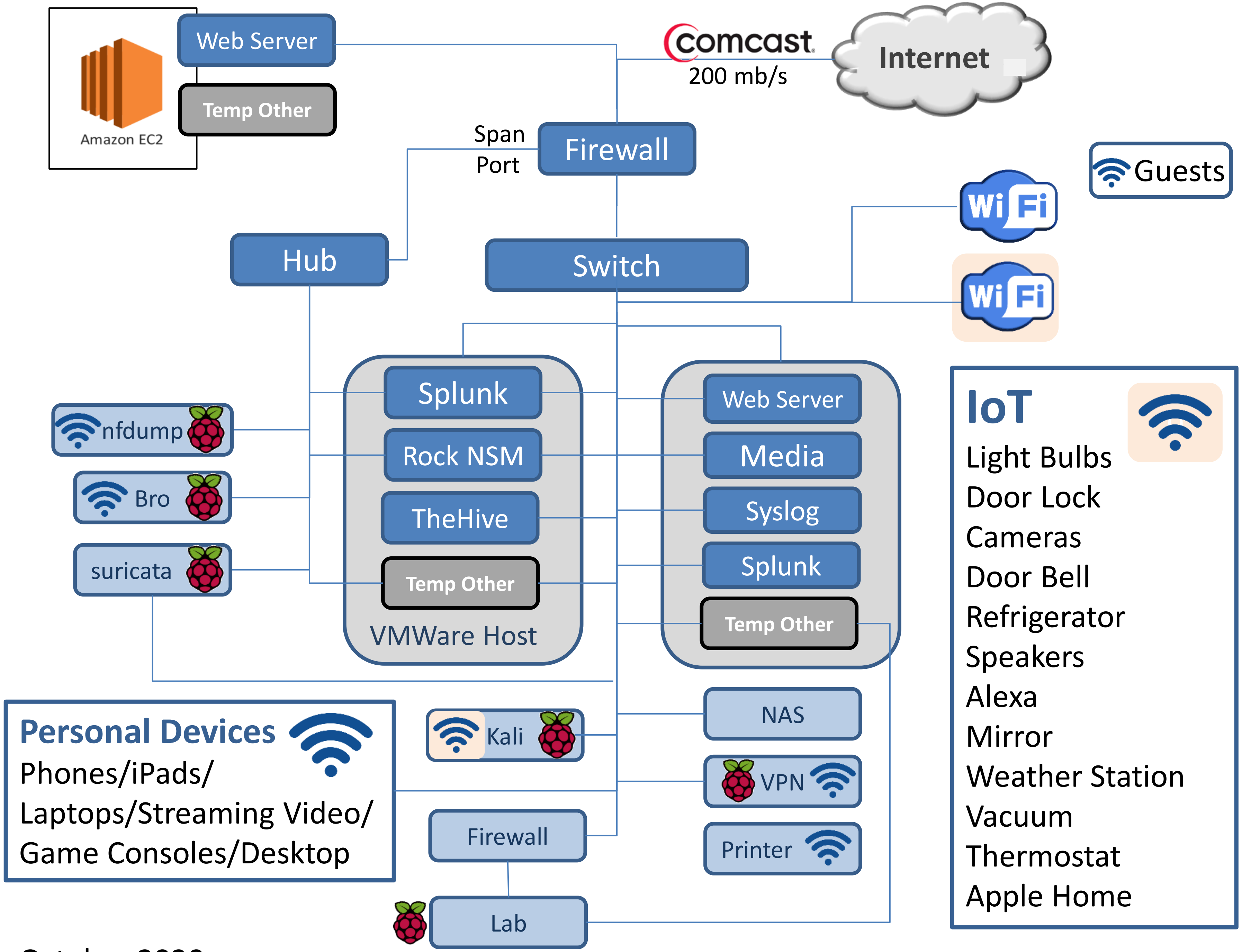
Not representing my company or any client organizations

About JP

- Professional Paranoid
- Retired US Army CW4
- Computer Hobbyist
- Irish Band and Solo Artist
- ‘65 Mustang
- Not looking for a job

If you’re not having fun, you’re in the wrong line of work.





Amazon EC2

Web Server

Temp Other

comcast

200 mb/s

Internet

Span Port

Firewall

Hub

Switch

Wi Fi

Wi Fi

Guests

nfdump

Bro

suricata

Splunk

Rock NSM

TheHive

Temp Other

VMWare Host

Web Server

Media

Syslog

Splunk

Temp Other

IoT

Light Bulbs
Door Lock
Cameras
Door Bell
Refrigerator
Speakers
Alexa
Mirror
Weather Station
Vacuum
Thermostat
Apple Home

Personal Devices

Phones/iPads/
Laptops/Streaming Video/
Game Consoles/Desktop

Kali

NAS

VPN

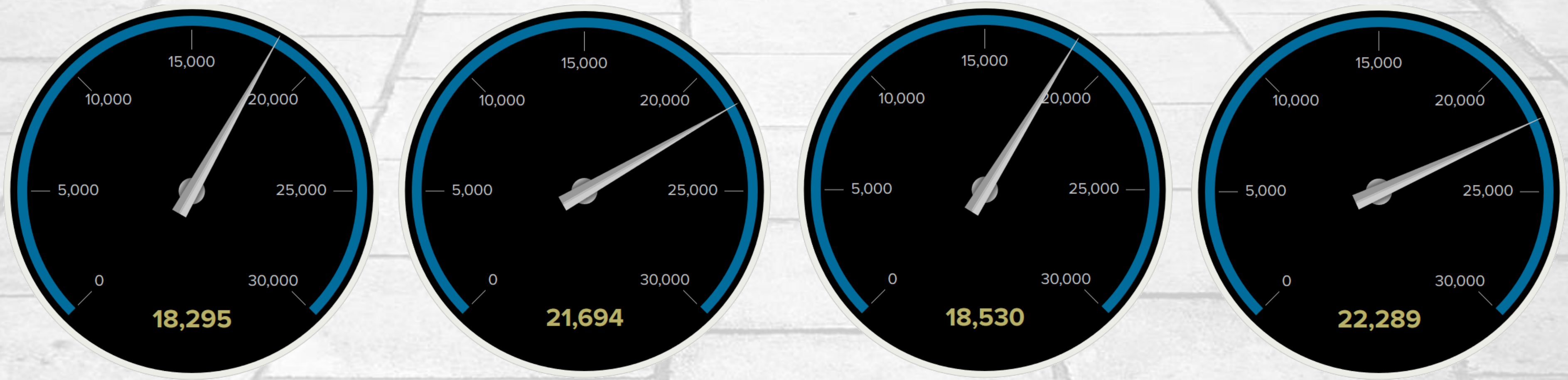
Firewall

Printer

Lab

October 2020

Running four netflow collectors



nfdump

bro

dedup uid

Suricata

dedup flow_id

Stream App

dedup flow_id

netflow collected for the same 60 min time frame

fprobe->nfcapd->nfdump

- Command Line Install
  - **fprobe** is the capture daemon
  - **nfcapd** writes what's captured - does not have to be on the same machine as fprobe
  - **nfdump** takes the nfcapd files and makes them human readable in configurable formats
- Running on Pi Model B+ with 512M (Rev. 1.2)
- Been running for more than 7 years

- **No bells and whistles**
- **Light weight – survey/assessment tool**
- **No native metrics on how it’s performing**
- **Not real time – ‘harvest’ script every 15 min – but can be adjusted**
- **Ran on battery for about 7 hours**

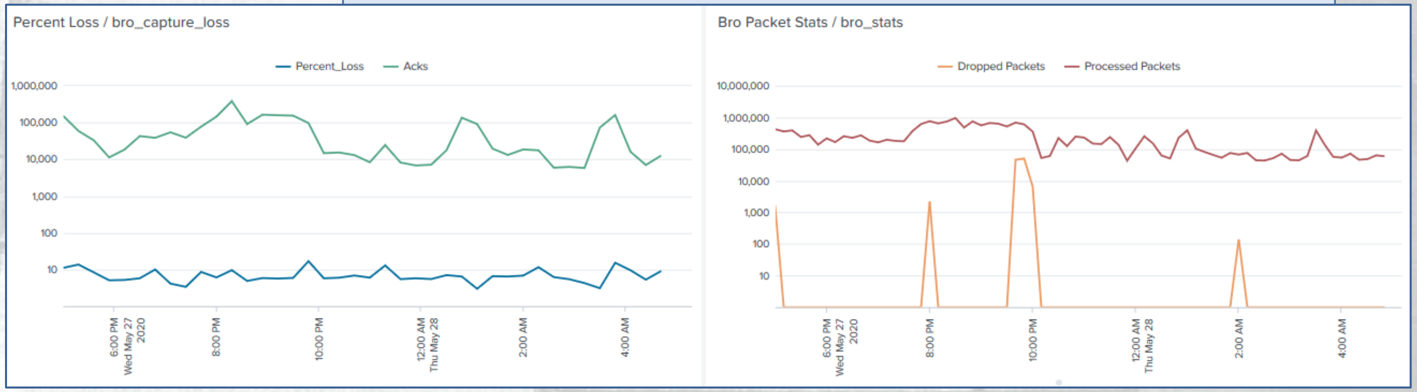
```
/usr/bin/nfcapd -p 2055 -l /netflow/current -D
/usr/sbin/fprobe -ieth1 127.0.0.1:2055
```

https://pandorafms.com/blog/netflow-probe-using-raspberry/

bro-2.5.3

- Easy setup, a lot on by default
  - Had to tweak some source types to reduce volume
- Running older version (for now)
- Running on Pi Zero Rev 1.3
- Configured for json output
- Upgrade planned

- Tap is keeping up
- Pi had 8% loss at peak
- Lots of data source types


Percent Loss / bro_capture_loss

Bro Packet Stats / bro_stats

https://bløgg.no/2015/11/installing-bro-the-network-security-monitor-on-raspberry-pi/

https://software.opensuse.org/download.html?project=security%3Azeek&package=zeek

Suricata

- BriarIDS
  - Built for raspberry pi's Raspbian OS
  - Suricata version, 4.0.4
- BriarIDS Also
  - Includes Bro
  - Support for md5 and sha256 file hashing
    (malicious file detection)
  - Alienvault intel feeds for Bro
- Running on Pi Model B2 with 1G (Rev 1.1)
  - Recently (2020) had to reimage new sd card

- Netflow wasn't on by default
  - Two Options
  - Uni or Bi directional
- Using it primarily for IDS
  - Emerging Threat Rules update every night
  - Oinkmaster Scripts

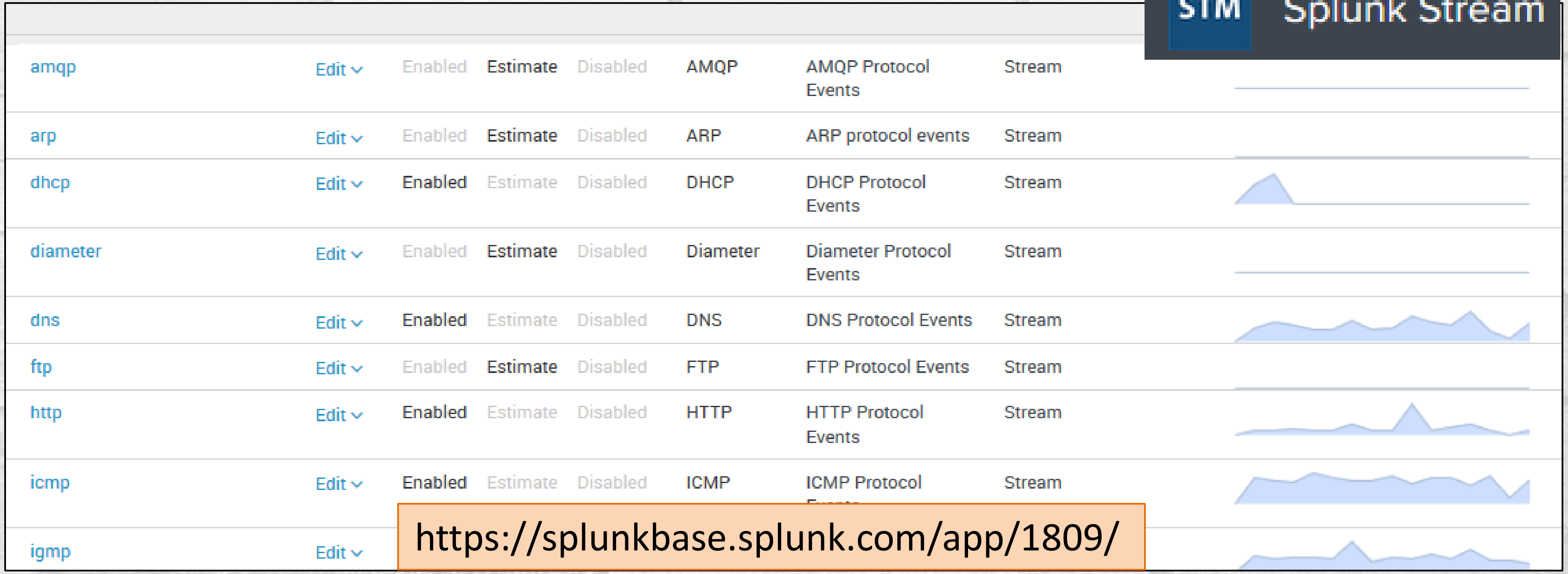
https://github.com/musicmancorley/BriarIDS

Splunk Stream App

- Native Splunk App (Free add on)
- Passively capture live streams of network event data.
- Extract files from network traffic.
- Network trends and app performance in pre-built dashboards.
- VM on ESXi (with Splunk indexer and search head)
  - CPUs 1 (~1.7 GHz consumed)
  - Memory 4 GB (~3.82 GB consumed)
  - 300 GB on SSD

- Highly configurable
- Watch your ingest
- Can read PCAP files as source
- Can be installed on a forwarder

STM Splunk Stream



amqp	Edit ⌄	Enabled	Estimate	Disabled	AMQP	AMQP Protocol Events	Stream	
arp	Edit ⌄	Enabled	Estimate	Disabled	ARP	ARP protocol events	Stream	
dhcp	Edit ⌄	Enabled	Estimate	Disabled	DHCP	DHCP Protocol Events	Stream	
diameter	Edit ⌄	Enabled	Estimate	Disabled	Diameter	Diameter Protocol Events	Stream	
dns	Edit ⌄	Enabled	Estimate	Disabled	DNS	DNS Protocol Events	Stream	
ftp	Edit ⌄	Enabled	Estimate	Disabled	FTP	FTP Protocol Events	Stream	
http	Edit ⌄	Enabled	Estimate	Disabled	HTTP	HTTP Protocol Events	Stream	
icmp	Edit ⌄	Enabled	Estimate	Disabled	ICMP	ICMP Protocol Events	Stream	
igmp	Edit ⌄							

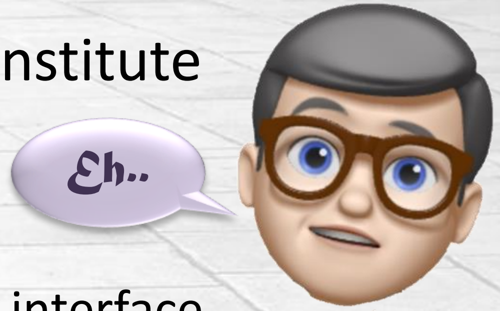
https://splunkbase.splunk.com/app/1809/

tcpdump (retired)

- Collected full PCAP from raspberry PI
  - Moved files to second host and read the flows out to text files
  - Was not real time ~ 15 min delay.
- Played with extracted ARP traffic
  - Killed my ingest
- Cascading PCAP Problem
  - Raspberry Pi only collected the PCAP
  - Pushed to another box on the network to convert to flow data
  - Collected second time as the raw PCAP files were moved
  - Added filters to exclude

Carnegie Mellon Software Engineering Institute

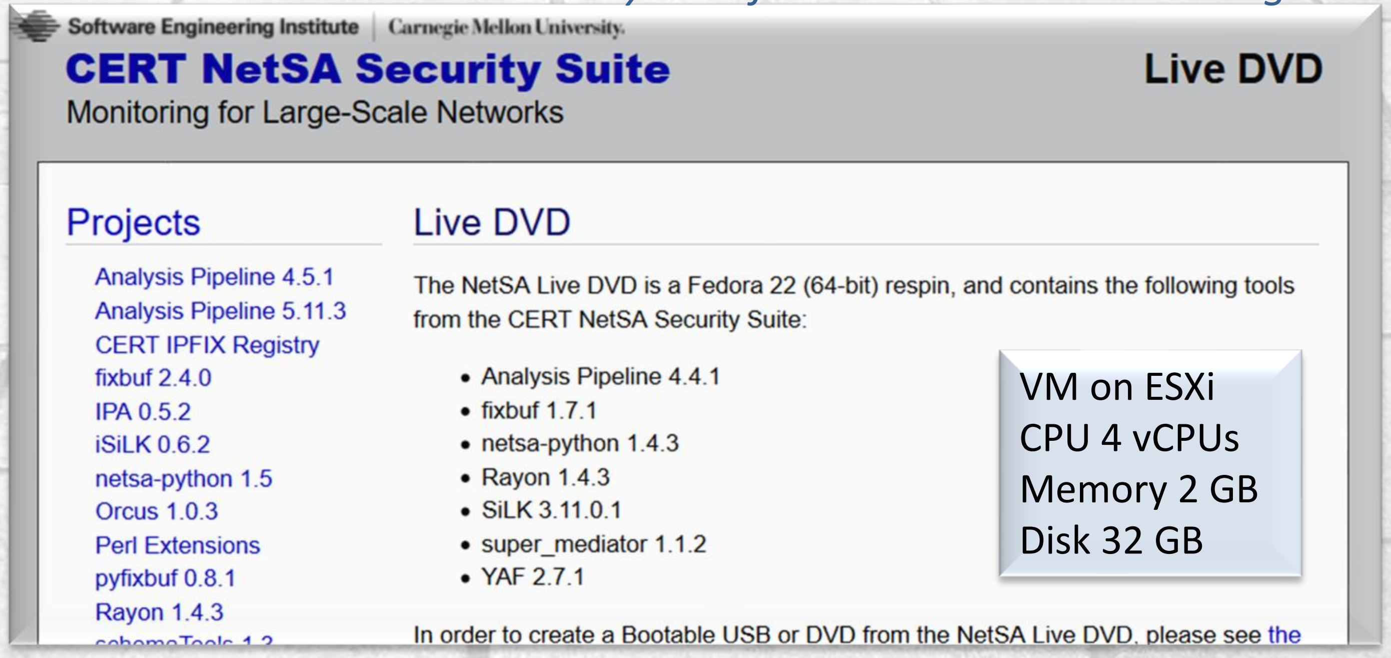
(dabbled not deployed) YAF->SiLK



- Optimized for extremely large data collections
- Very compact record format (more data stored in less space)

- Command line interface
- Most useful for incident forensics
- Silk: 80+ CLI Commands

Yet Another Flowmeter *System for Internet-Level Knowledge*



Software Engineering Institute | Carnegie Mellon University

CERT NetSA Security Suite

Monitoring for Large-Scale Networks

Live DVD

Projects

- Analysis Pipeline 4.5.1
- Analysis Pipeline 5.11.3
- CERT IPFIX Registry
- fixbuf 2.4.0
- IPA 0.5.2
- iSiLK 0.6.2
- netsa-python 1.5
- Orcus 1.0.3
- Perl Extensions
- pyfixbuf 0.8.1
- Rayon 1.4.3
- schemaTools 1.2

Live DVD

The NetSA Live DVD is a Fedora 22 (64-bit) respin, and contains the following tools from the CERT NetSA Security Suite:

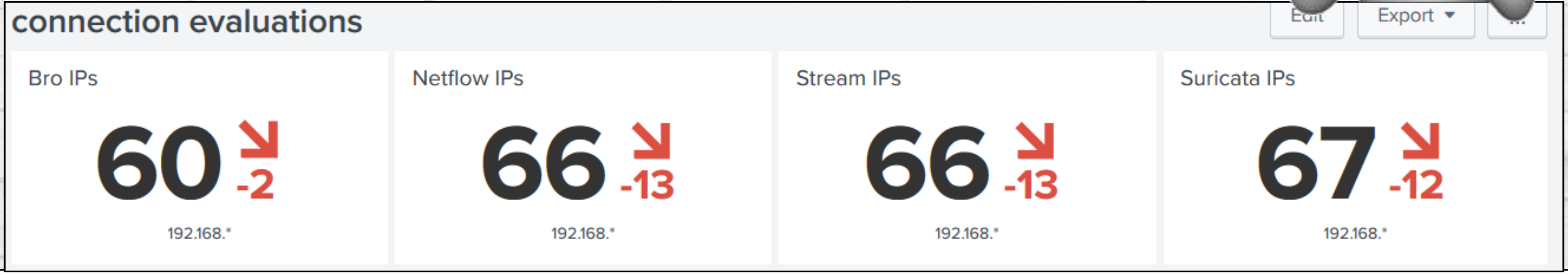
- Analysis Pipeline 4.4.1
- fixbuf 1.7.1
- netsa-python 1.4.3
- Rayon 1.4.3
- SiLK 3.11.0.1
- super_mediator 1.1.2
- YAF 2.7.1

In order to create a Bootable USB or DVD from the NetSA Live DVD, please see the

VM on ESXi
CPU 4 vCPUs
Memory 2 GB
Disk 32 GB

https://tools.netsa.cert.org/livedvd.html
https://tools.netsa.cert.org/silk/silk-quickref.pdf

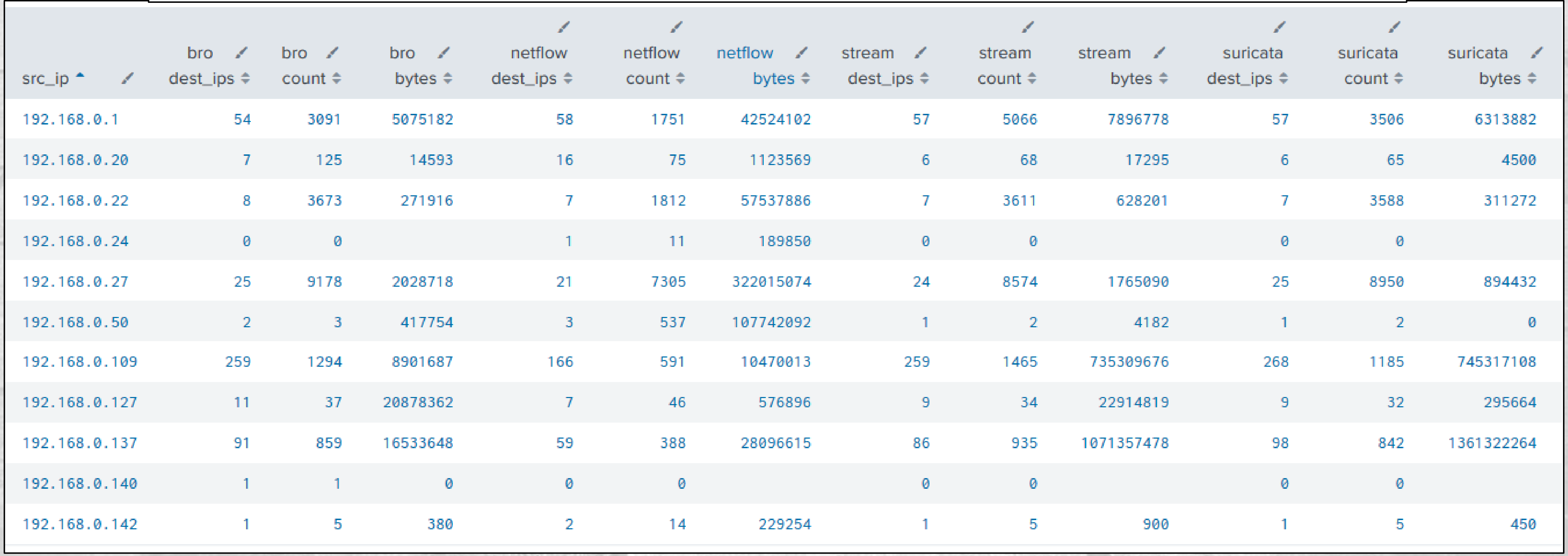
So, how do they do?



connection evaluations

Bro IPs	Netflow IPs	Stream IPs	Suricata IPs
60 ↘ -2	66 ↘ -13	66 ↘ -13	67 ↘ -12
192.168.*	192.168.*	192.168.*	192.168.*

24 hours of select IPs

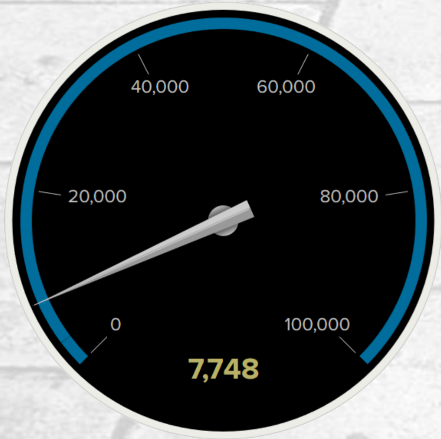


src_ip	bro dest_ips	bro count	bro bytes	netflow dest_ips	netflow count	netflow bytes	stream dest_ips	stream count	stream bytes	suricata dest_ips	suricata count	suricata bytes
192.168.0.1	54	3091	5075182	58	1751	42524102	57	5066	7896778	57	3506	6313882
192.168.0.20	7	125	14593	16	75	1123569	6	68	17295	6	65	4500
192.168.0.22	8	3673	271916	7	1812	57537886	7	3611	628201	7	3588	311272
192.168.0.24	0	0		1	11	189850	0	0		0	0	
192.168.0.27	25	9178	2028718	21	7305	322015074	24	8574	1765090	25	8950	894432
192.168.0.50	2	3	417754	3	537	107742092	1	2	4182	1	2	0
192.168.0.109	259	1294	8901687	166	591	10470013	259	1465	735309676	268	1185	745317108
192.168.0.127	11	37	20878362	7	46	576896	9	34	22914819	9	32	295664
192.168.0.137	91	859	16533648	59	388	28096615	86	935	1071357478	98	842	1361322264
192.168.0.140	1	1	0	0	0		0	0		0	0	
192.168.0.142	1	5	380	2	14	229254	1	5	900	1	5	450

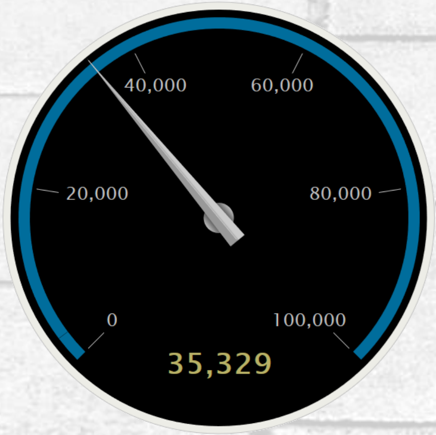
Netflow Collector Metrics

Not all collectors are equal

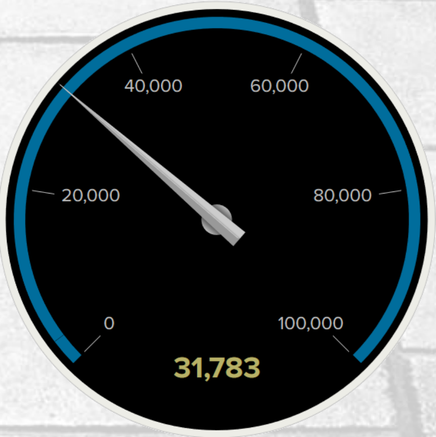
- Disparities in Flows; Protocols; Formats; Break downs
- Destination IP, Event Count and Bytes by Source IP
  - Different hardware platforms (bro is on a Pi Zero)
  - Different data details (collected fields)
  - Different data format & granularity



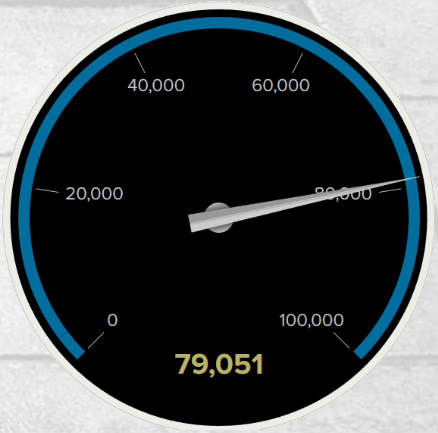
nfdump



bro



Suricata

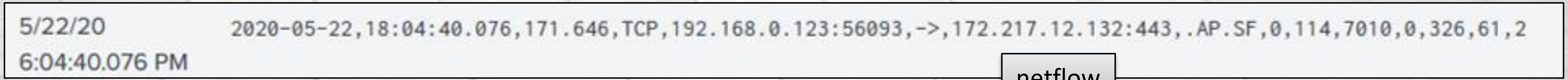


Stream App

Total of all (including non netflow) events ingested in same 60 minutes.

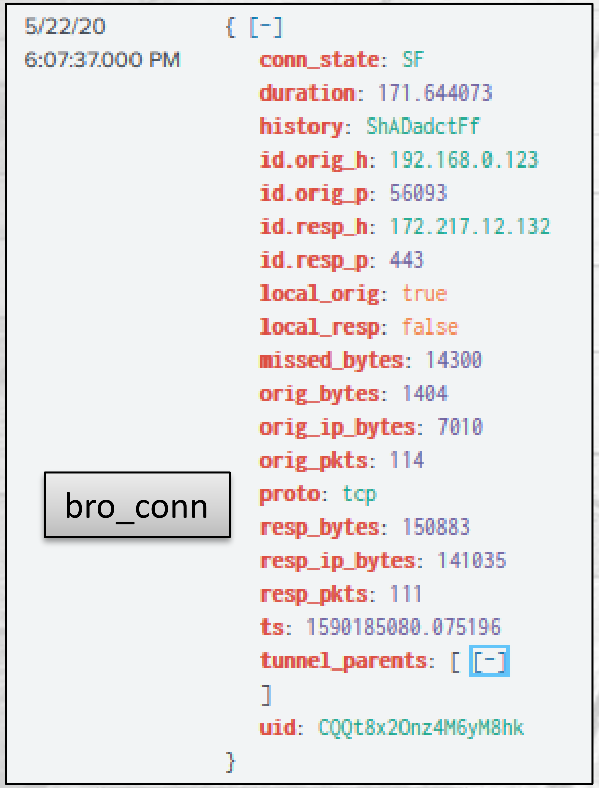
Data Format
TCP

- Varying levels of granularity



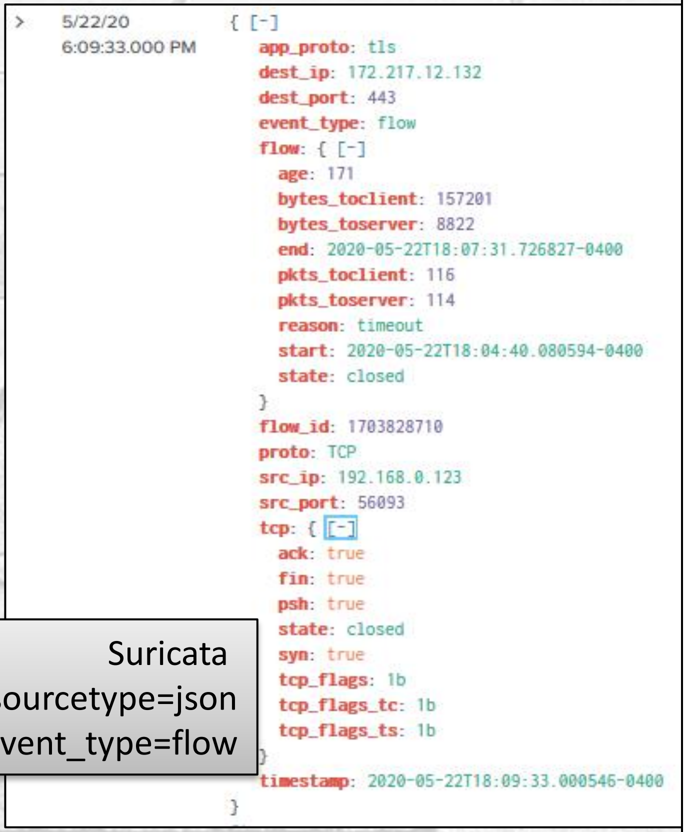
```
5/22/20           2020-05-22,18:04:40.076,171.646,TCP,192.168.0.123:56093,->,172.217.12.132:443,.AP.SF,0,114,7010,0,326,61,2
6:04:40.076 PM
```

netflow



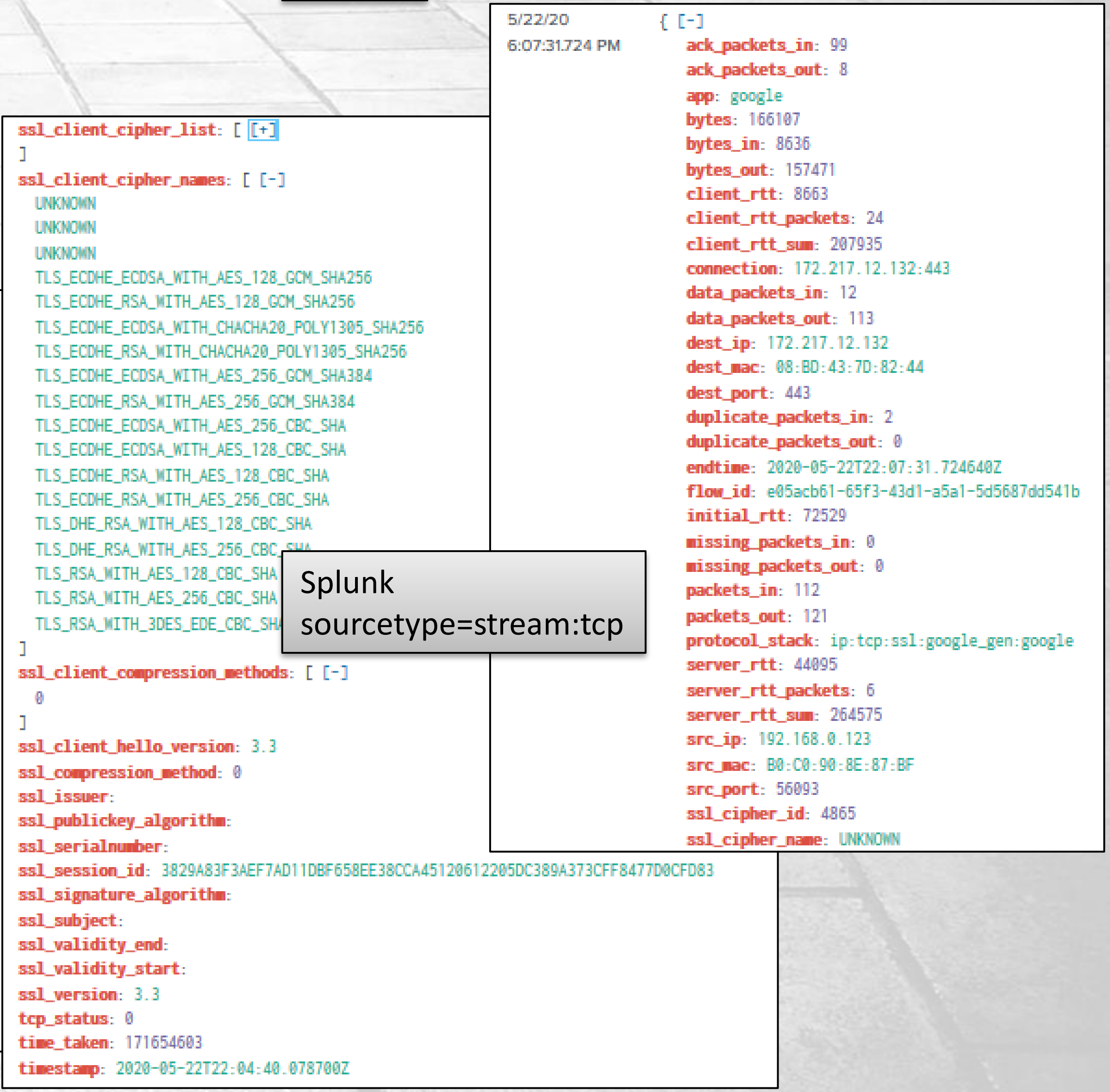
```
5/22/20              { [-]
6:07:37.000 PM          conn_state: SF
                        duration: 171.644073
                        history: ShADadctFf
                        id.orig_h: 192.168.0.123
                        id.orig_p: 56093
                        id.resp_h: 172.217.12.132
                        id.resp_p: 443
                        local_orig: true
                        local_resp: false
                        missed_bytes: 14300
                        orig_bytes: 1404
                        orig_ip_bytes: 7010
                        orig_pkts: 114
                        proto: tcp
                        resp_bytes: 150883
                        resp_ip_bytes: 141035
                        resp_pkts: 111
                        ts: 1590185080.075196
                        tunnel_parents: [ [-]
                        ]
                        uid: CQQt8x2Onz4M6yM8hk
                     }
```

bro_conn



```
>   5/22/20              { [-]
    6:09:33.000 PM          app_proto: tls
                            dest_ip: 172.217.12.132
                            dest_port: 443
                            event_type: flow
                            flow: { [-]
                              age: 171
                              bytes_toclient: 157201
                              bytes_toserver: 8822
                              end: 2020-05-22T18:07:31.726827-0400
                              pkts_toclient: 116
                              pkts_toserver: 114
                              reason: timeout
                              start: 2020-05-22T18:04:40.080594-0400
                              state: closed
                            }
                            flow_id: 1703828710
                            proto: TCP
                            src_ip: 192.168.0.123
                            src_port: 56093
                            tcp: { [-]
                              ack: true
                              fin: true
                              psh: true
                              state: closed
                              syn: true
                              tcp_flags: 1b
                              tcp_flags_tc: 1b
                              tcp_flags_ts: 1b
                            }
                            timestamp: 2020-05-22T18:09:33.000546-0400
                         }
```

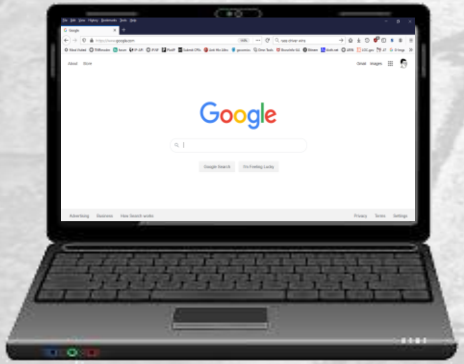
Suricata
sourcetype=json
event_type=flow



```
5/22/20              { [-]
6:07:31.724 PM          ack_packets_in: 99
                        ack_packets_out: 8
                        app: google
                        bytes: 166107
                        bytes_in: 8636
                        bytes_out: 157471
                        client_rtt: 8663
                        client_rtt_packets: 24
                        client_rtt_sum: 207935
                        connection: 172.217.12.132:443
                        data_packets_in: 12
                        data_packets_out: 113
                        dest_ip: 172.217.12.132
                        dest_mac: 08:BD:43:7D:82:44
                        dest_port: 443
                        duplicate_packets_in: 2
                        duplicate_packets_out: 0
                        endtime: 2020-05-22T22:07:31.724640Z
                        flow_id: e05acb61-65f3-43d1-a5a1-5d5687dd541b
                        initial_rtt: 72529
                        missing_packets_in: 0
                        missing_packets_out: 0
                        packets_in: 112
                        packets_out: 121
                        protocol_stack: ip:tcp:ssl:google_gen:google
                        server_rtt: 44095
                        server_rtt_packets: 6
                        server_rtt_sum: 264575
                        src_ip: 192.168.0.123
                        src_mac: B0:C0:90:8E:87:BF
                        src_port: 56093
                        ssl_cipher_id: 4865
                        ssl_cipher_name: UNKNOWN
```

```
ssl_client_cipher_list: [ [+]
]
ssl_client_cipher_names: [ [-]
  UNKNOWN
  UNKNOWN
  UNKNOWN
  TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
  TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
  TLS_DHE_RSA_WITH_AES_128_CBC_SHA
  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
  TLS_RSA_WITH_AES_128_CBC_SHA
  TLS_RSA_WITH_AES_256_CBC_SHA
  TLS_RSA_WITH_3DES_EDE_CBC_SH
]
ssl_client_compression_methods: [ [-]
  0
]
ssl_client_hello_version: 3.3
ssl_compression_method: 0
ssl_issuer:
ssl_publickey_algorithm:
ssl_serialnumber:
ssl_session_id: 3829A83F3AEF7AD11DBF658EE38CCA45120612205DC389A373CFF8477D0CFD83
ssl_signature_algorithm:
ssl_subject:
ssl_validity_end:
ssl_validity_start:
ssl_version: 3.3
tcp_status: 0
time_taken: 171654603
timestamp: 2020-05-22T22:04:40.078700Z
```

Splunk
sourcetype=stream:tcp

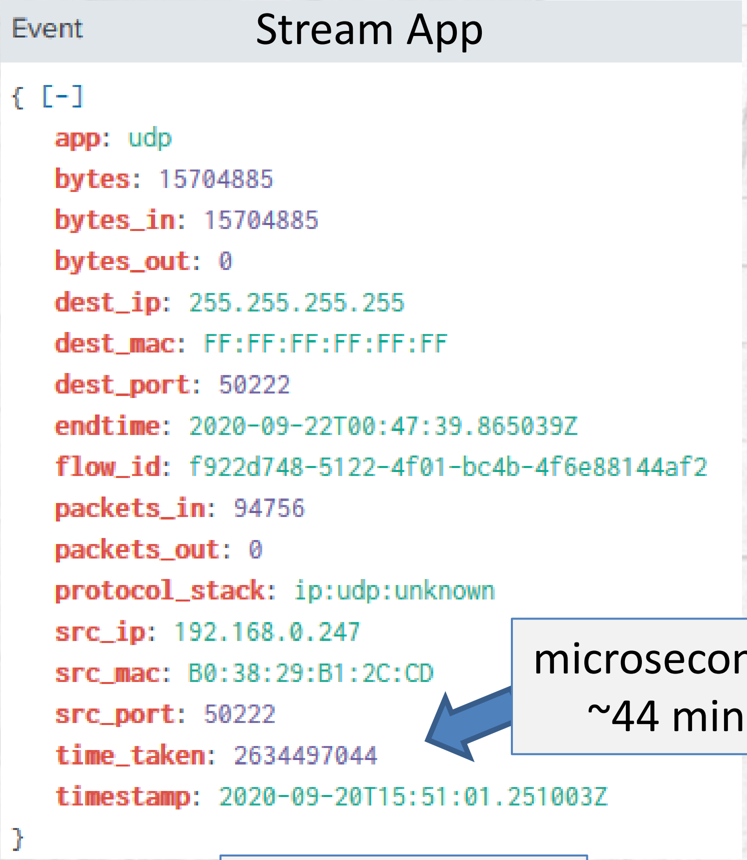


WeatherFlow

Weather station broadcasts UDP
tcpdump shows three packets in quick succession every three seconds

Data Format
UDP

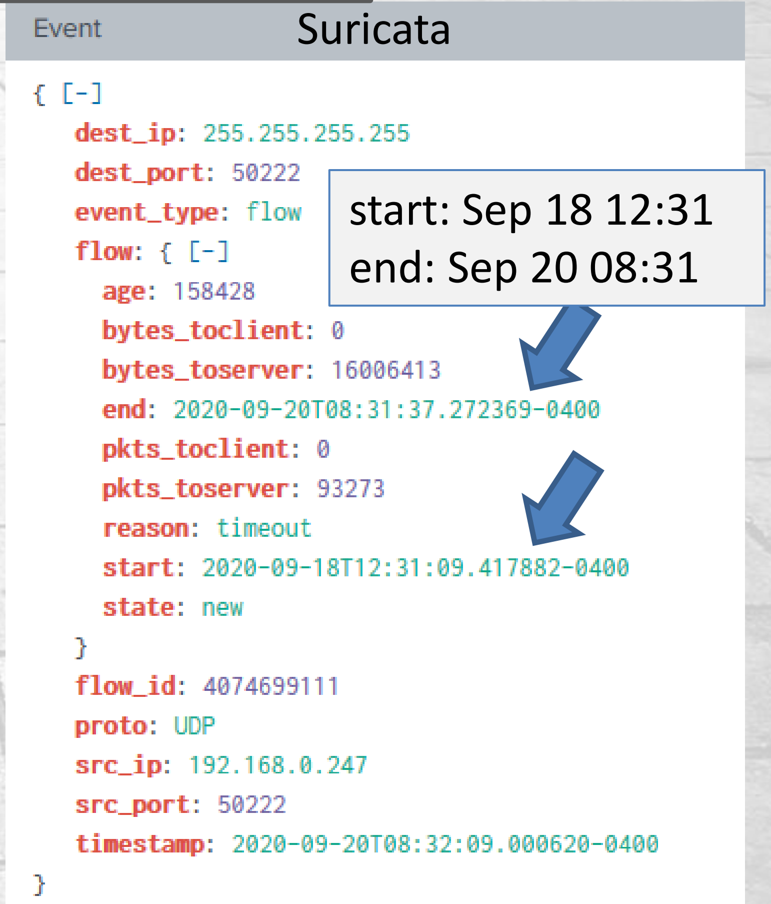
- No end of flow marker
- Can lead to delays in ingest
- Nfdump shortest write cycle
- No flows in bro_conn



Event Stream App

```
{ [-]
   app: udp
   bytes: 15704885
   bytes_in: 15704885
   bytes_out: 0
   dest_ip: 255.255.255.255
   dest_mac: FF:FF:FF:FF:FF:FF
   dest_port: 50222
   endtime: 2020-09-22T00:47:39.865039Z
   flow_id: f922d748-5122-4f01-bc4b-4f6e88144af2
   packets_in: 94756
   packets_out: 0
   protocol_stack: ip:udp:unknown
   src_ip: 192.168.0.247
   src_mac: B0:38:29:B1:2C:CD
   src_port: 50222
   time_taken: 2634497044
   timestamp: 2020-09-20T15:51:01.251003Z
}
```

microseconds
~44 min



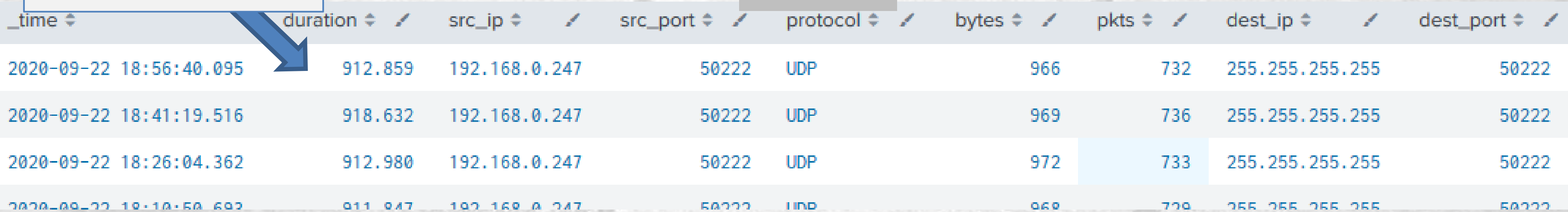
Event Suricata

```
{ [-]
   dest_ip: 255.255.255.255
   dest_port: 50222
   event_type: flow
   flow: { [-]
     age: 158428
     bytes_toclient: 0
     bytes_toserver: 16006413
     end: 2020-09-20T08:31:37.272369-0400
     pkts_toclient: 0
     pkts_toserver: 93273
     reason: timeout
     start: 2020-09-18T12:31:09.417882-0400
     state: new
   }
   flow_id: 4074699111
   proto: UDP
   src_ip: 192.168.0.247
   src_port: 50222
   timestamp: 2020-09-20T08:32:09.000620-0400
}
```

start: Sep 18 12:31
end: Sep 20 08:31

~15min intervals

nfdump



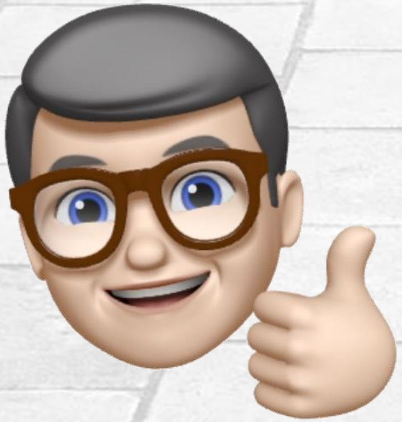
_time	duration	src_ip	src_port	protocol	bytes	pkts	dest_ip	dest_port
2020-09-22 18:56:40.095	912.859	192.168.0.247	50222	UDP	966	732	255.255.255.255	50222
2020-09-22 18:41:19.516	918.632	192.168.0.247	50222	UDP	969	736	255.255.255.255	50222
2020-09-22 18:26:04.362	912.980	192.168.0.247	50222	UDP	972	733	255.255.255.255	50222
2020-09-22 18:10:50.693	911.847	192.168.0.247	50222	UDP	968	729	255.255.255.255	50222

So what to I do with all this data?

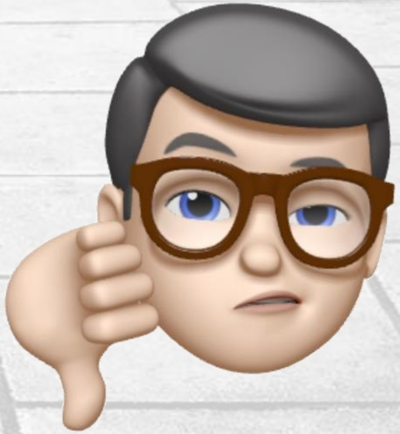
- Keep an eye on things
- Visualize things
- Learn new things
- Try new things
- Combine and compare things
- Inventory new things

But you gotta put it somewhere…

splunk>



People love it
and
people hate it.



- Dabbled with ELK stack / it wasn’t intuitive for me
- Been using Splunk at home since around 2012(ish)

Free Splunk License

- The Free license includes **500 MB/day** of indexing volume and ***Now has no expiration date***
- The following Enterprise License features are **disabled in Splunk Free:**
  - Multiple user accounts and role-based access controls
  - Distributed search
  - Forwarding in TCP/HTTP formats (you can forward data to other Splunk instances, but not to non-Splunk instances)
  - Deployment management (including for clients)
  - Alerting/monitoring
- Started with this, preserving conf files between installs

Splunk Developer and Developer/Test License

enables exploration of new non-production uses of Splunk Enterprise
two purposes/license terms/similar attributes

- For internal, non-production use
- Limited to 10 GB (Dev) or 50 GB (Dev/Test) per day
- Good for six months, then can renew
- Assigned to individuals, not organizations
- Dev available to non-customers (prospects)

“The program enables individual users within your organization to experiment with new data sources, as well as encourage others in the organization to try out the Splunk platform in a frictionless manner.”

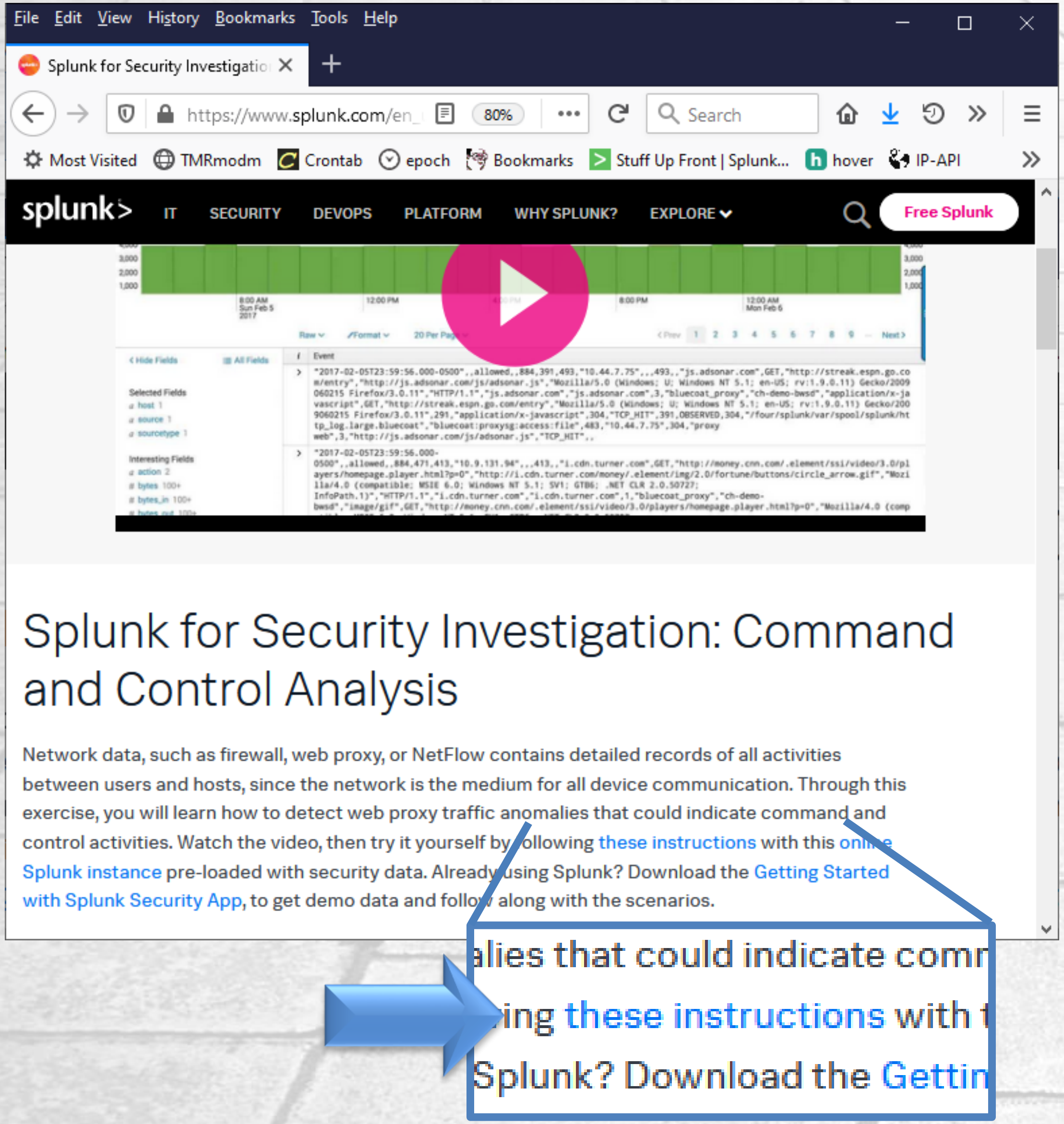
https://dev.splunk.com/enterprise/dev_license/
https://www.splunk.com/en_us/resources/personalized-dev-test-licenses.html

Splunk Cloud Trial

- 5 GB per day ingest
- 15 day duration
- Designed to transition trial instance to a production account

- Not using this for home network

Online Experience


File Edit View History Bookmarks Tools Help
Splunk for Security Investigatio
https://www.splunk.com/en_
80%
Search
Most Visited
TMRmodm
Crontab
epoch
Bookmarks
Stuff Up Front | Splunk...
hover
IP-API
splunk>
IT
SECURITY
DEVOPS
PLATFORM
WHY SPLUNK?
EXPLORE
Free Splunk

Splunk for Security Investigation: Command and Control Analysis

Network data, such as firewall, web proxy, or NetFlow contains detailed records of all activities between users and hosts, since the network is the medium for all device communication. Through this exercise, you will learn how to detect web proxy traffic anomalies that could indicate command and control activities. Watch the video, then try it yourself by following these instructions with this online Splunk instance pre-loaded with security data. Already using Splunk? Download the Getting Started with Splunk Security App, to get demo data and follow along with the scenarios.

alies that could indicate comm
ing these instructions with t
Splunk? Download the Gettin

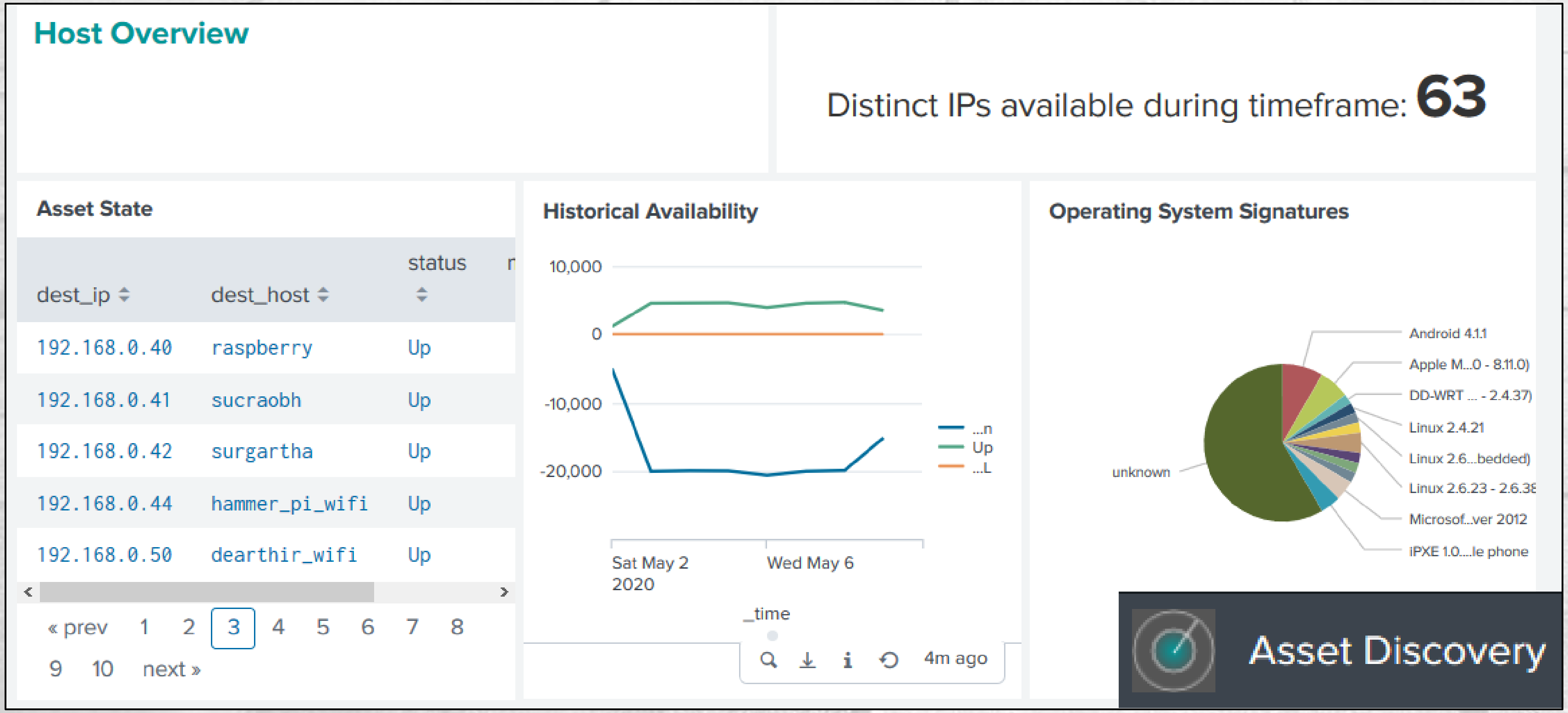
- Limited live instance on splunk.com
- Designed for familiarization
- No new data
  - Security data from 2016
  - 29 security focus sourcetypes
  - Multiple Data Models
- Searching & Reporting App
  - No admin capabilities

https://www.splunk.com/en_us/resources/videos -> Investigation: Command and Control -> “these instructions” link

Use case: Inventory

- MAC addresses from flows and arp command output
- IP address from flows and nmap
- Host names from dhcp logs and /etc/host files
- Grouping by type/purpose from OUI & static mappings
- Not all devices appear in all source types, and getting Type/Purpose is a challenge

MAC Address	Host Name	IP Address	Time	Type



Host Overview

Distinct IPs available during timeframe: **63**

Asset State

dest_ip	dest_host	status
192.168.0.40	raspberry	Up
192.168.0.41	sucraobh	Up
192.168.0.42	surgartha	Up
192.168.0.44	hammer_pi_wifi	Up
192.168.0.50	dearthir_wifi	Up

« prev 1 2 3 4 5 6 7 8 9 10 next »

Historical Availability

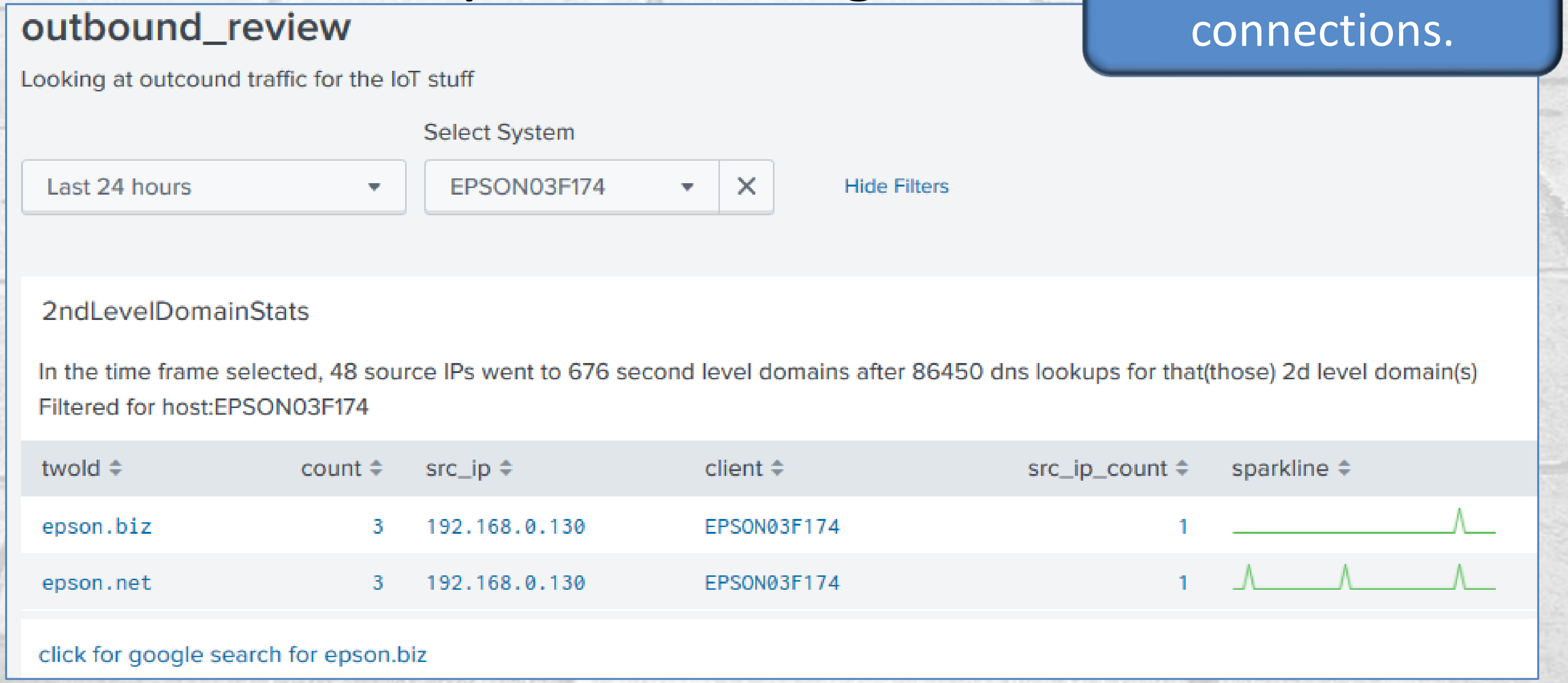
Operating System Signatures

Ping scan finds responding IP addresses
Netflow finds source MAC addresses
DHCP provides client names (7 Day Lease)
Static csv resolves static IPs and client "Type"
(IoT, Infrastructure, Personal Device, etc)
Bro_known_hosts

Use case: Baselining IoT

- Outbound DNS Count
  - CDNs make this a challenge
- Frequency Analysis
- Odd connections
- Flow data is key to baselining

New printer’s first connections.



outbound_review

Looking at outcound traffic for the IoT stuff

Select System

Last 24 hours ▾ | EPSON03F174 ▾ ✕

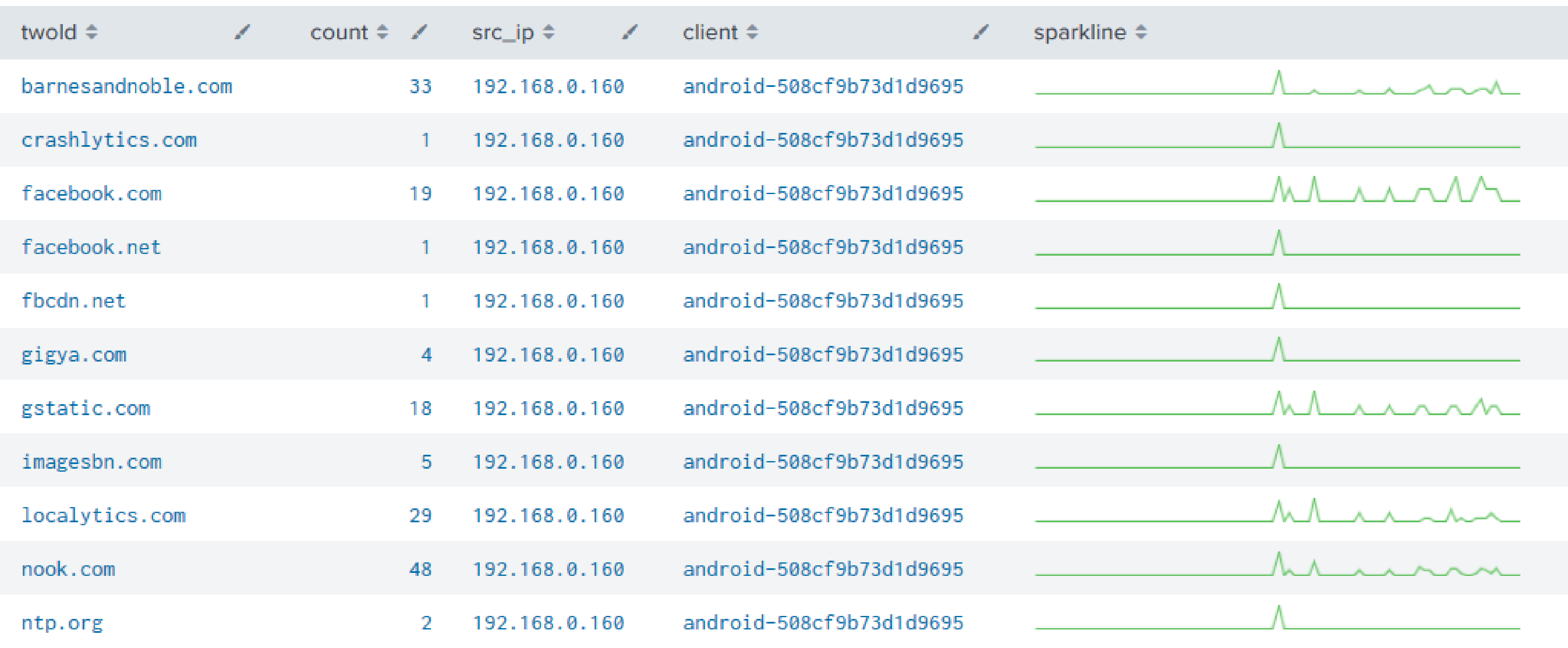
2ndLevelDomainStats

In the time frame selected, 48 source IPs went to 676 second level domains after 86450 dns lookups for that(those) 2d level domain(s)
Filtered for host:EPSON03F174

twold	count	src_ip	client	src_ip_count	sparkline
epson.biz	3	192.168.0.130	EPSON03F174	1	
epson.net	3	192.168.0.130	EPSON03F174	1	

click for google search for epson.biz

Use case: Baselining IoT New E-Reader

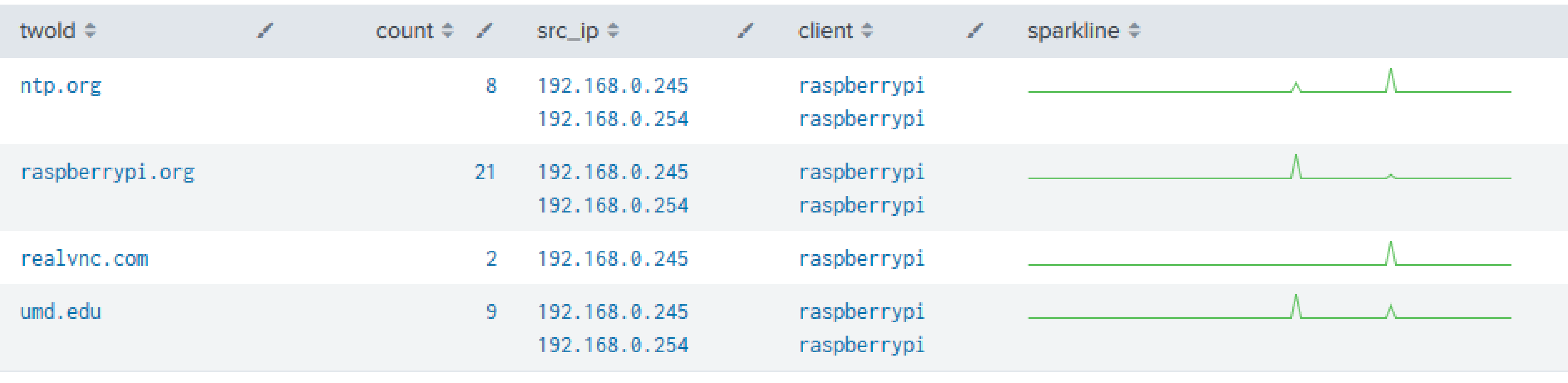


twold	count	src_ip	client	sparkline
barnesandnoble.com	33	192.168.0.160	android-508cf9b73d1d9695	
crashlytics.com	1	192.168.0.160	android-508cf9b73d1d9695	
facebook.com	19	192.168.0.160	android-508cf9b73d1d9695	
facebook.net	1	192.168.0.160	android-508cf9b73d1d9695	
fbcdn.net	1	192.168.0.160	android-508cf9b73d1d9695	
gigya.com	4	192.168.0.160	android-508cf9b73d1d9695	
gstatic.com	18	192.168.0.160	android-508cf9b73d1d9695	
imagesbn.com	5	192.168.0.160	android-508cf9b73d1d9695	
localytics.com	29	192.168.0.160	android-508cf9b73d1d9695	
nook.com	48	192.168.0.160	android-508cf9b73d1d9695	
ntp.org	2	192.168.0.160	android-508cf9b73d1d9695	

…ThAt's a LoT of FaCebOOk foR a GaDgeT w/o a BrOwSer!

Use case: Baselining IoT
New Raspberry PI 4

Now, that’s more like it…

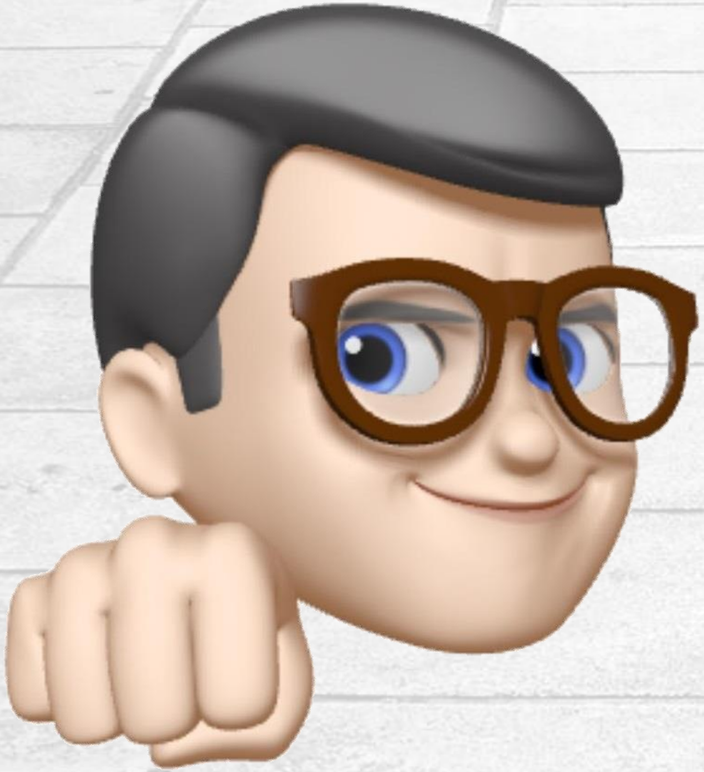


twold	count	src_ip	client	sparkline
ntp.org	8	192.168.0.245 192.168.0.254	raspberrypi raspberrypi	
raspberrypi.org	21	192.168.0.245 192.168.0.254	raspberrypi raspberrypi	
realvnc.com	2	192.168.0.245	raspberrypi	
umd.edu	9	192.168.0.245 192.168.0.254	raspberrypi raspberrypi	

buh-bye…

Questions
Comments
Snide Remarks

some after bits



Extra slides with scripts and details

nfdump scripts

```
harvest.sh:
#!/bin/sh# Quick script to harvest netflow from nfcapd files and get them ready for splunk ingest.
# the files are in the "/current" subdirectory
# three other dirs needed are /temp2move, /tempflows and /shannon
#
# This is called from crontab at a pace of your choosing.
#
# James Callahan - The Professional Paranoid
# written over time, but this version finalized 18 Feb 2015
#
# Clean up any left overs from last iteration
rm -f /netflow/processing/capfiles/nfcapd.2*
# I like to pause between steps, to ensure things get caught up and to reflect on the journey
/bin/sleep 5
# get the files you want to process. the .2* will have to be changed once we reach the year 3000
mv /netflow/current/nfcapd.2* /netflow/processing/capfiles
# set the date format
NOWDATE=`date +%Y%m%d-%H%M`
# set the format for the output files - check man nfdump for other options
/usr/bin/nfdump -o extended -b -q -R /netflow/processing/capfiles >
/netflow/processing/dumpfiles/netflow_$NOWDATE.flow
# clean up the output.  Top line gets rid of header and footer info.
/bin/sed -i 's/[\t ]/,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
# the files come in with many extra empty fields seperated by commas, these lines get rid of those.
/bin/sed -i 's/  /,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
/bin/sed -i 's/,,/,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
/bin/sed -i 's/,,/,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
/bin/sed -i 's/,,/,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
/bin/sed -i 's/,,/,/g' /netflow/processing/dumpfiles/netflow_$NOWDATE.flow
/bin/sleep 6
# Now move the output to the directory where splunk inputs.conf file is watching.
mv /netflow/processing/dumpfiles/*.flow /netflow/shannon/
```

```
removeOldFiles.sh:
#!/bin/bash
echo "Deleting files in /netflow/shannon older than 7 days"
find /netflow/shannon/* -mtime +7 -exec rm {} \;
```

```
crontab
# m h  dom mon dow   command
*/15 * * * * /netflow/src/harvest.sh
30 02 * * * /netflow/src/removeOldFiles.sh
```

PCAP -> Stream App

```
/usr/sbin/tcpdump -n -i eth1 -F dump.filter  -G 180
-w '/captures/dumps/rPi_%Y-%m-%d_%H:%M:%S.pcap'
```

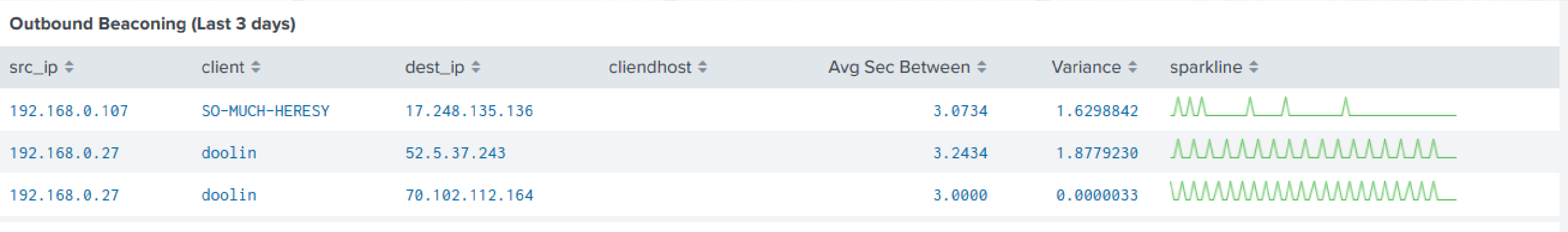
```
(/bin/find /dumps/rPi* -mmin +1
 -exec /opt/splunk/etc/apps/Splunk_TA_stream/linux_x86_64/bin/streamfwd
-r {} \; ) 2>&1 1>/dumps/docs/logs.txt
```

Used when you have collected PCAP externally and want to analyze it in Splunk as netflow

Above scripts designed to run as cron job for recurring ingest

Find Outbound Beaconing

```
sourcetype=stream:ip src_ip=192.168.0.1/16 dest_ip!=[my external ip]
        | streamstats current=f last(_time) as next_time by dest_ip
        | eval gap = next_time - _time |search gap>0 | eval gapm = gap/3600
        | stats count avg(gapm) AS asb var(gapm) AS vary sparkline by src_ip dest_ip
        |search count >3 asb>3 vary<2
        |eval "Avg Sec Between"=round(asb,4)
        |eval "Variance"=round(vary,7)
        |eval clientip=(dest_ip)
        |lookup dnslookup clientip
        |lookup static_macs.csv ip AS src_ip OUTPUT ip AS src_ip client AS client
        |table src_ip client dest_ip cliendhost "Avg Sec Between" "Variance" sparkline
```



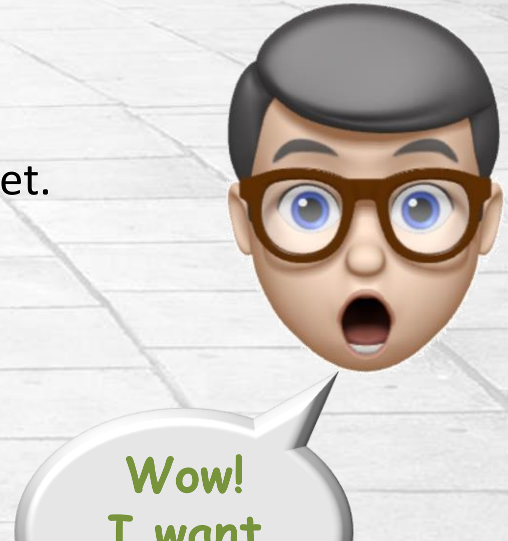
Outbound Beaconing (Last 3 days)

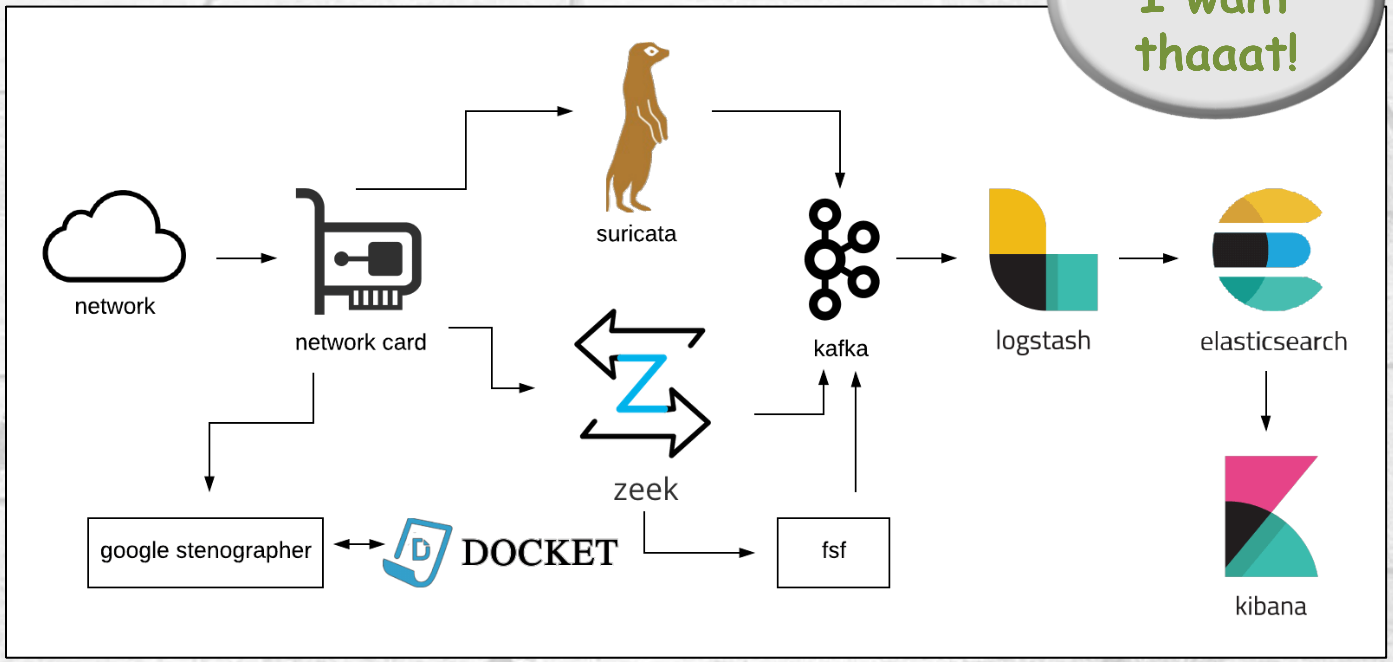
src_ip	client	dest_ip	cliendhost	Avg Sec Between	Variance	sparkline
192.168.0.107	SO-MUCH-HERESY	17.248.135.136		3.0734	1.6298842	
192.168.0.27	doolin	52.5.37.243		3.2434	1.8779230	
192.168.0.27	doolin	70.102.112.164		3.0000	0.0000033	

There are a lot of apps that phone home.

Rock NSM

- Full Packet Capture with Google’s Stenographer and Docket.
- Protocol Analysis and Metadata via Zeek.
- Signature Based Alerting via Suricata.
- Recursive File Scanning via FSF.
- Message Queuing and Distribution via Apache Kafka.
- ELK Stack





- Currently only using this for PCAP on a VM

http://rocknsm.io/

Inventory with ARP

Huh… arp sees more than nmap.

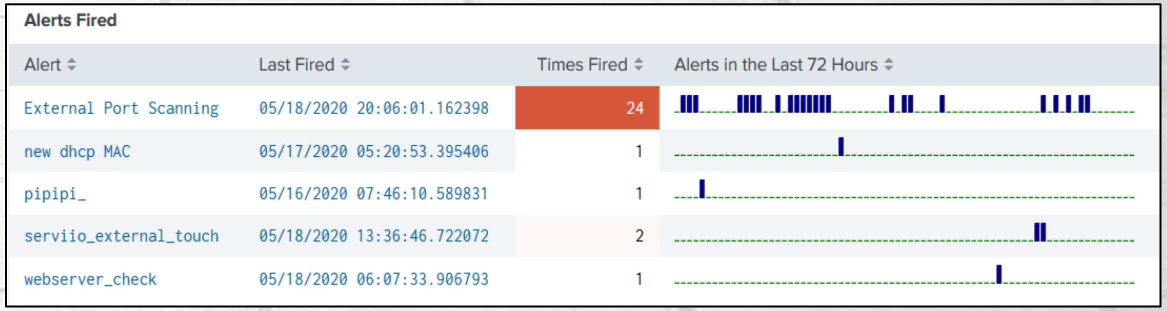
```
user@splunkbox ~]# nmap -sP 192.168.0.1-254 |grep MAC |cut -d " " -f 3 |sort |uniq |wc -l
48
user@splunkbox ~]# arp -a |cut -d " " -f 4 |sort |uniq |wc -l
80
```

Output directly into splunk – can use cron to run

```
/bin/cut -d' ' -f1,2,4 /root/arp.txt |  /bin/sed 's/[(),]//g; s/\ /\,/g;1i  client,ip,mac' >
/opt/splunk/etc/apps/my_app/lookups/inventory_doolin_arp.csv
```

Alerts Fired Panel

```
<row>
 <panel>
  <title>Alerts Fired</title>
  <table>
   <title>Alerts Fired</title>
   <search>
    <query>index=_audit action=alert_fired  |rename ss_name AS Alert
     |stats latest(_time) AS "Last Fired" count AS "Times Fired" sparkline AS "Alerts in the Last 72 Hours"
     first(sid) AS sid by Alert
     |convert ctime("Last Fired")</query>
    <earliest>-72h</earliest>
    <latest>now</latest>
    <refresh>90s</refresh>
   </search>
   <fields>Alert, "Last Fired", "Times Fired", "Alerts in the Last 72 Hours"</fields>
   <option name="wrap">true</option>
   <option name="rowNumbers">false</option>
   <option name="dataOverlayMode">heatmap</option>
   <option name="count">10</option>
   <option name="link.inspectSearch.visible">false</option>
   <option name="link.openSearch.visible">false</option>
   <format field="Alerts in the Last 72 Hours" type="sparkline">
    <option name="type">bar</option>
    <option name="barColor">green</option>
    <option name="colorMap">
     <option name="1:3">navy</option>
     <option name="3:7">orange</option>
     <option name="8:">red</option>
    </option>
   </format>
   <drilldown target="_blank">
    <link>search?sid=$row.sid$</link>
   </drilldown>
   <option name="drilldown">cell</option>
  </table>
  <html>
   <p>Location specific instructions in html
   </p>
  </html>
 </panel>
</row>
```



Alerts Fired

Alert ⇕	Last Fired ⇕	Times Fired ⇕	Alerts in the Last 72 Hours ⇕
External Port Scanning	05/18/2020 20:06:01.162398	24	
new dhcp MAC	05/17/2020 05:20:53.395406	1	
pipipi_	05/16/2020 07:46:10.589831	1	
serviio_external_touch	05/18/2020 13:36:46.722072	2	
webserver_check	05/18/2020 06:07:33.906793	1	

Also available from

go SPLUNK

gosplunk.com

Auto IP Block *Rube Goldberg Style*

```
sourcetype="firewall_log" tag::action="droppers"
| stats last(_time) as last_time first(_time) as first_time dc(dest_port) as dport_count dc(dest_ip) as dip_count
min(dest_port) AS Low_Port max(dest_port) AS High_Port
count as events by src_ip, dest_ip
| eval seconds=first_time-last_time
| eval minutes=(seconds/60)
| search dport_count > 3
| eval clientip=src_ip
| lookup dnslookup clientip
| table src_ip, clienthost, dest_ip, events, Low_Port High_Port dport_count, minutes
| sort -dport_count
|outputlookup drops_to_block.csv append=true
```

ips with >3 dropped ports

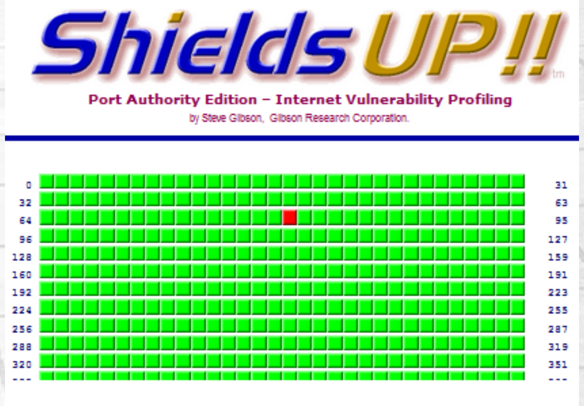
```
|inputlookup http_logs_to_drop.csv append=true
|append
[|search
index="websitelogs" http_response_code=40*
NOT [|search index="websitelogs" http_response_code=200 OR http_response_code=30* NOT uri
IN("/","robots.txt") |fields clientip]
|stats last(_time) as LastSeen dc(uri) as uric values(uri) as uriv values(http_response_code) as
http_response_code count by clientip
|eval uri=if(uric>3,uric+" uris",uriv)
| convert ctime(LastSeen)
|table clientip http_response_code uri LastSeen count]
 |eval rolloff1=relative_time(now(), "-4d")
 |eval dtger1=(strptime(LastSeen, "%m/%d/%Y %H:%M:%S"))
 |where dtger1>rolloff1
|dedup clientip
|table clientip http_response_code uri LastSeen count
|outputlookup http_logs_to_drop.csv
```

404's w/no 200's

```
|inputlookup drops_to_block.csv |fields src_ip
|append [
|inputlookup http_logs_to_drop.csv |fields clientip]
|eval ip=if(isnull(src_ip), clientip, src_ip)
|dedup ip
|table ip
|outputlookup ip_block_list.csv
```

merge lists

Rube Goldberg? www.rubegoldberg.com


Shields UP!!
Port Authority Edition – Internet Vulnerability Profiling
by Steve Gibson, Gibson Research Corporation.
0
32
64
96
128
160
192
224
256
288
320
31
63
95
127
159
191
223
255
287
319
351

www.grc.com

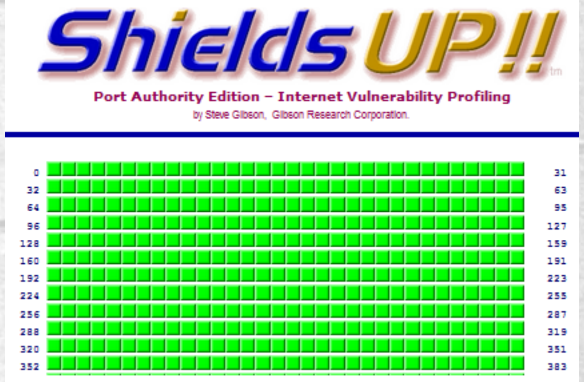
```
#bash
/usr/bin/sshpass -p 'password' /usr/bin/scp
benign@192.168.0.24:/splunk/etc/app
s/MalwareSpecial/lookups/ip_block_list.csv
/htdocs/website/firewall_iplist/ip_block_list.csv

/bin/sed  -e 1,1d -e 's/"//g'
/opt/lampp/htdocs/website/firewall_iplist/ip_block_list.csv >
/opt/lampp/htdocs/website/firewall_iplist/ip_block_list.txt

wc -l /opt/lampp/htdocs/website/firewall_iplist/ip_block_list.txt  >>
/var/log/syslog
```

pull to webserver then firewall pulls from there

Firewall reaches out to this URL for a block list on a cron */5 * * * *


Shields UP!!
Port Authority Edition – Internet Vulnerability Profiling
by Steve Gibson, Gibson Research Corporation.
0
32
64
96
128
160
192
224
256
288
320
352
31
63
95
127
159
191
223
255
287
319
351
383

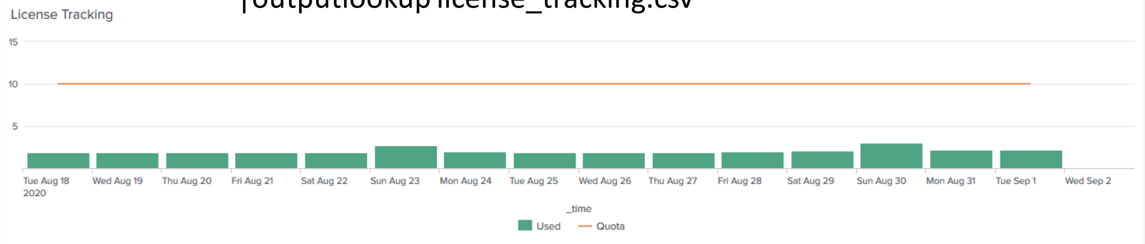
License Tracking

Workaround – the internal dash didn’t work.

Scheduled Search Cron: 59 22 * * *

```
|inputlookup license_tracking.csv append=true
|append
[| rest splunk_server=local /services/licenser/pools
| rename title AS Pool
| search
    [ rest splunk_server=local /services/licenser/groups
    | search is_active=1
    | eval stack_id=stack_ids
    | fields stack_id]
| eval quota=if(isnull(effective_quota),quota,effective_quota)
| eval "Used"=round(used_bytes/1024/1024/1024, 3)
| eval "Quota"=round(quota/1024/1024/1024, 3)
| eval "% used"=round(used_bytes/quota*100,2)
| fields Pool "Used" "% used" "Quota"
 |eval dtger=(now())
 |eval dtgr=strftime(dtger, "%Y-%m-%d %H:%M:%S")
]
|table dtger dtgr Pool "Used" "% used" "Quota"
|outputlookup license_tracking.csv
```

```
<row>
  <panel>
    <title>License Tracking</title>
    <chart>
      <search>
        <query>|inputlookup  license_tracking.csv
| eval dtgrepoch = strptime(dtgr, "%Y-%m-%d %H:%M:%S")
|eval _time=(dtgrepoch)
 |eval rolloff1=relative_time(now(), "-15d@d")
 |where dtgrepoch&gt;rolloff1
|timechart sum(Used) max(Quota) span=1d
|rename sum(Used) AS Used, max(Quota) as Quota</query>
        <earliest>-24h@h</earliest>
        <latest>now</latest>
        <sampleRatio>1</sampleRatio>
      </search>
      <option name="charting.axisLabelsX.majorLabelStyle.overflowMode">ellipsisNone</option>
      <option name="charting.axisLabelsX.majorLabelStyle.rotation">0</option>
      <option name="charting.axisTitleX.visibility">visible</option>
      <option name="charting.axisTitleY.visibility">visible</option>
      <option name="charting.axisTitleY2.visibility">visible</option>
      <option name="charting.axisX.abbreviation">none</option>
      <option name="charting.axisX.scale">linear</option>
      <option name="charting.axisY.abbreviation">auto</option>
      <option name="charting.axisY.scale">linear</option>
      <option name="charting.axisY2.abbreviation">none</option>
      <option name="charting.axisY2.enabled">0</option>
      <option name="charting.axisY2.scale">inherit</option>
      <option name="charting.chart">column</option>
      <option name="charting.chart.bubbleMaximumSize">50</option>
      <option name="charting.chart.bubbleMinimumSize">10</option>
      <option name="charting.chart.bubbleSizeBy">area</option>
      <option name="charting.chart.nullValueMode">gaps</option>
      <option name="charting.chart.overlayFields">Quota</option>
      <option name="charting.chart.showDataLabels">none</option>
      <option name="charting.chart.sliceCollapsingThreshold">0.01</option>
      <option name="charting.chart.stackMode">default</option>
      <option name="charting.chart.style">shiny</option>
      <option name="charting.drilldown">none</option>
      <option name="charting.layout.splitSeries">0</option>
      <option name="charting.layout.splitSeries.allowIndependentYRanges">0</option>
      <option name="charting.legend.labelStyle.overflowMode">ellipsisMiddle</option>
      <option name="charting.legend.mode">standard</option>
      <option name="charting.legend.placement">bottom</option>
      <option name="charting.lineWidth">2</option>
      <option name="trellis.enabled">0</option>
      <option name="trellis.scales.shared">1</option>
      <option name="trellis.size">medium</option>
    </chart>
  </panel>
</row>
```



Suricata Details Chart

```
index="suricata" | fillnull value="nope" event_type
|eval event_type=case(
source="/opt/suricata/etc/suricata/rules/oinkmater_update.info","Oinkmaster_update",
source="/opt/suricata/etc/suricata/rules/sid-msg.map","sid-msg.map",
source="/var/log/suricata/fast.log","fast.log",
source="/var/log/suricata/http.log","http.log",
source="/var/log/suricata/stats.log", "stats.log",1=1,event_type)
| timechart count by event_type useother=false limit=0
```

Turning on netflow

```
/opt/suricata/etc/suricata/suricata.yaml
~
types:
~
      # bi-directional flows
      #- flow
      # uni-directional flows
      - netflow
```