

Netflow in Daily Information Security Operations

Mike Pochan

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Software Engineering Institute

Carnegie Mellon University

© 2016 Carnegie Mellon University

Distribution Statement A: Approved for Public Release;
Distribution is Unlimited

Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon®, CERT® and FloCon® are registered marks of Carnegie Mellon University.

DM-0003173



Agenda

- Types of netflow tools being used
- Sensor architecture
- Sensor and endpoint configurations
- Use cases
 - Malicious domain lookup detection
 - Beacon detection
 - Outbound SSH anomalies
 - Augmenting IDS coverage with pDNS



Why use netflow tools?

- Free
- Lightweight in terms of:
 - Processing, since it's not dealing with whole data streams
 - Storage. 3T can store up to a year's worth of flow data
 - Analysis. Queries run extremely quickly
- Great for strengthening existing security posture



Two types of toolsets

1. Collection and metering tools
 1. YAF (Yet Another Flowmeter) – flow collector
 2. Super_mediator – flow importer/exporter
2. Analysis Tools
 1. SiLK – flow data repository
 2. Orcus – passive DNS database
 3. Analysis Pipeline (AP) – real-time alerting on flows

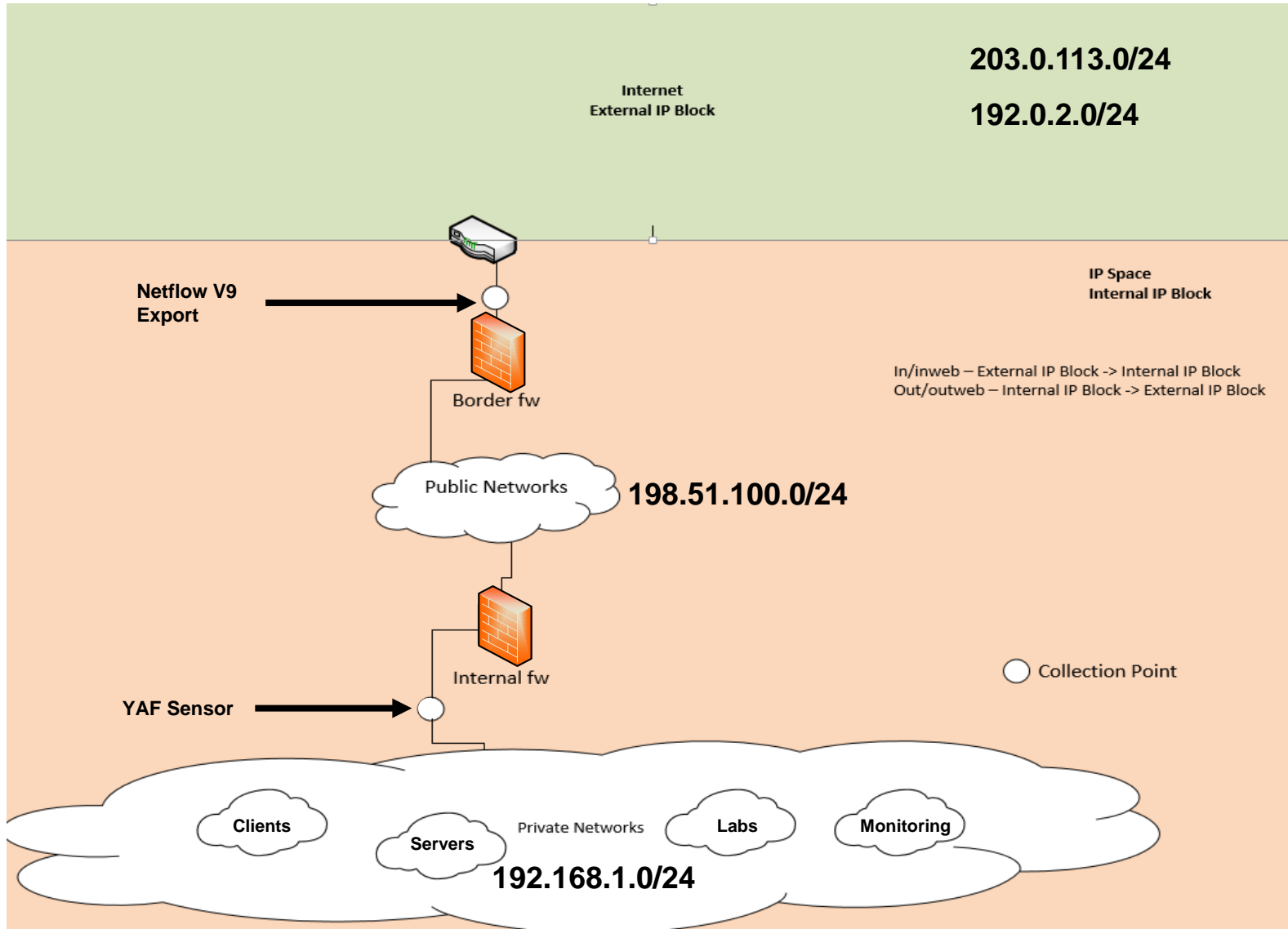


Analysis Tools – Common Processes/Commands

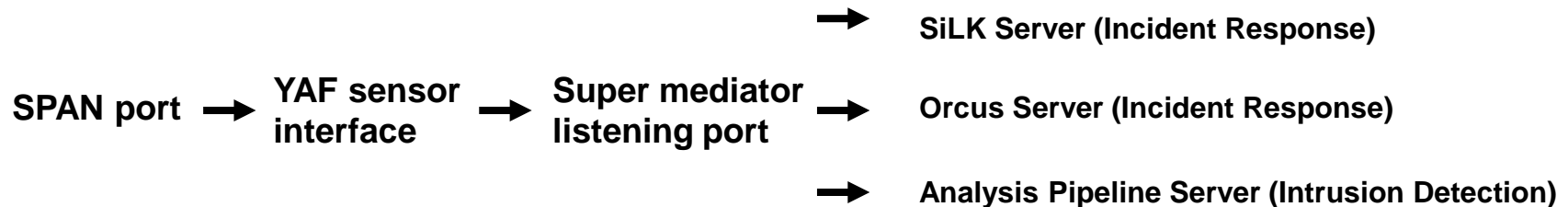
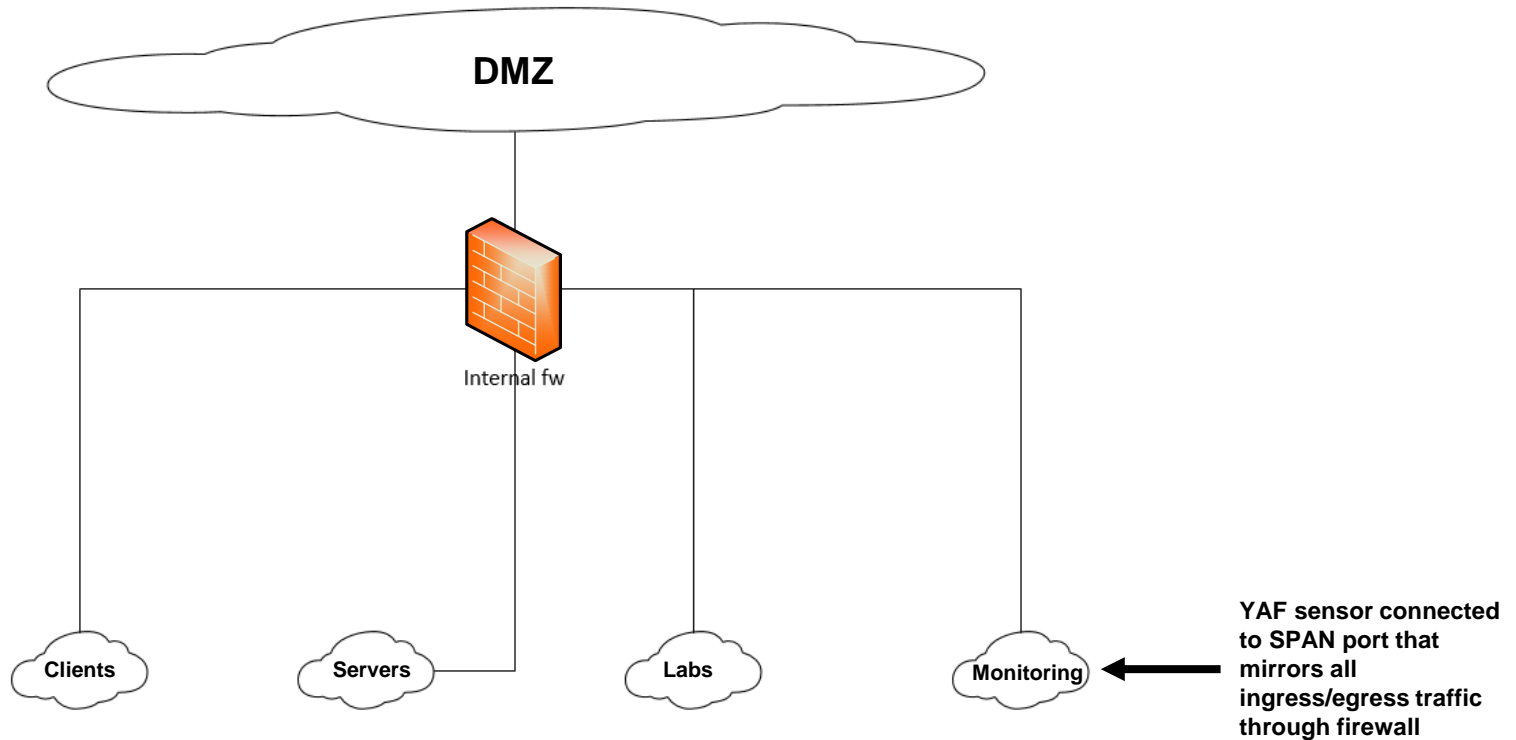
- SiLK
 - Rwflowpack – collection process
 - Rwfilter – primary query command
- Orcus
 - Orlookup – query to map between IPs and domain names
 - Orquery – query to access DNS records from database
- Analysis Pipeline
 - Filter – similar to rwfilter, but preconfigured
 - Evaluations – series of checks that are performed on flows that pass the filters



Sensor Architecture



YAF flow distribution (internal network)



SPAN to YAF to Super Mediator

```
/usr/bin/yaf --silk --in=p1p1 --live=pcap --ipfix=tcp --out=127.0.0.1 --ipfix-  
port=18004 --become-user tcpdump --become-group tcpdump --mac --plugin-  
name=/usr/lib64/yaf/dpacketplugin.la --applabel --applabel-  
rules=/etc/yafApplabelRules.conf --plugin-conf=/etc/yafDPIRules.conf --max-  
payload=5000 --udp-uniflow=53 --verbose --log=/var/log/messages --plugin-opts 53
```

--in=p1p1 – YAF server interface connected to SPAN sport

--ipfix-port=18004 – listening Super Mediator port on same host

--plugin-opts 53 – DPI on DNS data (important for later)



SM to Analysis Endpoints (SiLK)

Super Mediator Server

Collect from YAF

COLLECTOR TCP

HOST "127.0.0.1"

PORT 18004

COLLECTOR END



Export to SiLK server

EXPORTER TCP

HOST "silk.server.ip"

PORT 9934

FLOW_ONLY

EXPORTER END

SiLK Server (rwflowpack)

Collect flow data from SM

probe Internalfw ipfix

listen-on-port 9934

protocol tcp

end probe

sensor Internalfw0

ipfix-probes Internalfw

internal-ipblock

@internal-networks

external-ipblock

remainder

end sensor



SM to Analysis Endpoints (Orcus)

Super Mediator Server

Export to Orcus Server

EXPORTER TCP

PORT 18009

HOST "orcus.host.ip" →

APPLICATION == 53

DPI_ONLY

EXPORTER END

Orcus Server (SM again)

Collect DPI DNS from SM

COLLECTOR TCP

PORT 18009

COLLECTOR END

EXPORTER FILEHANDLER

PATH "/var/orcus/fw0"

ROTATE 300

LOCK

EXPORTER END



SM to Analysis Endpoints (Analysis Pipeline)

Super Mediator Server

Analysis Pipeline

Export to AP

EXPORTER TCP

PORT 9970

HOST "AP.host.ip" →

EXPORTER END

#Collect flow and DPI DNS from SM

PRIMARY DATA SOURCE flow_dpi_data

YAF BUILDER

TCP PORT 9970

BREAK ON RECS 5000

TIMING FIELD NAME flowEndMilliseconds

END DATA SOURCE



What about the V9 border flows?

- Only sent to SiLK and Analysis Pipeline since DPI is not an option.

Silk Server (rwflowpack)

```
probe Border netflow-v9
    listen-on-port 9920
    protocol udp
end probe
sensor Border0
    netflow-v9-probes Border
    internal-ipblock @internal-networks
    external-ipblock remainder
end sensor
```

Analysis Pipeline Server

```
SECONDARY DATA SOURCE silk
    SILK BUILDER
    INCOMING DIRECTORY "/AP/incoming"
    ERROR DIRECTORY "/AP/error"
END DATA SOURCE
```



Analysis Pipeline

What traffic do we detect with it?

- Malicious domain lookups on internal resolvers
- Beaconsing
- Traffic to/from IP blacklists
- Lateral movement
- Anomalous outbound ssh/rdp traffic
- Traffic to/from foreign nations



Analysis Pipeline – Malicious Domain Queries

- Need a list of malicious domains to start out with
 - ~35,000 unique ones
- File name - pipeline_domain_blacklist.txt
 - Format

```
##format:dns  
baddomain.com  
notaRAT.com  
givemePII.net  
asdlkfjsadfsad.org  
qowenzie.com
```



Analysis Pipeline – Malicious Domain Queries

```
FILTER bad_domains
    destinationTransportPort==53
    sourceIPv4Address IN_LIST "/etc/lookup_list.set"
    sourceIPv4Address NOT_IN_LIST "/etc/mx_list.set"
    destinationIPv4Address IN_LIST "/etc/lookup_list_dest.set"
    dnsQName IN_LIST "/etc/pipeline_domain_blacklist.txt"
END FILTER
```

Translation

- Destination port is 53
- Lookup source IP is in our home network
- Lookup source IP is not one of our MX servers (noisy)
- Lookup destination is one of our internal resolvers
- Domain in the query is in our malicious domain file



Analysis Pipeline – Malicious Domain Queries

```
EVALUATION malicious_domain_lookup
  FILTER bad_domains
  CHECK EVERYTHING PASSES
  END CHECK
  ALERT ALWAYS
  ALERT EVERYTHING
  EXTRA ALERT FIELD dnsQName
END EVALUATION
```

Domain Looked up

Source IP (client)

Destination IP
(internal resolver
address)

From alert.log

2015-10-30

14:03:19|Evaluation|malicious_domain_lookup|1|2015-10-30

14:03:19|2015-10-30

14:03:19|62|1|192.168.1.22|192.168.1.7|57112|53|0|17|31|53|
107|0|0|d9d40f7f|www.i-am-bad.com. |



Analysis Pipeline – Malicious Domain Queries

- Utilize Splunk to send out real-time email alerts

Subject: Splunk Alert: Malicious Domain Lookup

The following malicious domain was looked up by the listed host. This activity should be investigated.

Alert: [Malicious Domain Lookup](#)

[View results in Splunk](#)

Pipeline_Domain	Pipeline_Source_IP	Pipeline_Time_UTC	host
www.i-am-bad.com.]	192.168.1.22	2015-10-30 14:03:19	Client.hostname.edu

If you believe you've received this email in error, please see your Splunk administrator.

splunk > the engine for machine data



Orcus – Malicious Domain Lookup Pivot

- What does the domain resolve to?

```
$ orlookup --start-date=2015/10/29 --end-date=2015/10/31 --  
name=com.i-am-bad.www
```

```
date|name|address|source
```

```
2015-10-29|com.i-am-bad.www|203.0.113.200|A
```

```
2015-10-30|com.i-am-bad.www|203.0.113.55|A
```

```
2015-10-31|com.i-am-bad.www|203.0.113.200|A
```

- Now we know the IP this domain resolves to on the day of the alert
 - Use SiLK to find source IPs
 - What type of traffic do we see to this IP?



SiLK – Malicious Domain Lookup Pivot

```
$ rwfilter --type=out,outweb --start-date=2015/10/30  
--end-date=2015/10/30 --daddress=203.0.113.55 --  
pass=stdout | rwstats --fields=sIP,dPort --packets --  
top --count=10
```

sIP	dPort	Packets	%Packets	cumul_%
198.51.100.101	80	225	60.483871	60.483871
198.51.100.105	25	147	39.516129	100.000000

Web proxy IP – search proxy logs for client IP (hopefully matches our AP alert's source IP). Proxy logs and full pcap will show if anything malicious was downloaded. Also can look for redirects to other sites based on time stamps.

MX server IP – most likely harmless



Analysis Pipeline – Beacon Detection

FILTER beacon

```
sourceIPv4Address NOT_IN_LIST "/etc/dns.set"  
sourceIPv4Address IN_LIST "/etc/internal.set"  
sourceTransportPort >= 1024  
destinationIPv4Address NOT_IN_LIST "/etc/whitelist.set"  
destinationTransportPort NOT_IN_LIST [25,1935,993,5223,5222,161,119,587,110,53]
```

END FILTER

- **EVERYTHING** beacons.
- Tune by:
 - Source Address
 - Destination Address
 - Destination Port
 - This takes time
 - DNS and SMTP should be whitelisted from the beginning



Analysis Pipeline – Beacon Detection

```
EVALUATION beacon_eval
  FILTER beacon
  CHECK BEACON
    COUNT 20 CHECK_TOLERANCE 5 PERCENT
    TIME_WINDOW 5 MINUTES
  END CHECK
  CLEAR NEVER
  SEVERITY 3
  OUTPUT TIMEOUT 1 DAY
  ALERT EACH_ONLY_ONCE
  ALERT 2 TIMES 1 HOURS
END EVALUATION
```

- At least 20 beacons with a minimum 5 minute intervals
- 5% error for the intervals



Analysis Pipeline – Beacon Detection

2015-10-28

19:43:13|Evaluation|beacon_eval|3|sourceIPv4Address,destinationIPv4Address,destinationTransportPort,protocolIdentifier|198.51.100.12,192.0.2.43,80,6|BEACON|20,330|

- Source Address
- Destination Address
- Destination Port
- Beacon Interval (in seconds)



Analysis Pipeline – Beacon Detection

Subject: Splunk Alert: Beacon Traffic Detected

Beacon traffic to the following external IP was detected from the listed host. This should be investigated.

Alert: [Beacon Traffic Detected](#)

[View results in Splunk](#)

Src_IP	Dst_IP	Dst_Port	Time	host	_time
198.51.100.12	192.0.2.43	80	2015-10-28 19:43:13	Client.hostname.edu	Wed Oct 28 19:43:13 2015

If you believe you've received this email in error, please see your Splunk administrator.

splunk > the engine for machine data



Orcus – Beacon Pivot

- What does this IP resolve to?

```
$ orlookup --start-date=2015/10/27 --end-date=2015/10/29 --  
address=192.0.2.43
```

```
date|name|address|source
```

```
2015-10-27|org.fedoraproject.mail|192.0.2.43|A
```

```
2015-10-28|org.fedoraproject|192.0.2.43|A
```

```
2015-10-29|org.fedoraproject|192.0.2.43|A
```

- False positive
 - Add address to “/etc/whitelist.set”



Analysis Pipeline – Outbound SSH Anomalies

```
FILTER outbound_SSH
```

```
    sourceIPv4Address IN_LIST "/etc/home.set"
```

```
    destinationIPv4Address NOT_IN_LIST "/etc/home.set"
```

```
    destinationIPv4Address NOT_IN_LIST "/etc/ssh_whitelist.set"
```

```
    destinationTransportPort==22
```

```
END FILTER
```

Translation

- SSH traffic from our network to external IPs
- External IPs are not in an SSH whitelist



Analysis Pipeline – Outbound SSH Anomalies

```
EVALUATION outbound_ssh_tracking
  FILTER outbound_SSH
  FOREACH sourceIPv4Address destinationIPv4Address
  CHECK THRESHOLD
    SUM PACKETS>4
    TIME WINDOW 1 MINUTES
  END CHECK
  OUTPUT TIMEOUT 12 HOURS
  ALERT 1 TIMES 5 MINUTES
  ALERT EACH_ONLY_ONCE
  CLEAR ALWAYS
END EVALUATION
```



Analysis Pipeline – Outbound SSH Anomalies

- From aux.log

2015-10-31

22:17:23|Evaluation|outbound_ssh_tracking|1|SIP,DIP|198.51.100.222
,192.0.2.77|SUM PACKETS|1762634|

- Source IP – NAT'd IP from our public network. Need to check the firewall logs to get private IP of client.
- Destination IP – unknown external SSH server
 - Obviously not in our ssh whitelist
 - HIGH volume of traffic (1,762,634 packets in one day)
 - Need DNS information



Orcus – Outbound SSH Anomalies Pivot

- What does the external IP resolve to?

```
$ orlookup --start-date=2015/10/31 --end-date=2015/10/31 --address=192.0.2.77
```

```
date|name|address|source
```

```
2015-10-21|net.akamaiedge.ce.e0000|192.0.2.77|A
```

- Doesn't tell us much. Need to find out the internal of the machine generating this traffic. From firewall logs:

```
2015-10-31T00:12:43-05:00 fw.host : Built dynamic TCP translation from inside:192.168.1.34/61077 to border:198.51.100.222/61077
```

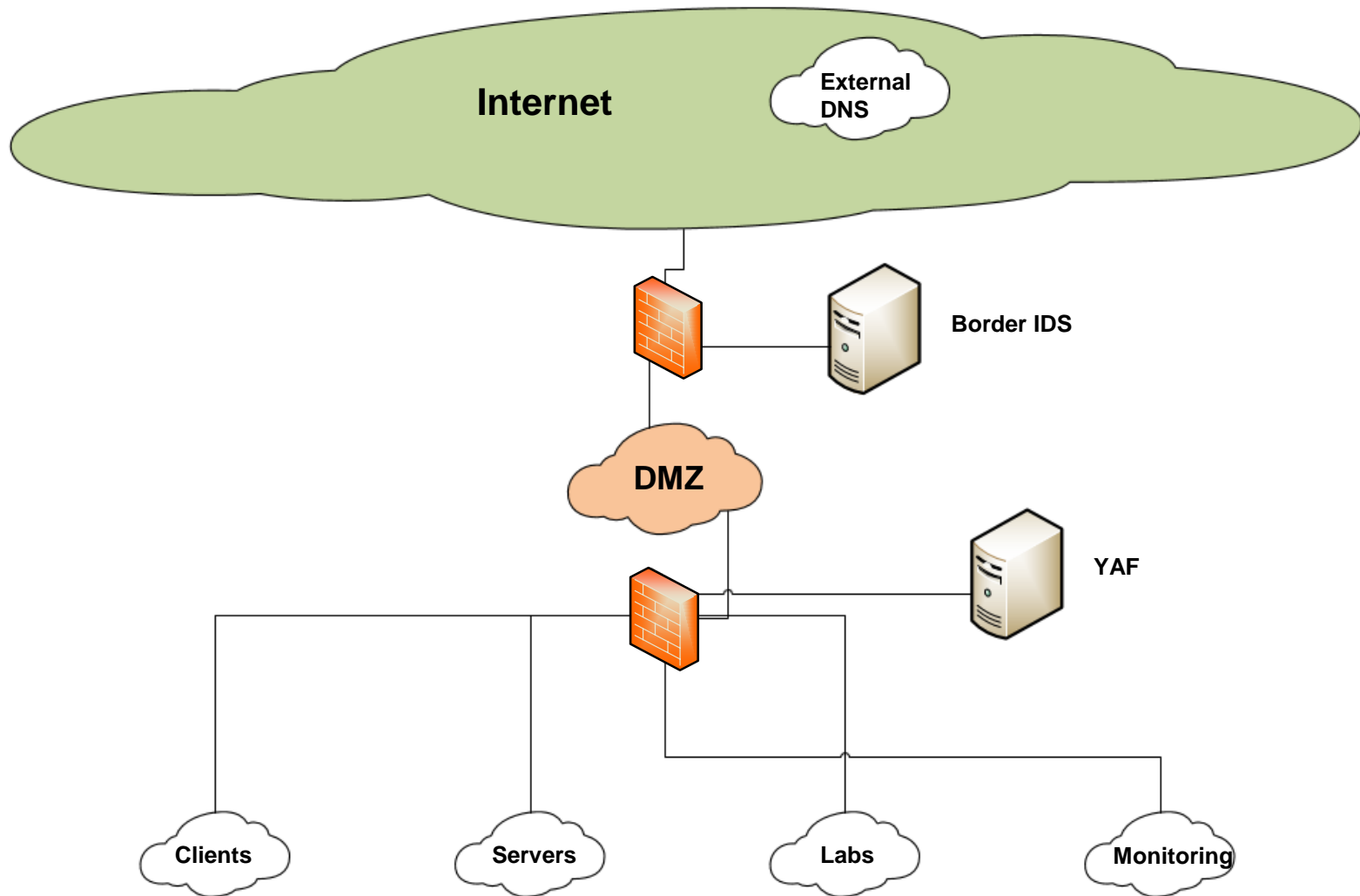


AD – Outbound SSH Anomalies Pivot

- 192.168.1.34 – search in SIEM for Windows Security Logs
 - User is st_smith
 - Confront user about traffic
 - Learn it's the user's private site being hosted via Akamai
 - Discipline + policy adjustments if necessary



Orcus – Augments IDS Coverage



Orcus – Finding source of malicious lookups

IDS Alert for Malicious Domain Lookup

```
11/10/15-05:02:44 [1:111:1] <eth2> Malicious Domain  
Lookup: www.i-am-bad-also.com {UDP}  
198.51.100.20:62943 -> 192.0.2.79:53
```

- Source IP – Public NAT address of our resolver
- Destination IP – Some unknown public DNS server
- Who actually queried our resolver in the first place?
 - IDS only monitors border
 - Doesn't capture internal client to server query
 - YAF saw it



Orcus - Orquery

- Who wanted to know what www.i-am-bad-also.com resolved to?

```
$ orquery --start-date=2015/11/10 --end-date=2015/11/10 --rr-name=com.i-am-bad-also.www
```

```
2015/11/10T05:02:44.043|internalfw0|int|A|192.168.1.7
```

```
2015/11/10T05:02:44.043|internalfw0|int|A|192.168.1.75
```

- Internal IP of resolver
- Client that initiated lookup
- Investigate client for signs of compromise



Conclusion

- Netflow can be a great tool to help strengthen your security posture and intrusion detections monitoring techniques
- Cannot function solely as replacement for existing security solutions, but can help make intrusion detection and analysis more efficient
- Other tools are still needed:
 - IDS/IPS
 - PCAPs
 - Web, Server, VPN, and Firewall logs
 - Proxy Logs



Questions...?

Mike Pochan

Software Engineering Institute

Carnegie Mellon

mjpochan@sei.cmu.edu

412-268-6293

