

Netflow

Campus Network Design & Operations Workshop



These materials are licensed under the Creative Commons Attribution-NonCommercial 4.0 International license
(<http://creativecommons.org/licenses/by-nc/4.0/>)



UNIVERSITY OF OREGON

Last updated 14th December 2018



Agenda

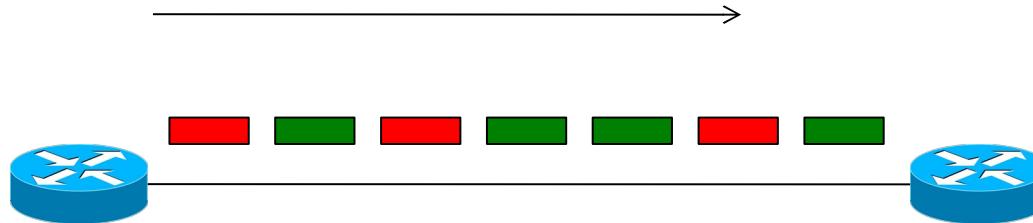
- Netflow
 - What it is and how it works
 - Uses and applications
- Generating and exporting flow records
- Nfdump and NfSen
 - Architecture
 - Usage
- Lab

What is a Network Flow

- A set of related packets
- Packets that belong to the same transport connection. e.g.
 - TCP, same src IP, src port, dst IP, dst port
 - UDP, same src IP, src port, dst IP, dst port
 - Some tools consider "bidirectional flows", i.e. A->B and B->A as part of the same flow

[http://en.wikipedia.org/wiki/Traffic_flow_\(computer_networking\)](http://en.wikipedia.org/wiki/Traffic_flow_(computer_networking))

Simple flows



 = Packet belonging to flow X

 = Packet belonging to flow Y

Cisco IOS Definition of a Flow

- Unidirectional sequence of packets sharing:
 - Source IP address
 - Destination IP address
 - Source port for UDP or TCP, 0 for other protocols
 - Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
 - IP protocol
 - Ingress interface (SNMP ifIndex)
 - IP Type of Service

IOS: which of these six packets are in the same flows?

	<i>Src IP</i>	<i>Dst IP</i>	<i>Protocol</i>	<i>Src Port</i>	<i>Dst Port</i>
A	1.2.3.4	5.6.7.8	6 (TCP)	4001	22
B	5.6.7.8	1.2.3.4	6 (TCP)	22	4001
C	1.2.3.4	5.6.7.8	6 (TCP)	4002	80
D	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
E	1.2.3.4	8.8.8.8	17 (UDP)	65432	53
F	8.8.8.8	1.2.3.4	17 (UDP)	53	65432

IOS: which of these six packets are in the same flows?

	<i>Src IP</i>	<i>Dst IP</i>	<i>Protocol</i>	<i>Src Port</i>	<i>Dst Port</i>
A	1.2.3.4	5.6.7.8	6 (TCP)	4001	22
B	5.6.7.8	1.2.3.4	6 (TCP)	22	4001
C	1.2.3.4	5.6.7.8	6 (TCP)	4002	80
D	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
E	1.2.3.4	8.8.8.8	17 (UDP)	65432	53
F	8.8.8.8	1.2.3.4	17 (UDP)	53	65432

What about packets “C” and “D”?

Flow Accounting

- A summary of all the packets seen in a flow (so far):
 - Flow identification: protocol, src/dst IP/port...
 - Packet count
 - Byte count
 - Start and end times
 - Maybe additional info, e.g. AS numbers, netmasks
- Records traffic volume and type but not content

Uses and Applications

- You can answer questions like:
 - Which user / department has been uploading / downloading the most?
 - Which are the most commonly-used protocols on my network?
 - Which devices are sending the most SMTP traffic, and to where?
- Identification of anomalies and attacks
- More fine-grained visualisation (graphing) than can be done at the interface level

Working with flows

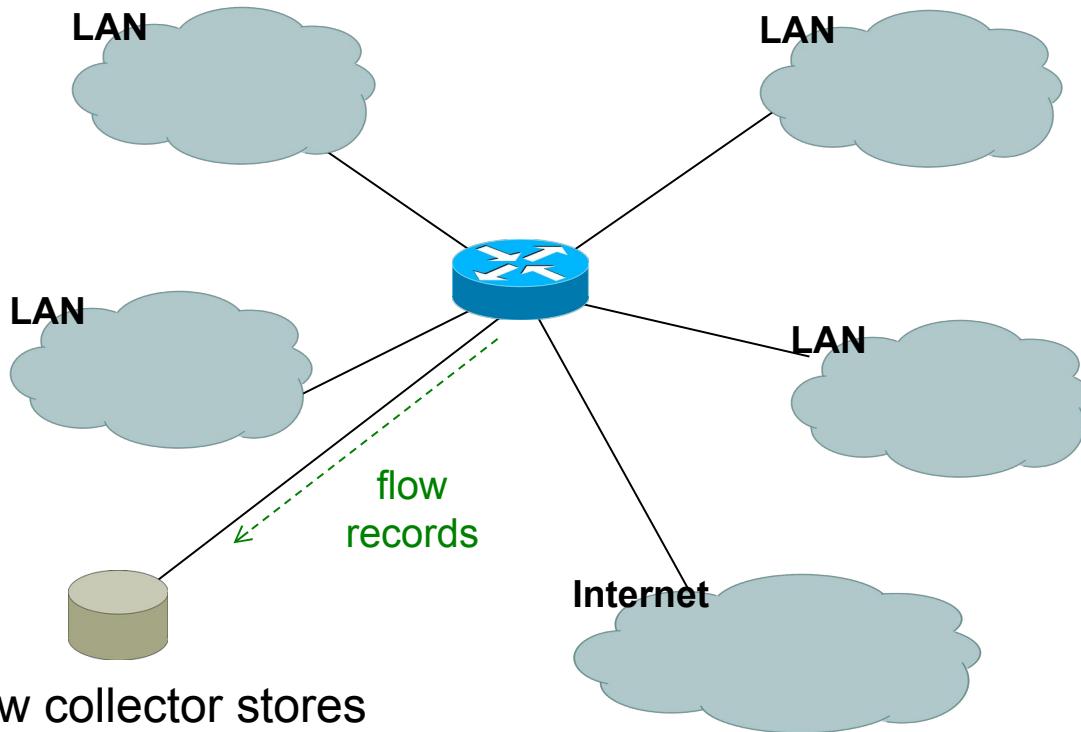
1. Configure device (e.g. router) to generate flow accounting records
2. Export the flows from the device (router) to a collector (PC)
 - Configure protocol version and destination
3. Receive the flows, write them to disk
4. Analyse the flows

Many tools available, both free and commercial

Where to generate flow records

1. On a router or other network device
 - If the device supports it
 - No additional hardware required
 - Might have some impact on performance
2. Passive collector (usually a Unix host)
 - Receives a copy of every packet and generates flows
 - Requires a mirror port
 - Resource intensive

Flow Collection

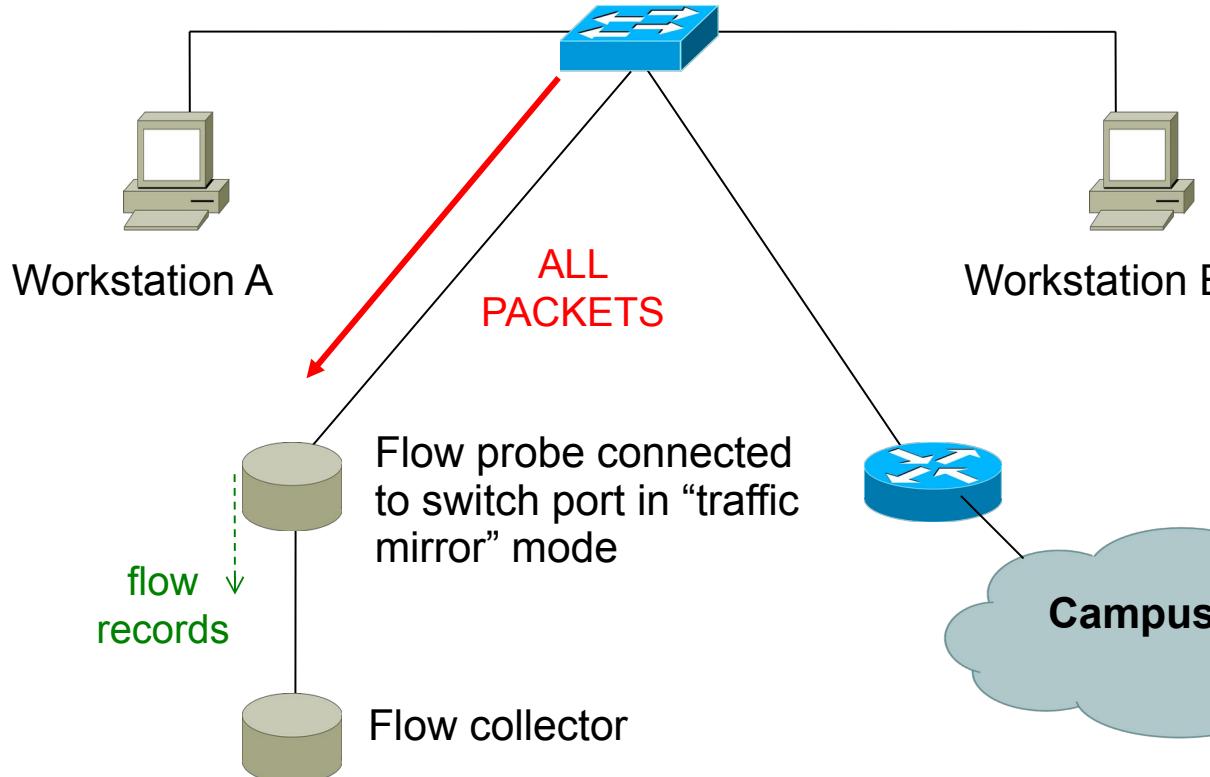


Flow collector stores
exported flows from router.

Flow Collection

- All flows through router can be observed
- Router overhead to process & export flows
- Can select which interfaces Netflow collection is needed on and not activate it on others
- If router on each LAN, Netflow can be activated on them to reduce load on core router

Passive Monitor Collection



Passive Collector

- Examples
 - softflowd (Linux/BSD)
 - pfflowd (BSD)
 - ng_netflow (BSD)
- Collector sees all traffic through the network point it is connected on and generates flows
- Relieves router from processing traffic, creating flows and exporting them

A thought:

Your network probably already has a device which is keeping track of IP addresses and port numbers of traffic flowing through it.

What is it?

Flow Export Protocols

- Cisco **Netflow**, different versions
 - v5: widely deployed
 - v9: newer, extensible, includes IPv6 support
- IP Flow Information Export (**IPFIX**):
 - IETF standard, based on Netflow v9
- **sFlow**: Sampling-based, commonly found on switches
- **jFlow**: Juniper
- We use Netflow, but many tools support multiple protocols

Cisco Netflow

- Unidirectional flows
- IPv4 unicast and multicast
 - (IPv6 in Netflow v9)
- Flows exported via UDP
 - Choose a port. No particular standard, although 2055 and 9996 are commonly used
- Supported on all IOS variants, as well as older ASA and CatOS platforms – but with different implementations

Cisco IOS Configuration

- Configured on each interface
 - Inbound and outbound
 - Older IOS only allows input
- Define the version
- Define the IP address and port of the collector (where to send the flows)
- Optionally enable aggregation tables
- Optionally configure flow timeout and main (v5) flow table size
- Optionally configure sample rate

Configuring Netflow: old platforms

- Enable CEF

```
ip cef  
ipv6 cef
```

- Enable flow on each interface

```
ip route-cache flow
```

(pre IOS 12.4)

OR

```
ip flow ingress  
ip flow egress
```

(IOS 12.4 onwards)

- Exporting Flows to a collector

```
ip flow-export version [5|9] [origin-as|peer-as]  
ip flow-export destination <x.x.x.x> <udp-port>
```

Netflow “Top-talkers”

- You can summarise flows directly on the router

- Configure the top-talkers feature

```
ip flow-top-talkers
  top 50
  sort-by bytes
  match input-interface GigabitEthernet0/0
```

- Will sort the output according to the top 50 incoming traffic source/destination pairs on GigabitEthernet0/0 interface

- And then show the top-talkers:

```
show ip flow top-talkers
```

Flexible Netflow: newer platforms

- Only way to monitor IPv6 flows on modern IOS
- Flexible Netflow available from IOS 15.0 onwards (and other IOS variants)
- Start using it now
 - IPv6 is deployed and flow info needed too
- Many mind-boggling options available, but basic configuration is straightforward

Flexible Netflow Configuration

- Define one or more exporters:

```
flow exporter EXPORTER-1
  destination 192.0.2.99
  transport udp 9996
  source Loopback0
  template data timeout 300
```

Flexible Netflow Configuration

- Define one or more flow monitors:

```
flow monitor FLOW-MONITOR-V4
  exporter EXPORTER-1
  cache timeout active 300
  record netflow ipv4 original-input
```

```
flow monitor FLOW-MONITOR-V6
  exporter EXPORTER-1
  cache timeout active 300
  record netflow ipv6 original-input
```

Flexible Netflow Configuration

- Apply flow monitors to active interface

```
interface GigabitEthernet0/0/0
    ip flow monitor FLOW-MONITOR-V4 input
    ip flow monitor FLOW-MONITOR-V4 output
    ipv6 flow monitor FLOW-MONITOR-V6 input
    ipv6 flow monitor FLOW-MONITOR-V6 output
```

Flexible Netflow “Top-talkers”

- You can summarise flows directly on the router e.g.

```
show flow monitor FLOW-MONITOR-V4 cache aggregate ipv4 source
address ipv4 destination address sort counter bytes top 50
```

- Yes, that is one long command!
 - Yes, the output needs a 280 character wide display!

- Or on some newer platforms:

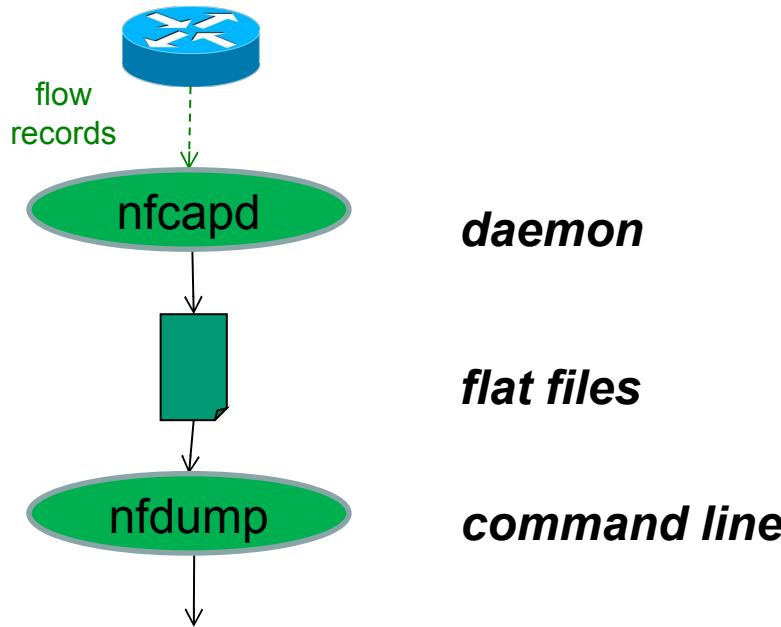
```
show flow monitor FLOW-MONITOR-V4 cache sort highest ipv4
source address top 50 format table
```

- Make an alias:

```
conf t
alias exec top-talkers show flow ...
```

Questions?

Nfdump architecture



Date	flow	start	Duration	Proto	Src IP	Addr:Port	Dst IP	Addr:Port	Packets	Bytes	Flows
2013-04-18		13:35:23.353	1482.000	UDP	10.10.0.119:55555	->	190.83.150.177:54597		8683	445259	1
2013-04-18		13:35:23.353	1482.000	UDP	190.83.150.177:54597	->	10.10.0.119:55555		8012	11.1 M	1
2013-04-18		13:48:21.353	704.000	TCP	196.38.180.96:6112	->	10.10.0.119:62099		83	20326	1
2013-04-18		13:48:21.353	704.000	TCP	10.10.0.119:62099	->	196.38.180.96:6112		105	5085	1

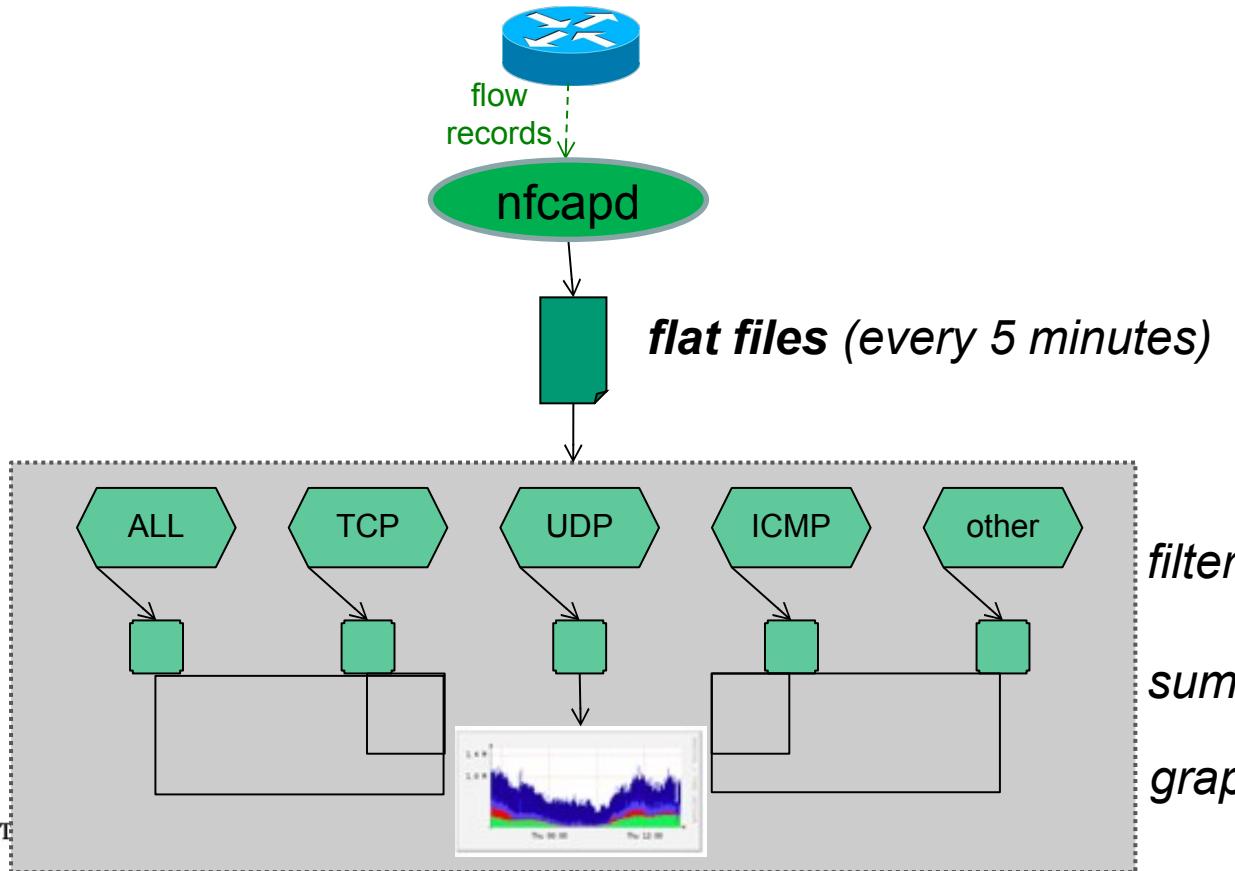
Collecting flows: Nfdump

- Free and open source – Runs on collector
- *nfcapd* listens for incoming flow records and writes them to disk (flat files)
 - typically starts a new file every 5 minutes
- *nfdump* reads the files and turns them into human-readable output
- *nfdump* has command-line options to filter and aggregate the flows

Analysing flows: NfSen

- Companion of Nfdump
- Web GUI
- Creates RRD graphs of traffic totals
- Lets you zoom into a time of interest and do nfdump analysis
- Manages nfcapd instances for you
 - Can run multiple nfcapd instances for listening to flows from multiple routers
- Plugins available like port tracker, surfmap

NfSen architecture



NfSen: points to note

- Every 5 minutes *nfcapd* starts a new file, and *nfsen* processes the previous one
- Hence each graph point covers 5 minutes
- The graph shows you the ***total*** of selected traffic in that 5-minute period
- To get more detailed information on the individual flows in that period, the GUI lets you drill down using *ndump*

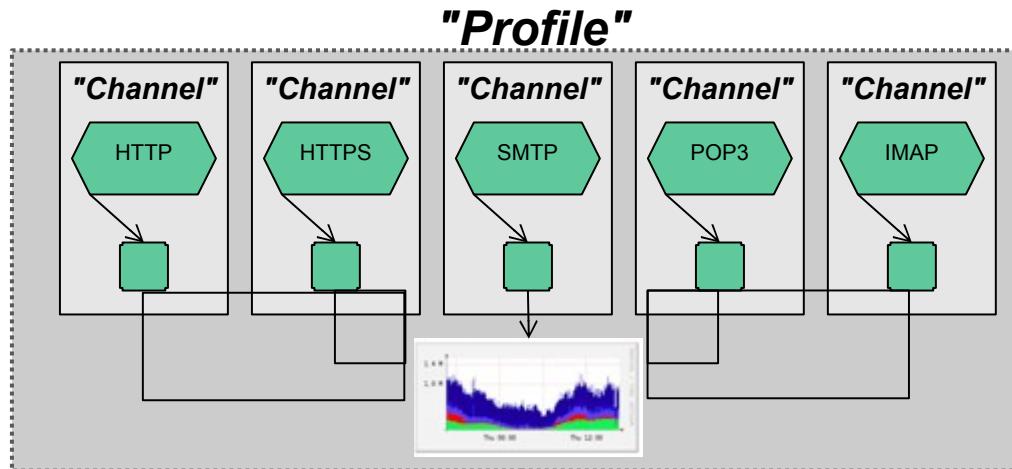
Demonstration

- Now we will use *nfsen* to find biggest users of bandwidth
- The instructors will demonstrate the traffic usage of the workshop network this week

Profiles and Channels

- A “channel” identifies a type of traffic to graph, and a “profile” is a collection of channels which can be shown together
- You can create your own profiles and channels, and hence graphs. e.g.
 - Total HTTP, HTTPS, SMTP traffic (etc)
 - Traffic to and from the Science department
 - ...
- Use filters to define the traffic of interest

Profiles and Channels



References – Tools

- nfdump and nfsen:
 - <http://nfdump.sourceforge.net/>
 - <http://nfsen.sourceforge.net/>
 - <http://nfsen-plugins.sourceforge.net/>
- pmacct and pmgraph:
 - <http://www.pmacct.net/>
 - <http://www.aptivate.org/pmgraph/>
- flow-tools:
 - <http://www.splintered.net/sw/flow-tools>

References – Further Info

- WikiPedia:
 - <http://en.wikipedia.org/wiki/Netflow>
- IETF standards effort
 - <http://www.ietf.org/html.charters/ipv4-charter.html>
- Abilene NetFlow page
 - <http://abilene-netflow.itec.oar.net/>
- Cisco Centric Open Source Community
 - <http://cosi-nms.sourceforge.net/related.html>
- Cisco NetFlow Collector User Guide
 - http://www.cisco.com/en/US/docs/net_mgmt/netflow_collection_engine/6.0/tier_one/user_guide/user.html

Questions?

- (Additional Reference Materials Follow)

Filter Examples

any	<i>all traffic</i>
proto tcp	<i>only TCP traffic</i>
dst host 1.2.3.4	<i>only traffic to 1.2.3.4</i>
dst net 10.10.1.0/24	<i>only traffic to that range</i>
not dst net 10.10.1.0/24	<i>only traffic <u>not</u> to that range</i>
proto tcp and src port 80	<i>only TCP with source port 80</i>
dst net 10.10.1.0/24 or dst net 10.10.2.0/24	<i>only traffic to those nets</i>
dst net 10.10.1.0/24 and proto tcp and src port 80	<i>only HTTP response traffic to that net</i>
(dst net 10.10.1.0/24 or dst net 10.10.2.0/24) and proto tcp and src port 80	
<i>...more complex combinations possible</i>	

Flows and Applications

More Examples

Uses for Netflow

- Problem identification / solving
 - Traffic classification
 - DoS Traceback (some slides by Danny McPherson)
- Traffic Analysis and Engineering
 - Inter-AS traffic analysis
 - Reporting on application proxies
- Accounting (or billing)
 - Cross verification from other sources
 - Can cross-check with SNMP data

Detect Anomalous Events: SQL 'Slammer' Worm*

peakflow |DoS

Recent Anomalies: Anomaly 125772 : Detailed 11:51:49 EST 27 Jan 2003

Statistics

Status Topology Ongoing Recent Dark IP Admin About

Anomaly 125772 Detailed Statistics

ID	Importance	Severity	Duration	Direction	Resource	Start Time	End Time	Class
125772	High	968.2% of 3.40 Kbps	09 h 06m 47s	Outgoing	Group#3 192.168.16.0/20 members_misc_nets	00:33:27 EST 25 Jan 2003	09:40:14 EST 25 Jan 2003	Profiled

pps of net8 for anomaly 125772

Affected Network Elements

Router net8 1.2.3.4

	Triggering	Expected	Difference	Maximum	Mean
Bitrate	71.69 Mbps	2.34 Mbps	69.35 Mbps	105.26 Mbps @ 03:19	76.05 Mbps
Packet Rate	22.49 Kpps	712 pps	21.49 Kpps	32.58 Kbps @ 03:19	23.54 Kbps

[Summary](#) | [Source Addresses](#) | [Destination Addresses](#) | [Source Ports](#) | [Destination Ports](#) | [Protocols](#) | [Output Interfaces](#) | [Input Interfaces](#) | [Generate Filter](#)

Summary of all Data Snapshots Collected:

	Bytes	Packets	Bytes/Pkt	bps	pps
	308.01 GB	762,849,500	404 B	76.05 Mbps	23.54 Kbps

[Summary](#) | [Source Addresses](#) | [Destination Addresses](#) | [Source Ports](#) | [Destination Ports](#) | [Protocols](#) | [Output Interfaces](#) | [Input Interfaces](#) | [Generate Filter](#)

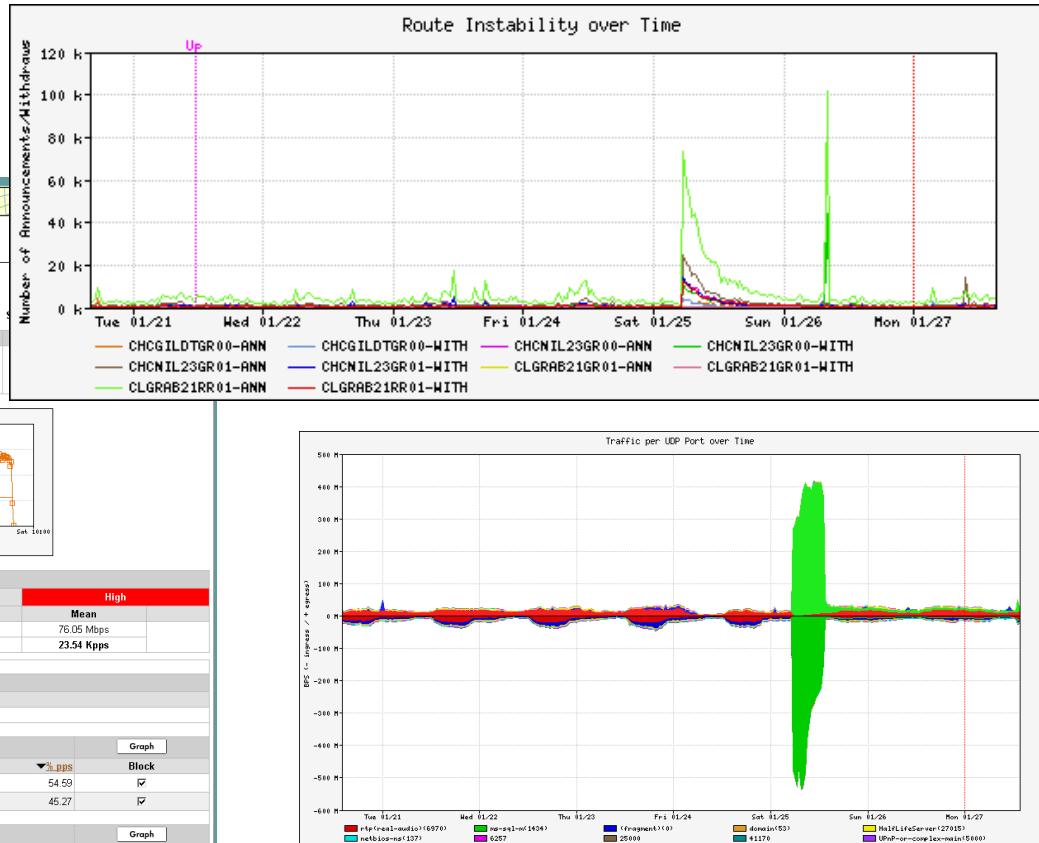
Source Addresses

Network / Mask	Bytes	▼Packets	Bytes/Pkt	bps	pps	▼%pps	Block
192.168.20.217/32	168.22 GB	416,436,800	404 B	41.54 Mbps	12.85 Kbps	54.59	☒
192.168.18.187/32	139.53 GB	345,372,800	404 B	34.45 Mbps	10.66 Kbps	45.27	☒

[Summary](#) | [Source Addresses](#) | [Destination Addresses](#) | [Source Ports](#) | [Destination Ports](#) | [Protocols](#) | [Output Interfaces](#) | [Input Interfaces](#) | [Generate Filter](#)

Destination Addresses

	Bytes	▼Packets	Bytes/Pkt	bps	pps	▼%pps	Block
							Graph



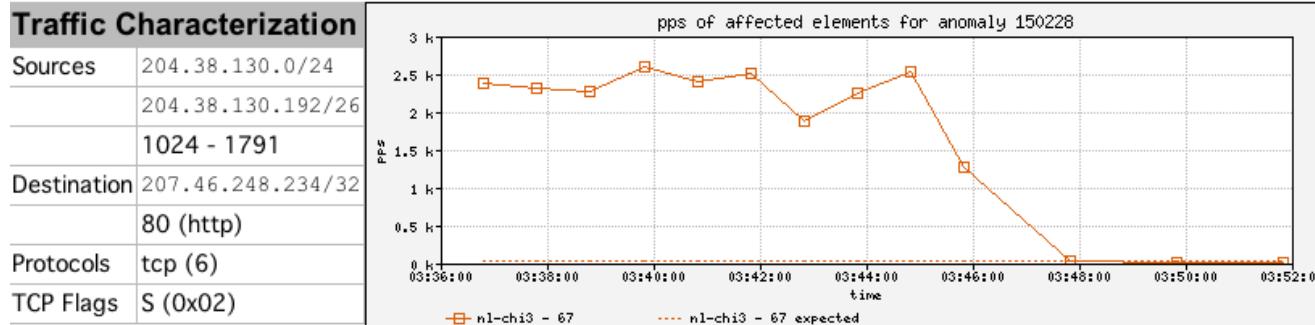
Flow-based Detection (cont)*

- Once baselines are built anomalous activity can be detected
 - Pure rate-based (pps or bps) anomalies may be legitimate or malicious
 - Many misuse attacks can be immediately recognized, even without baselines (e.g., TCP SYN or RST floods)
 - Signatures can also be defined to identify “interesting”
 - transactional data (e.g., proto udp and port 1434 and 404 octets(376 payload) == slammer!)
 - Temporal compound signatures can be defined to detect with higher precision

Flow-based Commercial Tools...*

Anomaly 150228					Get Report: PDF XML	
----------------	--	--	--	--	---	--

ID	Importance	Duration	Start Time	Direction	Type	Resource
150228	High 130.0% of 2 Kpps	17 mins	03:34, Aug 16	Incoming	Bandwidth (Profiled)	Microsoft 207.46.0.0/16 windowsupdate.com

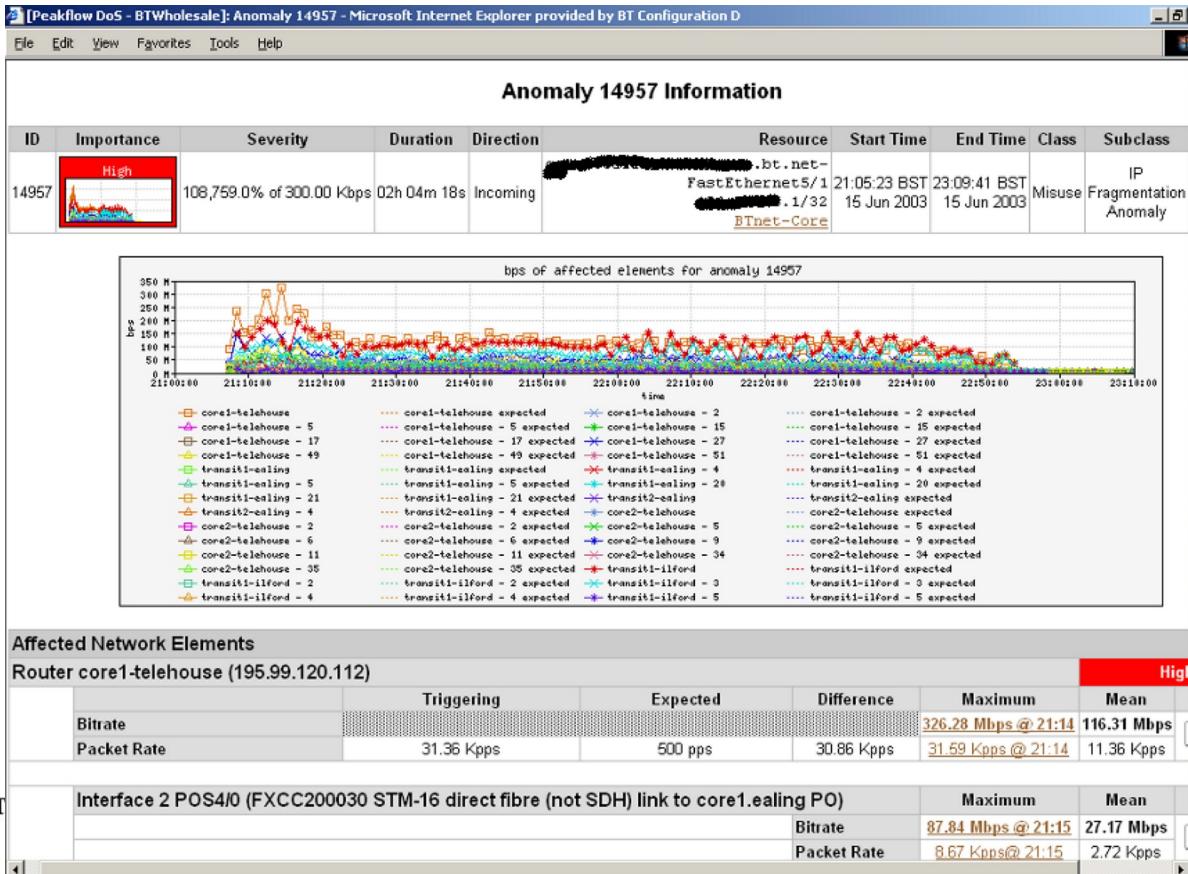


Affected Network Elements		Expected	Observed bps		Observed pps		
	Importance	pps	Max	Mean	Max	Mean	
Router nl-chi3 198.110.131.125	High						
Interface 67 at-1/1/0.14 pvc to WMU		26	832 K	563.1 K	2.6 K	1.7 K	Details

UNIV

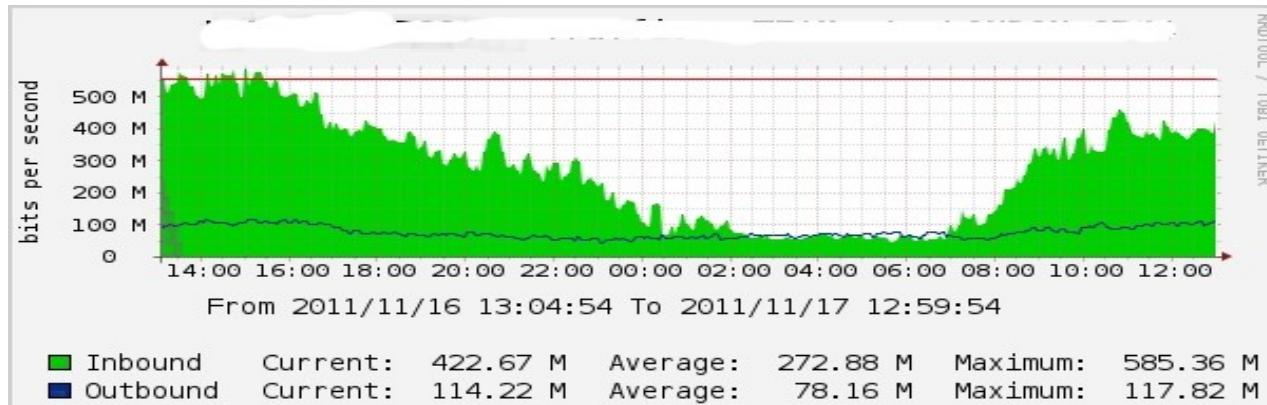
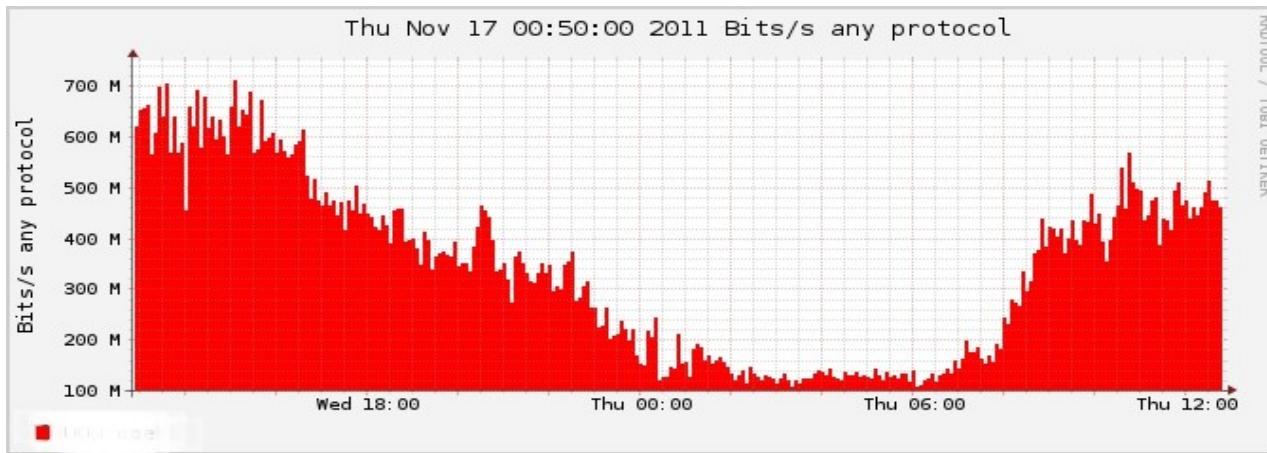
Anomaly Comments

Commercial Detection: A Large Scale DOS Attack



Accounting

- Flow based accounting can be a good supplement to SNMP based accounting



Cisco Netflow Versions

Netflow v1

- Key fields: Source/Destination IP, Source/Destination Port, IP Protocol, ToS, Input interface.
- Accounting: Packets, Octets, Start/End time, Output interface
- Other: Bitwise OR of TCP flags.
- Does not have sequence numbers – no way to detect lost flows
- Obsolete

Netflow v2 to v4

- Cisco internal
- Were never released

Netflow v5

- Key fields: Source/Destination IP, Source/Destination Port, IP Protocol, ToS, Input interface.
- Accounting: Packets, Octets, Start/End time, Output interface.
- Other: Bitwise OR of TCP flags, Source/Destination AS and IP Mask.
- Packet format adds sequence numbers for detecting lost exports.
- IPv4 only

Netflow v6 & v7

- Used exclusively on the Cisco Catalyst line of ethernet switches
- Requires the Netflow Feature Card, a specialist forwarding engine for the Catalyst Switches
- Not compatible or comparable with Netflow on Cisco routers

Netflow v8

- Aggregated v5 flows.
- Not all flow types available on all equipment
- Much less data to post process, but loses fine granularity of v5 – no IP addresses.

Netflow v9

- IPv6 support
- 32-bit ASN support
- Additional fields like MPLS labels
- Builds on earlier versions
- Periodically sends "template" packet, all flow data fields reference the template

Questions?