Using Netflow & Open Source Tools for Network Behavioral Analysis

Yves B. Desharnais, MBA, CISSP, PCIP

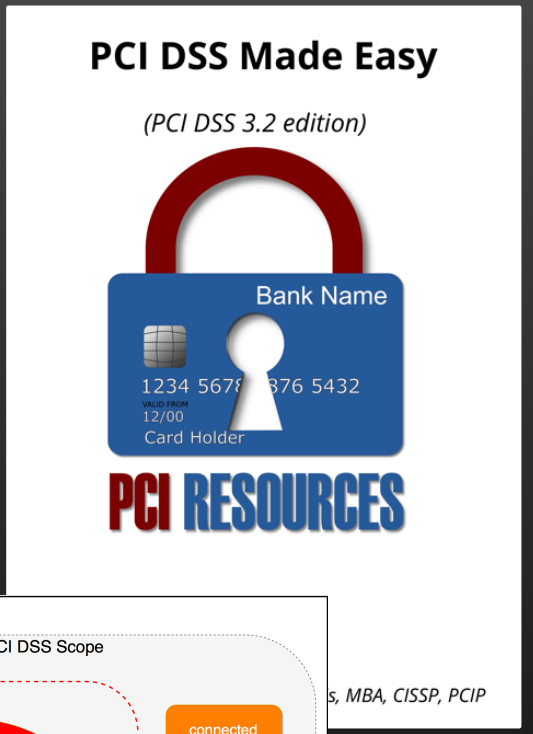
BSides Calgary 2017 – October 19, 2017

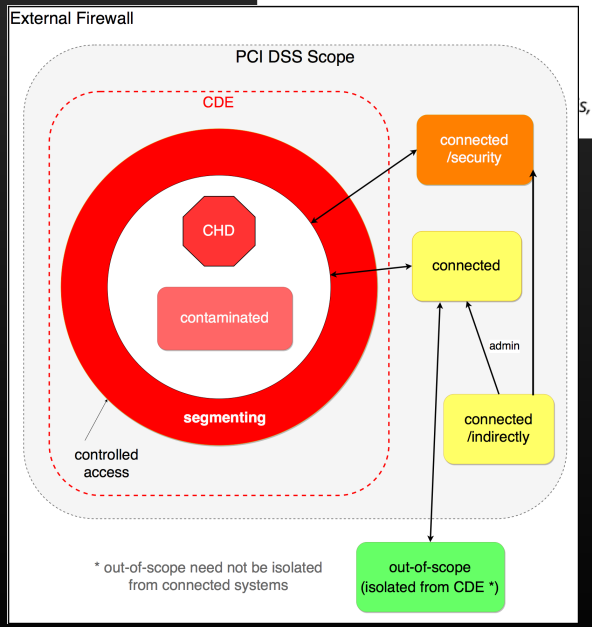
Disclaimer

- This presentation is the result of my experience and only represents my understanding, and is not endorsed by anyone other than myself
- YMMV – I had to do a lot of “printf” debugging to get things working
- I do not guarantee I will fix any issue in the code (I’m a Ruby newbie)

About Yves

- IT/InfoSec Expert generalist.
- Over 16 years professional experience (IT/Infosec)
- Background in software development, Unix/Linux administration
- Author of books on PCI DSS including
  - PCI DSS Scope methodology and approach (one of 2 public methodologies at the time) CC-BY-SA





About this talk

- Story about necessity being the mother of inventions
- Could have been called: Extreme PCI DSS Scope Reduction or how we remediated a medium size flat-network in 19 weeks
- But the approach and tools developed have much broader use
- It started with a LinkedIn message …

Project Overview

From interview: Status

- 2nd year PCI DSS compliance extension (< 5 months left) – acquired company network
- flat network, mostly VMs, no patching/hardening
- few diagrams used to prepare system migrations to the new network (i.e. incomplete documentation)
- card number tokenization in use
- very competent IT staff

How to achieve compliance?

- This called for massive reduction in scope if they were to have any chance of achieving compliance
- With limited information on applications, we needed to identify what talked on the network to whom and on what protocols…
- first thought: span port and packet capture, but speaking to network friend I learned of the NetFlow protocol

Start of contract
High-level plan after 24h

- 6 major steps
  1. Scope identification and reduction
  2. Vulnerability Management / Penetration Testing
  3. RBAC Controls
  4. Remediate systems (patching hardening)
  5. Logging and Monitoring (of the CDE)
  6. Miscellaneous
     - Policies and Procedures
     - Review business Processes
     - Security Awareness Training
- Other than step 6, all depend on step 1
- Other steps are subject for another talk

NetFlow version overview

	5	9	10 (IPFix)
Owner of Standard	Cisco	Cisco	IETF
RFC		3954 (©2004)	7011 (©2013)
Format	Fixed	Template-based	
IP version	IPv4	Any (v4, v6) & Ethernet	
Equipment Supporting	Older Cisco switches/routers	Newer Cisco, some other network providers	Most other providers (including VMWare)

** Sflow is not NetFlow, it is sampled and not useful for our purposes*

How to capture and analyze NetFlow?

- Market research - little time, and little budget
- Many tools supported NetFlow (including NTOP, SiLK) – none met my needs for scoping
- Roll-your-own with either Logstash or Fluentd - both ruby based
- Logstash (famously the L in the ELK stack) – Ruby/J (requires JDK, more portable)
- Fluentd is more decentralized – Ruby/C (uses less RAM)
- Both are basically data aggregator/transformations engines using plugins, aka the *nix command line on steroids but using JSON
- I could describe further…. But I'd rather show you

(Controlled) Demos

- Demos are controlled by using saved PCAP files (tcpdump, tcpreplay) – can allow comparison when performing changes
- Demo slides are captured from what is shown in the talk

- Instructions, code, and this presentation will be available online after the talk

Demo #1

- Standard NetFlow Plugin (fluent-plugin-netflow)
- PCAP: 9 packets - v5 and v9 (v5.pcap)
- Output is JSON (JavaScript Object Notation)


```
2016-08-23T09:27:42-04:00      netflow.event   {"version":5,"uptime":1743829144
,"flow_records":30,"flow_seq_num":2913226023,"engine_type":0,"engine_id":0,"samp
ling_algorithm":0,"sampling_interval":0,"ipv4_src_addr":"10.4.65.172","ipv4_dst_
addr":"10.1.4.151","ipv4_next_hop":"10.1.0.3","input_snmp":14,"output_snmp":2,"i
n_pkts":1,"in_bytes":40,"first_switched":1743825392,"last_switched":1743825392,"
l4_src_port":57599,"l4_dst_port":80,"tcp_flags":20,"protocol":6,"src_tos":0,"src
_as":0,"dst_as":0,"src_mask":24,"dst_mask":24,"host":"192.168.2.32"}
```


- Warning in td-agent.log:
  - No matching template for host="192.168.2.32" source_id=0 flowset_id=265
  - This is because v9 are template-based and we haven't received templates yet (time between templates being sent is a parameter from network devices)

```
[warn]: No matching template for host="192.168.2.32" source_id=0 flowset_id=265
[warn]: No matching template for host="192.168.2.32" source_id=0 flowset_id=256
```

Demo #2

- Standard NetFlow Plugin (fluent-plugin-netflow)
- PCAP: same 9 packets - v5 and v9 (v5.pcap)
- Output is CSV
- Use initially with: “sort | uniq | grep” -> transfer to PC, filter with Excel
- Data for v5 only (same ”no matching template” v9 issue)

```
"5","192.168.2.32","10.1.8.61","10.1.43.12","389","53171","6"
"5","192.168.2.32","10.4.65.164","10.1.1.15","55855","80","6"
"5","192.168.2.32","10.4.65.164","10.1.1.15","55855","80","6"
"5","192.168.2.32","10.1.3.241","10.1.34.130","1035","161","17"
"5","192.168.2.32","10.1.34.130","10.1.3.241","161","1035","17"
"5","192.168.2.32","10.1.34.130","10.1.3.241","161","1035","17"
/var/log/td-agent/demo2/netflow-csv.20160823.b55b8c0048629db3c
```

Demo #3

- Standard NetFlow Plugin (fluent-plugin-netflow)
- PCAP: new files v9 (v9cleanly.pcap) and v10 (v10.pcap) from VMWare ESXi
- Output is CSV
- There were v9 bugs that I fixed (printf!) and submitted (without really understanding why…)
- New v9 issue in plugin:

```
2017-10-14 21:58:12 -0400 [warn]: Skip unsupported field type=33001 length=12
2017-10-14 21:58:12 -0400 [warn]: Skip unsupported field type=40000 length=65
```

- “Skip unsupported field” - Fields in template are provider defined (and can be added manually)
- And Version 10 is not supported!

```
[warn]: Unsupported Netflow version v10: Fixnum
```

Demo #4

- New (reworked) NetFlow/IPFix Plugin (netflowipfix_input) – support for v5, v9, v10 (IPFix)
- PCAP: 1000 v10 packets (v10-1000.pcap)
- Missing information for some fields in CSV

```
"10","192.168.2.32","","","","","",""
"10","192.168.2.32","","","","","",""
```

- But JSON includes the full data … different names in different versions

```
2016-08-23T14:37:33-04:00       netflow.event   {"version":"10","flow_seq_num
":"4063237","flowset_id":"256","sourceIPv4Address":"10.195.64.31","destinatio
nIPv4Address":"10.195.64.130","octetDeltaCount":"992","packetDeltaCount":"1",
"flowStartMilliseconds":"1471977439000","flowEndMilliseconds":"1471977439000"
,"sourceTransportPort":"3268","destinationTransportPort":"48940","ingressInte
rface":"1881","egressInterface":"1794","layer2SegmentId":"0","protocolIdentif
ier":"6","flowEndReason":"1","tcpControlBits":"6146","ipClassOfService":"128"
,"maximumTTL":"1","flowDirection":"0","paddingOctets":"[]","host":"192.168.2.
32"}
```

Demo #5

- New (reworked) NetFlow/IPFix Plugin (netflowipfix_input)
- PCAP: multiple (v5.pcap, v10-1000.pcap, v9cleanly.pcap)
- Workflow using Fluentd plugins () => complex to create and managed
- Output to multiple files, but could recombine using longer workflows

```
[root@netflowpres td-agent]# ls /var/log/td-agent/demo5/
netflow-v10-csv.20160823.log.gz   netflow-v5-csv.20160823.log.gz
netflow-v10-json.20160823.log.gz  netflow-v5-json.20160823.log.gz
```

```
"10","192.168.2.32","10.195.64.165","10.195.64.16","6969","63144","TCP"
/var/log/td-agent/demo5/netflow-v10-csv.20160823.log.gz
```

```
"5","192.168.2.32","10.64.1.8","10.64.1.2","123","123","UDP"
/var/log/td-agent/demo5/netflow-v5-csv.20160823.log.gz (END)
```

Demo #6

- New (reworked) NetFlow/IPFix Plugin (netflowipfix_input)
- PCAP: multiple (v5.pcap, v10-1000.pcap, v9cleanly.pcap)
- New plugin to normalize traffic for my use to replace workflow, allows for simpler config file
- CSV output, still using Excel for analysis

```
"10","192.168.2.32","10.42.54.30","10.195.64.57","161","55591","17","UDP"
"10","192.168.2.32","10.195.64.121","10.195.64.104","443","62575","6","TCP"
"10","192.168.2.32","10.195.64.121","10.195.64.104","443","62575","6","TCP"
"10","192.168.2.32","10.42.140.100","10.195.64.70","0","0","1","ICMP"
```

Demo #7

- New (reworked) NetFlow/IPFix Plugin (netflowipfix_input)
- PCAP: 1000 v10 packets (v10-1000.pcap)
- New plugin to normalize traffic for my use to replace workflow
- Matching ACLs (CSV format) – IDS or Firewalls rules validation functionality
- CSV output, still using Excel for analysis

```
"10","192.168.2.32","10.195.64.4","224.0.0.2","1985","1985","17","UDP","nomatch"
"10","192.168.2.32","10.195.64.33","10.195.64.62","88","15598","6","TCP","KERBEROS"
"10","192.168.2.32","10.195.64.33","10.195.64.62","88","15598","6","TCP","KERBEROS"
"10","192.168.2.32","10.195.64.39","224.0.0.252","59920","5355","17","UDP","HOSTMON"
"10","192.168.2.32","10.195.64.33","10.195.64.62","88","15598","6","TCP","KERBEROS"
```

Demo #8

- New (reworked) NetFlow/IPFix Plugin (netflowipfix_input)
- PCAP: 1000 v10 packets (v10-1000.pcap)
- New plugin to normalize traffic for my use to replace workflow
- Output to MySQL for (easier) analysis

```
MariaDB [demo8]> select * from nf limit 10;
+-----+---------------------+--------------+--------------+--------------+---------+----
-----+----------+-------+---------+--------------+---------+------------+
| nid | added               | reportinghost | srcip        | dstip        | srcport | dst
port | protocol | nfver | dstzone | acldirection | aclno   | aclentryid |
+-----+---------------------+--------------+--------------+--------------+---------+----
-----+----------+-------+---------+--------------+---------+------------+
|   1 | 2017-10-16 05:48:30 | 192.168.2.32  | 10.195.64.130 | 10.195.64.61 | 6391    | 425
57   | TCP      | 10    | NULL    | NULL         | nomatch | NULL       |
|   2 | 2017-10-16 05:48:31 | 192.168.2.32  | 10.195.64.33  | 10.195.64.61 | 389     | 415
61   | TCP      | 10    | NULL    | NULL         | nomatch | NULL       |
```

Demo #9

- Full final system – with other plugins
  - How do we know there are new systems on the network?
    - New IP in NetFlow and PING sweeps (nmap cron job)
  - How do we know directionality of flows (dynamic ports and services using same port e.g. tcp/123)
    - Add new IP to a list (new plugin)
    - Have a periodic process perform port scans
  - We can split traffic into
    - Known services Flows
    - Unknown services Flows
    - Daily and overall
- Using MySQL, nmap (pings, port scans)

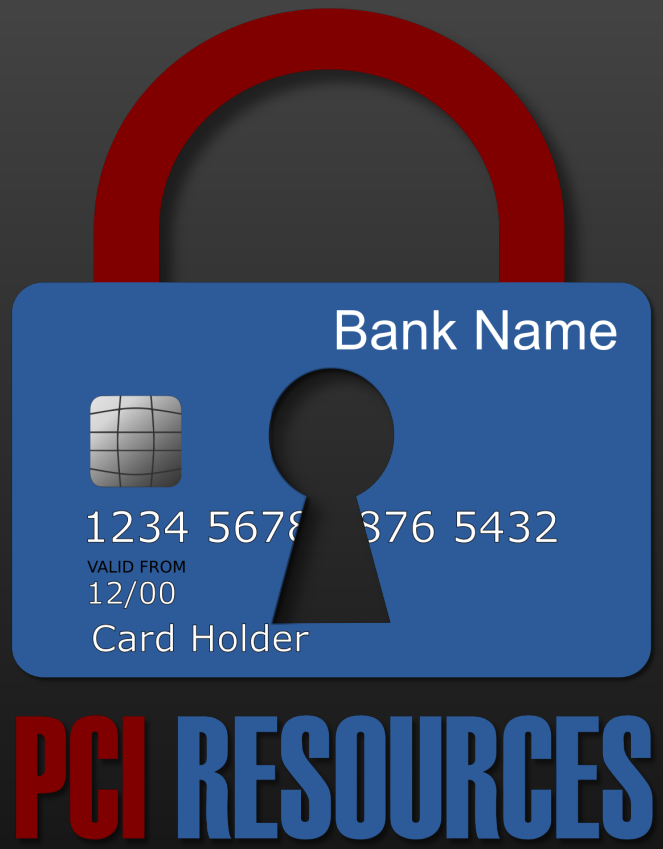
```
MariaDB [assets]> show tables;
+------------------+
| Tables_in_assets |
+------------------+
| dailyflow        |
| flow             |
| ipv4             |
| otherDailyFlow   |
| otherFlow        |
| service          |
| unknownDailyFlow |
| unknownFlow      |
+------------------+
8 rows in set (0.00 sec)
```

Solution Issues / Parting Thoughts

- Netflow
  - is UDP, so risk of packet loss – keep collectors close to source (and distribute processed information)
  - Multiple devices could provide duplicate traffic information
  - Different devices provide different templates, offer different possibilities
- Fluentd
  - Can be setup in hierarchy for distributed sites
  - has other uses: Logging and Monitoring (using nxlog)
  - Can be combined with other tools (elasticsearch, etc.)
- Solution
  - May require more manual edits and tweaks
  - New plugins are not packaged using gems (Ruby packaging system)

References

- Everything is published on www.PCIresources.com
  - Blog Post including links to
    - Instructions / Source code
    - This presentation
  - PCI Resources Scoping Model and Approach



- Blog post URL:

  www.PCIresources.com/blog/bsides-calgary-2017-netflow