HUNTING ATTACKERS WITH NETWORK AUDIT TRAILS

Tom Cross
tcross@lancope.com

Charles Herring
cherring@lancope.com
CREATING THE AUDIT TRAIL
Creating the Trail

Logging

• Provides user and application details
• Requires translation and aggregation
• Limited to configured hosts
• Subject to manipulation

Packet Captures

• Provide complete network record and unencrypted content
• Large storage requirement limits retention length
Netflow Basics

- Devices with one or more Flow producing interfaces are “Exporters”
- Exporters cache and forward records to “Collectors”
- Common Exporters include firewalls, switches, and routers
Transactional Audits of ALL activities

- Much like a phone bill

<table>
<thead>
<tr>
<th>Start Active Time</th>
<th>Client Host</th>
<th>Client Zone</th>
<th>Server Host</th>
<th>Server Zone</th>
<th>Service Summary</th>
<th>Service Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 12, 2010 8:41:56 AM (6 hours 32 minutes 605 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.21.202.71</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:43:14 AM (6 hours 20 minutes 523 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>216.165.129.141</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:45:51 AM (6 hours 28 minutes 153 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>68.142.118.82</td>
<td>LimeLight Networks</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:43:34 AM (6 hours 30 minutes 325 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.21.202.96</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:52:48 AM (6 hours 21 minutes 184 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>10.202.1.223</td>
<td>Engineering</td>
<td>http-alt (8080/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 7:22:53 AM (7 hours 51 minutes 135 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>10.202.1.223</td>
<td>Engineering</td>
<td>http-alt (8080/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 12:13:13 PM (3 hours 53 minutes ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>10.202.1.223</td>
<td>Engineering</td>
<td>http-alt (8080/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 9:02:34 AM (6 hours 11 minutes 325 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.233.96.254</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:43:36 AM (6 hours 30 minutes 305 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.167.164.64</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:57:33 AM (6 hours 16 minutes 333 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.21.202.165</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 10:16:50 AM (4 hours 57 minutes 160 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>10.201.0.15</td>
<td>Sales and Marketing</td>
<td>ldap (389/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:43:55 AM (6 hours 30 minutes 315 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>63.245.217.21</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 2:59:36 PM (14 minutes 305 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>72.5.124.55</td>
<td>United States</td>
<td>http (80/tcp)</td>
<td></td>
</tr>
<tr>
<td>Apr 12, 2010 8:42:09 AM (6 hours 30 minutes 575 ago)</td>
<td>10.201.3.96</td>
<td>Sales and Marketing</td>
<td>63.245.209.115</td>
<td>United States</td>
<td>https (443/tcp)</td>
<td></td>
</tr>
</tbody>
</table>
Create New TCP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:06</td>
<td>1</td>
<td>195</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>S</td>
</tr>
</tbody>
</table>
### Create New TCP Flow

Ingress and Egress ports are based on the interface on which the packets entered and left the router.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:06</td>
<td>1</td>
<td>195</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>S</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23:14:07</td>
<td>23:14:07</td>
<td>1</td>
<td>132</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SA</td>
</tr>
</tbody>
</table>

**NETFLOW CACHE**

- **SYN/ACK**: 1024
- **10.1.1.1**: 80
- **10.2.2.2**: TCP
- **Data**
Update Existing TCP Flow

Packet and Byte counts are incremented accordingly. Last Seen is also updated.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23.14.07</td>
<td>23.14.07</td>
<td>1</td>
<td>132</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SA</td>
</tr>
</tbody>
</table>

NETFLOW CACHE

Data | TCP | 10.1.1.1 | 1024 | 10.2.2.2 | 80 | ACK
### Update Existing TCP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23.14:07</td>
<td>23.14:08</td>
<td>2</td>
<td>862</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SAP</td>
</tr>
</tbody>
</table>

**Data**

| ACK/PSH | 1024 | 10.1.1.1 | 80 | 10.2.2.2 | TCP | Data |

**NETFLOW CACHE**
### Create New UDP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23:14:07</td>
<td>23:14:08</td>
<td>2</td>
<td>862</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SAP</td>
</tr>
<tr>
<td>UDP</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>10.2.8.12</td>
<td>53</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>176</td>
<td>Gi4/12</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
</tbody>
</table>

### NETFLOW CACHE

```
Data  UDP  10.3.1.1  2918  10.2.8.12  53
```
# Create New UDP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23:14:07</td>
<td>23:14:08</td>
<td>2</td>
<td>862</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SAP</td>
</tr>
<tr>
<td>UDP</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>10.2.8.12</td>
<td>53</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>176</td>
<td>Gi4/12</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
<tr>
<td>UDP</td>
<td>10.2.8.12</td>
<td>53</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>212</td>
<td>Gi2/1</td>
<td>Gi4/12</td>
<td>-</td>
</tr>
</tbody>
</table>

**NETFLOW CACHE**

```
2918  10.3.1.1  53  10.2.8.12  UDP  Data
```
## Create New ICMP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23:14:07</td>
<td>23:14:08</td>
<td>2</td>
<td>862</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SAP</td>
</tr>
<tr>
<td>UDP</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>10.2.8.12</td>
<td>53</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>176</td>
<td>Gi4/12</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
<tr>
<td>UDP</td>
<td>10.2.8.12</td>
<td>53</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>212</td>
<td>Gi2/1</td>
<td>Gi4/12</td>
<td>-</td>
</tr>
<tr>
<td>ICMP</td>
<td>10.1.1.4</td>
<td>-</td>
<td>10.2.8.14</td>
<td>ECHO-REQUEST</td>
<td>23:14:12</td>
<td>23:14:12</td>
<td>1</td>
<td>96</td>
<td>Gi4/19</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
</tbody>
</table>

Most NetFlow caches do not offer ICMP type and code fields so the Destination Port column is overloaded with with ICMP information.

![Netflow Cache Diagram]
## Update Existing ICMP Flow

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>First Seen</th>
<th>Last Seen</th>
<th>Packets</th>
<th>Bytes</th>
<th>Ingress Interface</th>
<th>Egress Interface</th>
<th>TCP Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>10.2.2.2</td>
<td>80</td>
<td>23:14:06</td>
<td>23:14:08</td>
<td>2</td>
<td>425</td>
<td>Gi4/13</td>
<td>Gi2/1</td>
<td>SA</td>
</tr>
<tr>
<td>TCP</td>
<td>10.2.2.2</td>
<td>80</td>
<td>10.1.1.1</td>
<td>1024</td>
<td>23:14:07</td>
<td>23:14:08</td>
<td>2</td>
<td>862</td>
<td>Gi2/1</td>
<td>Gi4/13</td>
<td>SAP</td>
</tr>
<tr>
<td>UDP</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>10.2.8.12</td>
<td>53</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>176</td>
<td>Gi4/12</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
<tr>
<td>UDP</td>
<td>10.2.8.12</td>
<td>53</td>
<td>10.3.1.1</td>
<td>2918</td>
<td>23:14:11</td>
<td>23:14:11</td>
<td>1</td>
<td>212</td>
<td>Gi2/1</td>
<td>Gi4/12</td>
<td>-</td>
</tr>
<tr>
<td>ICMP</td>
<td>10.1.1.4</td>
<td>-</td>
<td>10.2.8.14</td>
<td>ECHO-REQUEST</td>
<td>23:14:12</td>
<td>23.14:13</td>
<td>2</td>
<td>192</td>
<td>Gi4/19</td>
<td>Gi2/1</td>
<td>-</td>
</tr>
</tbody>
</table>

**NETFLOW CACHE**

- Data
- ICMP
- ECHO-REQUEST
- 10.1.1.4
- 10.2.8.14
Netflow Pros & Cons

• Doesn’t need to be configured per device
• Visibility down to the access layer
• Compact records can provide long history
• Doesn’t store content

NetFlow Collector

<table>
<thead>
<tr>
<th>NetFlow Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>src and dst ip</td>
</tr>
<tr>
<td>src and dst port</td>
</tr>
<tr>
<td>start time</td>
</tr>
<tr>
<td>end time</td>
</tr>
<tr>
<td>mac address</td>
</tr>
<tr>
<td>byte count</td>
</tr>
<tr>
<td>- more -</td>
</tr>
</tbody>
</table>
Intrusion Audit Trails

Do you know what went on while you were mitigating?

- 1:06:15 PM: Internal Host Visits Malicious Web Site
- 1:06:30 PM: Malware Infection Complete, Accesses Internet Command and Control
- 1:06:35 PM: Malware begins scanning internal network
- 1:13:59 PM: Multiple internal infected hosts
- 1:07:00 PM: Gateway malware analysis identifies the transaction as malicious
- 1:14:00 PM: Administrators manually disconnect the initial infected host

© 2013 Lancope, Inc. All rights reserved.
Visibility through out the Kill Chain

- Netflow is excellent at observing some stages of the kill chain
  - Reconnaissance – Both external and internally post infection
  - Unknown Command and Control channels
  - Internal Pivoting and Probing
  - Data Exfiltration
Logging Standards

- Netflow v5
- NetFlow v9 (RFC-3954)
- IPFIX (RFC-5101)
  - Sflow – Foundry, Extreme, Procurve
  - Jflow – Juniper
  - Cflowd – Juniper/Alcatel-Lucent
  - NetStream – 3Com/Huawei
  - Rflow – Ericsson
  - AppFlow - Citrix

Basic/Common Fields
### NetFlow v5*

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 3</td>
<td>srcaddr</td>
<td>Source IP address.</td>
</tr>
<tr>
<td>4 to 7</td>
<td>dstaddr</td>
<td>Destination IP address.</td>
</tr>
<tr>
<td>8 to 11</td>
<td>nexthop</td>
<td>IP address of the next hop router.</td>
</tr>
<tr>
<td>12 to 15</td>
<td>input and output</td>
<td>SNMP index of the input and output interfaces.</td>
</tr>
<tr>
<td>16 to 19</td>
<td>dPkts</td>
<td>Packets in the flow.</td>
</tr>
<tr>
<td>20 to 23</td>
<td>dOctets</td>
<td>Total number of Layer 3 bytes in the flow’s packets.</td>
</tr>
<tr>
<td>24 to 27</td>
<td>First</td>
<td>SysUptime at start of flow.</td>
</tr>
<tr>
<td>28 to 31</td>
<td>Last</td>
<td>SysUptime at the time the last packet of flow was received.</td>
</tr>
<tr>
<td>32 to 35</td>
<td>srcport and dstport</td>
<td>TCP/UDP source and destination port number or equivalent.</td>
</tr>
<tr>
<td>36 to 39</td>
<td>pad1, tcp_flags, prot, and tos</td>
<td>Unused (zero) byte, cumulative OR of TCP flags, IP protocol (for example, 6 = TCP, 17 = UDP), and IP ToS.</td>
</tr>
<tr>
<td>40 to 43</td>
<td>src_as and dst_as</td>
<td>Autonomous system of the source and destination, either origin or peer.</td>
</tr>
<tr>
<td>44 to 47</td>
<td>src_mask, dst_mask, and pad2</td>
<td>Source and destination address prefix mask bits. Pad 2 is unused (zero) bytes.</td>
</tr>
</tbody>
</table>

* fixed format, cannot be extended to include new fields
Extensible Data Fields

Data sources can provide additional log information

Examples of Extensible Fields

- Network Based Application Recognition
- Performance Metrics (SRT/RRT, Collisions)
- HTTP Headers
- NAT Data
- Security Action (Permit/Deny)
- TTL
- DSCP
- Payload
1. Configure the Exporter

Router(config)# flow exporter my-exporter
Router(config-flow-exporter)# destination 1.1.1.1

2. Configure the Flow Record

Router(config)# flow record my-record
Router(config-flow-record)# match ipv4 destination address
Router(config-flow-record)# match ipv4 source address
Router(config-flow-record)# collect counter bytes

3. Configure the Flow Monitor

Router(config)# flow monitor my-monitor
Router(config-flow-monitor)# exporter my-exporter
Router(config-flow-monitor)# record my-record

4. Apply to an Interface

Router(config)# interface gi0/1
Router(config-if)# ip flow monitor my-monitor input
Router(config)# flow record my-record
Router(config-flow-record)# match
Router(config-flow-record)# collect

Router(config-flow-record)# match ?
  application          Application Fields
  datalink             Datalink (layer 2) fields
  flow                 Flow identifying fields
  interface            Interface fields
  ipv4                 IPv4 fields
  ipv6                 IPv6 fields
  routing              routing attributes
  transport            Transport layer field

Router(config-flow-record)# collect ?
  application          Application Fields
  counter              Counter fields
  datalink             Datalink (layer 2) fields
  flow                 Flow identifying fields
  interface            Interface fields
  ipv4                 IPv4 fields
  ipv6                 IPv6 fields
  routing              IPv4 routing attributes
  timestamp            Timestamp fields
  transport            Transport layer fields

Specify a Key Field
Specify a Non-Key Field
Router(config)# flow record my-record
Router(config-flow-record)# match ipv4 tos
Router(config-flow-record)# match ipv4 protocol
Router(config-flow-record)# match ipv4 destination address
Router(config-flow-record)# match ipv4 source address
Router(config-flow-record)# match transport source-port
Router(config-flow-record)# match transport destination-port
Router(config-flow-record)# collect routing destination as
Router(config-flow-record)# collect routing next-hop address ipv4
collect ipv4 dscp
Router(config-flow-record)# collect ipv4 ttl maximum
Router(config-flow-record)# collect ipv4 ttl minimum
Router(config-flow-record)# collect transport tcp flags
Router(config-flow-record)# collect interface output
Router(config-flow-record)# collect counter bytes
Router(config-flow-record)# collect counter packets
Router(config-flow-record)# collect timestamp sys-uptime first
collect timestamp sys-uptime last
NETFLOW TOOLS
SiLK

• Download at [http://tools.netsa.cert.org](http://tools.netsa.cert.org)
• Stores and processes flow
• Project Managed by Carnegie Mellon CERT
# Import the global variables needed for processing the record
global smtpports, counts

# Pull data from the record
sip = rec.sip
bytes = rec.bytes

# Get a reference to the current data on the IP address in question
data = counts.setdefault(sip, [0, 0])

# Update the total byte count for the IP address
data[0] += bytes

# Is the flow mail related? If so add the byte count to the mail bytes
if (rec.protocol == 6 and rec.sport in smtpports and
    rec.packets > 3 and rec.bytes > 120):
    data[1] += bytes
    return True

# If not mail related, fail the record
return False
## Other OpenSource Tools

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Primary Use</th>
<th>Comment</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cflowd</td>
<td>Traffic Analysis</td>
<td>No longer supported</td>
<td>UNIX</td>
</tr>
<tr>
<td>Flow-tools</td>
<td>Collector Device</td>
<td>Scalable</td>
<td>UNIX</td>
</tr>
<tr>
<td>Flowd</td>
<td>Collector Device</td>
<td>Support V9</td>
<td>BSD, Linux</td>
</tr>
<tr>
<td>FlowScan</td>
<td>Reporting for Flow-Tools</td>
<td></td>
<td>UNIX</td>
</tr>
<tr>
<td>IPFlow</td>
<td>Traffic Analysis</td>
<td>Support V9, IPv4, IPv6, MPLS, SCTP, etc..</td>
<td>Linux, FreeBSD, Solaris</td>
</tr>
<tr>
<td>NetFlow Guide</td>
<td>Reporting Tools</td>
<td></td>
<td>BSD, Linux</td>
</tr>
<tr>
<td>NetFlow Monitor</td>
<td>Traffic Analysis</td>
<td>Supports V9</td>
<td>UNIX</td>
</tr>
<tr>
<td>Netmet</td>
<td>Collector Device</td>
<td>V5, support v9</td>
<td>Linux</td>
</tr>
<tr>
<td>NTOP</td>
<td>Security Monitoring</td>
<td></td>
<td>UNIX</td>
</tr>
<tr>
<td>Stager</td>
<td>Reporting for Flow-Tools</td>
<td></td>
<td>UNIX</td>
</tr>
<tr>
<td>Nfdump/nfsen</td>
<td>Traffic Analysis</td>
<td>Support V5 and v9</td>
<td>UNIX</td>
</tr>
</tbody>
</table>

Different costs: implementation and customization
Commercial Solutions

- Arbor PeakFlow
- IBM Qradar
- Invea-Tech FlowMon
- Lancope StealthWatch
- ManageEngine
- McAfee NTBA
- Plixer Scrutinizer
- ProQSys FlowTraq
- Riverbed Cascade (formerly Mazu)

* For comparison see Gartner Network Behavior Analysis Market December 2012 (G00245584)
NETWORK AUDIT LOG DETECTION
Network Load Statistics

Throughput


Investigating Performance Issues

Traffic Statistics

<table>
<thead>
<tr>
<th></th>
<th>Total Bytes</th>
<th>Last (bps)</th>
<th>Mean (bps)</th>
<th>Peak (bps)</th>
<th>95th (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound</td>
<td>147.53G</td>
<td>435.1k</td>
<td>488.51k</td>
<td>18.59M</td>
<td>2.08M</td>
</tr>
<tr>
<td>Outbound</td>
<td>485.27G</td>
<td>6.5M</td>
<td>1.64M</td>
<td>19.43M</td>
<td>9.04M</td>
</tr>
</tbody>
</table>

©2011 Lancope, Inc. All Rights Reserved. Company Confidential (not for distribution)
Signature Matching

- IP Blacklists / IP Reputation
- Match Indicators Of Compromise for stored values
- Policy Enforcement
- Measure Compliance
What can you detect with the audit log?

Unsanctioned Device and Application Detection

- Identify the use of unsanctioned applications
- Detect rogue servers and other rogue devices
Firewall Policy Monitoring And Enforcement

- Audit Firewall rules
- Immediately detect misconfigurations
- Enforce policy between hosts on the same segment
What can you detect with the audit log?

Reveal BotNet Hosts

Layer 3

Layer 4 and URL
Behavioral Analysis

• Data Exfiltration/Collection
• Internal Pivot
• Worm Propagation
• Covert Channels
• Abnormal Behavior
## What Can Behavioral NetFlow Analysis Do?

### Reveal Recon

<table>
<thead>
<tr>
<th>Concern Index</th>
<th>Host Groups</th>
<th>Host</th>
<th>CI</th>
<th>CI%</th>
<th>Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Atlanta</td>
<td></td>
<td>3,520,635</td>
<td>1.17%</td>
<td>Excess Clients, Port_Scan</td>
</tr>
<tr>
<td></td>
<td>Atlanta</td>
<td></td>
<td>259.182.184.1</td>
<td>269%</td>
<td>Rejects, UDP_Scan</td>
</tr>
<tr>
<td></td>
<td>Sales and Marketing, Atlanta, Users, Windows</td>
<td>jbuchanan-d2.lancope.local (10.201.13.24)</td>
<td>15,095,525</td>
<td>160%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.90.10.254</td>
<td>91%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.20.10.254</td>
<td>83%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.40.10.254</td>
<td>83%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.80.10.254</td>
<td>83%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.70.10.254</td>
<td>82%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.50.10.254</td>
<td>81%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.100.10.254</td>
<td>80%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.20.10.254</td>
<td>77%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York</td>
<td></td>
<td>10.110.10.254</td>
<td>76%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>New York, Windows</td>
<td></td>
<td>10.60.10.254</td>
<td>72%</td>
<td>TCP_Scan</td>
</tr>
<tr>
<td></td>
<td>Atlanta</td>
<td></td>
<td>209.182.176.42</td>
<td>67%</td>
<td>Rejects</td>
</tr>
<tr>
<td></td>
<td>SSI Private</td>
<td></td>
<td>lscfs01.lancope.local (10.192.0.1)</td>
<td>5972.924</td>
<td>Ping, Ping_Oversized_Packet, Ping_Scan, Rejects</td>
</tr>
<tr>
<td></td>
<td>Sales and Marketing, Atlanta, Users, Windows</td>
<td>10.201.3.83</td>
<td>5903806</td>
<td>60%</td>
<td>Ping_Oversized_Packet, TCP_Scan</td>
</tr>
</tbody>
</table>
What Can Behavioral NetFlow Analysis Do?

Loss of Protected Data
FORENSIC INVESTIGATIONS USING THE NETWORK AUDIT TRAIL
Following IOC

Waterhole campaign targeting your industry has been publicly disclosed.

A quick search of your network audit trail reveals an internal host that accessed the disclosed site.
Following IOC

Check host details around that time

Suspicious HTTP connections right after contact - good candidate for a drive-by-download

Suspicious download followed by a reverse SSH shell. Most SSH bytes sent by “client”
Following IOC

Attacker recons your network. Investigate any hosts contacted by the compromised host. Additionally, look for any other hosts scanning for 445 and 135.
Following IOC

Since we have uncovered a new IOC (IP address controlling the reverse SSH shell), we should check to see if that host has touched the network anywhere else.

Another host showing a reverse shell
Large data transfer from your web server to an outside host was detected.
SQL Injection

Where did the data go?
Look for suspicious activity targeting the web server and your DMZ
Running Forensic Behavioral Reports

- Use Alarm history to pull “associated flows”. View Reports >> Alarm Report by Type for an extended period of time (e.g. 30 days) for a given host to see if there were any behavior changes
- Let StealthWatch Baseline normal activity and define times there are flows of interest

- Run “associated flows” from the Alarm Table
Combating Insider Threat is a multidisciplinary challenge

- IT cannot address insider threat by itself
  - People have a tendency to think that IT is solely responsible for all computer security issues.
- Legal: Are policies in place? Are they realistic? Does legal support IT practices?
- HR: Who is coming and going? Who has workplace issues? Are there soft solutions?
- IT: Is the privacy of end users adequately protected?
- What impact on workplace harmony are policies, monitoring, and enforcement having?
- Are you applying policies consistently?
Sometimes investigations start with user intelligence
Following the User

Alarm Table - 1 record

<table>
<thead>
<tr>
<th>Policy</th>
<th>Start Active Time</th>
<th>Source Host Group</th>
<th>Source User</th>
<th>Target</th>
<th>Target Hosts</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compliance Hosts</td>
<td>Jul 2, 2013 6:45:00 AM (1 day 4 hours 26 minutes ago)</td>
<td>Cardholder data, Protected Company Data, Windows</td>
<td>lucy</td>
<td>Multiple Hosts</td>
<td>Observed 2.25G bytes. Policy maximum allows up to 1k bytes.</td>
<td></td>
</tr>
</tbody>
</table>

Quick View for Flow

- User Name: lucy
- Host: 10.210.7.38
- Host Group(s): Cardholder data, Protected Company Data, Windows
- Country: RFC 1918
- MAC Address: 00:4da2:ad29:47 (Dell Inc.)

Service Summary: ftp (tcp/21)
Application: FTP (unclassified)

Active Duration: 29 minutes 21s
(Active for 29 minutes 21s)
(5 hours 4 minutes 21s ago) (4 hours 35 minutes 21s ago)

2.25G bytes (11.15M bps) in 2.47M packets (1.4k bps)

Port 21

Service Summary: ftp (tcp/21)
Application: FTP (unclassified)

First Port Seen: 16384

622.82M bytes (2.97M bps) in 1.29M packets (730.01 pps)

2.89G bytes (14.11M bps) in 3.75M packets (2.13k bps)

Domain: Lancope FlowCollector (10.192.0.133)
One of your users has uploaded a large amount of data to the internet.

### Beron’s abnormal disclosure

<table>
<thead>
<tr>
<th>Policy</th>
<th>Start Active Time</th>
<th>Alarm</th>
<th>Source</th>
<th>Source Host Group</th>
<th>Source User</th>
<th>Target Group</th>
<th>Target Hosts</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compliance Hosts</td>
<td>Mar 3, 2013 7:35:00 AM (1 day 1 hour ago)</td>
<td>Suspect Data Loss</td>
<td>10.210.7.38</td>
<td>Control Servers, Windows</td>
<td>lucy</td>
<td>Multiple Hosts</td>
<td>Observed 2.416G bytes. Policy maximum allows up to 1k bytes.</td>
<td></td>
</tr>
<tr>
<td>Inside Hosts</td>
<td>Mar 3, 2013 9:15:00 AM (23 hours 20 minutes 48s ago)</td>
<td>Suspect Data Loss</td>
<td>10.252.0.219</td>
<td>Desktops, Windows, Chicago</td>
<td>beron</td>
<td>Multiple Hosts</td>
<td>Observed 8.28G bytes. Policy maximum allows up to 500M bytes.</td>
<td></td>
</tr>
</tbody>
</table>
What did Beron send? Who received it?
Where could have Beron gotten the data?
Data Theft

Active Duration: 11 minutes 14s
(active for 11 minutes 14s)

(23 hours 42 minutes 24s ago) (23 hours 31 minutes 10s ago)

Client
- Host: 10.252.0.219
- Host Group(s): Desktops Windows Chicago
- Country: RFC 1918
- MAC Address: 00:08:a4:00:00:09 (Cisco Systems)
- Application Details: ...

Server
- Host: 10.252.0.10
- Host Group(s): BC Database Chicago
- Country: RFC 1918
- MAC Address: 00:05:dc:1d:10:00 (Cisco Systems, Inc.)
- SRT Average: 1 ms
- Application Details: ...

First Port Seen: 45823
- 130.3Mb/s (2.05kpps)
- 194.25M bytes (2.42M bps) in 3.92M pack
- Port 3306
- Service Summary: mysql (tcp/3306)
  Application: SQL
  15 TCP Connections

10.41G bytes (132.71M bps) in 5.9M packets (8.76k pps)
12 packets retransmitted (0%)
RTT: 1 ms

Domain: Lancope FlowCollector (10.192.0.133)
Why did Beron do it?

Control Channel

Data Theft
The Five W’s

• Who did this?
  – Usernames, IP Addresses

• What did they do?
  – What behavior did they engage in?

• Where did they go?
  – What hosts on my network were accessed?

• When?
  – Have we investigated the full intrusion timeline?

• Why? What is their objective?
• **Web**
  www.lancope.com (Company)

• **Twitter**
  @Lancope (Company)
  @netflowninjas (Company Blog)

**Tom Cross**
*Director of Research, Lancope*
 tcross@lancope.com

**Charles Herring**
*Sr. Systems Engineer, Lancope*
 cherring@lancope.com