Lab 7: Advanced Network Attack Analysis

A Evidence Bag

The malicious activity has been captured in the file:

https://dl.dropboxusercontent.com/u/40355863/newtrace.rar
Video: http://youtu.be/Sup4JUPlmgQ

B Data gathering

Open up the trace and find some of the IP addresses on the local network involved in the communications (look in Statistics-> Conversations) – you can find the local network by looking at the ARP activity (**arp**):

List a few of the MAC addresses of the hosts on the network, and their mapping to the IP addresses (look at **arp**):

Using the **tcp.flags.syn==1 && tcp. Flags.ack==0** filter, find all the connections that have been made. Now, using the Follow TCP stream option, determine some of the Web pages that each of the IP addresses on the network has access:

Using a filter of **udp.port==53**, determine some of the sites that each of the hosts have accessed:

What is the IP address of the gateway and what is its MAC address (look at the **arp** filter)?

C NMAP Evidence

In this investigation the malicious person has done some reconnaissance first, where the scanned the network for the hosts that are on-line, and then found the one they wanted, and scanned the TCP ports that are open.

One host does an ARP scan of the local network (use a filter of **arp**). Which computers are on-line, and which is the scanning node:

One host does an NMAP scan of a host. Which is the scanning host (you can search with **tcp.flags.syn==1** and scroll until you find lots of activity at the same time):

Which is the IP address of the host being scanned:

Which ports has the NMAP scan identified as being open (look for a SYN/ACK return in the scan):

Which type of NMAP scan has been used (a SYN flood or a Connect scan)? For a SYN flood there is no ACK from the client at the end, whereas for a Connect scan there will be.

D Hydra scan

After the intruder found some open ports, they looked for a vulnerability. For this they went after a username/password crack on the FTP server. There is thus a Hydra scan in the traffic, where an intruder has tried to crack a user name and password on one of the server ports.

With FTP, the server returns back some codes depending on whether the access is successful of not:

ftp.response.code==220 (for a login)
ftp.response.code==230 (for a successful login)
ftp.response.code==530 (for a unsuccessful login)

Can you locate the first signs of the Hydra attack (identify the packet numbers for the start and end of the Hydra attack)?

Outline a few of the names and passwords that have been tried:

Determine the successful login on FTP (with a user name and password):

E FTP access

After the intruder gained FTP access, they have then used the FTP server for malicious activities.

Which is the IP address of the malicious node:

How many times does this node connect the FTP server:

List the actions undertaken on the FTP servers. Which directories do the user go into, and what files do they upload/download?

There is a transfer of: STOR C60346DCA558858ABF74582F8E11D2E3.txt Identify how many characters this has. Is it a hash code? If so, can you crack it?

F Cross-correlating against host (FTP traffic)

Now we also have the host to examine.

In the Napier Cloud, go to your folder on vCenter and you should find a folder named **Production->evidence**. Find an instance which is powered-off and investigate it.

This machine has the evidence of the malicious activities on the disk.

Can you find the files that were transferred to the FTP server (look in c:\inetput\ftproot)? Which directories where created? Which is the content of the files (and can you view them)?

G Telnet login

There is quite a bit of Telnet activity. Find the activity with **tcp.port==23**. Which hosts are involved in the Telnet sessions:

Can you determine what the user did within the Telnet sessions: How many Telnet sessions are there? What folders did they go into? What files did they delete? What login name did the user use? How did they get this?

H Cross-correlating against host (Telnet traffic)

Now let's examine the host for the Telnet activity. In the Napier Cloud, go to your folder on vCenter and you should find an instance named **evidence**. This machine has the evidence of the malicious activities on the disk and cross-correlate the evidence found in the trace:

Can you find the activity on the host? What directories/files are created? Which files are moved? Can you find them?

I Emails login

In your network trace you can trace emails sent with **tcp.port==25** (for sending) and **tcp.port==110**.

Can you trace the emails sent? Outline the details of emails sent (sender, receiver and contents):

We can search for email addresses in an email message with:

smtp matches "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._%+-]"

Can you find all the email addresses in the traffic:

J Content searching

We can search for things using some magic numbers, such as:

http contains “%PDF” (PDF)
http contains "GIF89a" (GIF)
http contains "\x89\x50\x4E\x47" (PNG)
http contains "\x50\x4B\x30\x04" (ZIP)

Can you find the names of some GIF, PDF, PNG and ZIP files in HTTP traffic:

Using the list of magic numbers at: http://asecuritysite.com/forensics/magic

Can you search for JPEG files using the magic number for JPEGs:

Are there any Flash files in the trace:

K Snort analysis

We can also use Snort to analyse network traces, by using an off-line filtering system. For this you can run Snort with a rules file and with a trace:

snort -c 1.rules -l log -r newtrace.pcap

You can then look in the log filter for the log file and alert.ids.

Some rules you can use are given in Appendix A.

Now test Snort to see if it can detect the same content that you found before:

Number of Bad FTP logins:

Number of Successful FTP logins:

Number of GIF files in the trace:

Number of PNG files in the trace:

Can you detect the port scan on a host:

M Splunk

Using Splunk at **http:// 52.1.37.4:8000** determine the following. You will be allocated a login.

Requirement	Answer
What is the start date of the log	
What is the first username in the security log that gave an incorrect password (Hint: failed password | reverse)	
What is the first IP address in the security log that gave an incorrect password (Hint: failed password | reverse)	
Refer to the Splunk analysis. How many accesses were accessed by a "Chrome" browser and a "GET" method request (Hint - "chrome" AND method=GET)	
How many accesses were accessed by a "Chrome" browser or a "GET" method request (Hint - "chrome" OR method=GET)	
How many accesses were accessed by a "Chrome" browser and a "POST" method request (Hint - "chrome" OR method=POST)	
How many accesses were access by a "Chrome" browser or a "POST" method request (Hint - "chrome" OR method=POST)	
When was the peak accesses by a "Chrome" browser or a "POST" method request (Hint - "chrome" OR method=POST)	
How many accesses are there from a Safari browser (Hint: "safari"):	
How many accesses are there from a Chrome browser (Hint: "chrome"):	
How many accesses are there from a Mozilla browser (Hint: "mozilla"):	
On what day is there most activity in the secure logs (sourcetype=secure*):	
On Tuesday 11 March 2014, at 12:15pm, which IP generated a good deal of incorrect logins (sourcetype=secure*)	
For the access.log from www1, which is the most popular HTTP response value (Hint - source="c:\\sp\\www1\\access.log"| top limit=5 status)	
For the access.log from www1, which is the second most popular HTTP response value (Hint = source="c:\\sp\\www1\\access.log"| top limit=5 status)	
For the access.log from www1, which is the most popular IP address for accesses (Hint - source="c:\\sp\\www1\\access.log"| top limit=5 clientip)	
For the access.log from www1, which is the second most popular IP address for accesses (Hint - source="c:\\sp\\www1\\access.log"| top limit=5 clientip)	
For the access.log from www1, which is the most popular action (Hint - source="c:\\sp\\www1\\access.log"| top limit=5 action):	
Refer to the Splunk analysis. For the access.log from www1, which is the second most popular action (source="c:\\sp\\www1\\access.log"| top limit=5 action)	
For the access.log from www1, estimate the number of iPad accesses (Hint - source="c:\\sp\\www1\\access.log" ipad)	

For the access.log from www1, what is the top refer domain (Hint - source="c:\\sp\\www1\\access.log"| top limit=20 referer)	
Which is the first time for a refer from google.com (Hint - source="c:\\sp\\www1\\access.log" referer="http://www.google.com" | reverse)	
Which is the IP address of the client which is first referred from google.com (source="c:\\sp\\www1\\access.log" referer="http://www.google.com" | reverse)	
Which IP address connects successfully access signals.zip (Hint - signals.zip status=200)	
Which IP address does not try to access signals.zip (Hint - signals.zip)	
Refer to the Splunk analysis for secure*.log. How many failed password attempts where there from 194.8.74.23 (Hint - sourcetype=secure* 194.8.74.23 failed)	
Refer to the Splunk analysis for secure*.log. What day of the week had the most failed password attempts from 194.8.74.23 (Hint - sourcetype=secure* 194.8.74.23 failed)	
Refer to the Splunk analysis for access*.log. What day had the most successful purchases (Hint - action=purchase status=200)	
Refer to the Splunk analysis for access*.log. What day had the fewest purchases (Hint - action=purchase status=200)	
Refer to the Splunk analysis for access*.log. What day had the most purchases which were not successfully processed (Hint - action=purchase status!=200)	
Refer to the Splunk analysis for access*.log. How many STRATEGY games have been successfully purchased (Hint - categoryId=STRATEGY action=purchase status=200)	
Refer to the Splunk analysis for access*.log. Which file access always produces a 404 return message	
Refer to the Splunk analysis for access*.log. Which file access always produces a 404 return message: anna_nicole.html, productscreen.html, numa.html, cart.do or oldlink	
Refer to the Splunk analysis for access*.log. How many ARCADE games have been successfully purchased (Hint - categoryId=ARCADE action=purchase status=200)	
Refer to the Splunk analysis for access*.log. How many TEE games have been successfully purchased (Hint - categoryId=TEE action=purchase status=200)	
Refer to the Splunk analysis for access*.log. How many SIMULATION games have been successfully purchased (Hint - categoryId=TEE action=purchase status=200)	
Refer to the Splunk analysis for access*.log. How many SHOOTER games have been successfully purchased (Hint - categoryId=SHOOTER action=purchase status=200)	
Refer to the Splunk analysis for secure*.log. What day of the week had the least failed password attempts from 194.8.74.23 (Hint - failed password 194.8.74.23)	
Refer to the Splunk analysis for access*.log. For an HTTP GET request, which is the most popular return code [Hint - sourcetype="access*" method="GET"| top limit=20 status]	
Refer to the Splunk analysis for access*.log. For an HTTP GET request, which is the 2nd most popular return code [Hint - sourcetype="access*" method="GET"| top limit=20 status]	

M Snort analysis

Using some Snort analysis. Use the following PCAP files, and detect the activity:

- http://asecuritysite.com/log/hydra_ftp.zip. Detect bad FTP login. Detected? Yes/No
- http://asecuritysite.com/log/hydra_telnet.zip. Detect Telnet login. Detected? Yes/No
- http://asecuritysite.com/log/nmap.zip. Detect port scan. Detected? Yes/No
- http://asecuritysite.com/log/hping_syn.zip. Detect SYN flood. Detected? Yes/No
- http://asecuritysite.com/log/hping_syn.zip. Detect FIN flood. Detected? Yes/No
- http://asecuritysite.com/log/email_two_attachments.zip. Detect file attachments. Detected? Yes/No
- http://asecuritysite.com/log/email_cc2.zip. Detect credit card details and also email addresses. Detected? Yes/No
- http://asecuritysite.com/log/ping_sweep.zip. Detect ping sweep. Detected? Yes/No
- http://asecuritysite.com/log/with_pdf.zip. PDF files. Detected? Yes/No

Appendix

Bad logins:

```
alert tcp any 21 -> any any (msg:"FTP Bad login"; content:"530 User "; nocase; flow:from_server,established; sid:491;
rev:5;)
```

Detecting email addresses:

```
alert tcp any any <> any 25 (pcre:"/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._%+-]/"; \
msg:"Email in message";sid:9000000;rev:1;)
```

Detect DNS:

```
alert udp any any -> any 53 (msg: "DNS"; sid:10000;)
```

File types:

```
alert tcp any any -> any any (content:"GIF89a"; msg:"GIF";sid:10000)
alert tcp any any -> any any (content:"%PDF"; msg:"PDF";sid:10001)
alert tcp any any -> any any (content:"|89 50 4E 47|"; msg:"PNG";sid:10002)
alert tcp any any -> any any (content:"|50 4B 03 04|"; msg:"ZIP";sid:10003)
```

Telnet login:

```
alert tcp any any <> any 23 (flags:S; msg:"Telnet Login";sid:9000005;rev:1;)
```

Port scan:

```
preprocessor sfportscan:\
        proto { all } \
        scan_type { all } \
        sense_level { high } \
        logfile { portscan.log }
```

DoS on Web server:

```
alert tcp any any -> any 80 (msg:"DOS flood denial of service attempt";flow:to_server; \
detection_filter:track by_dst,  count 60, seconds 60; \
sid:25101; rev:1;)
```

Stealth scans:

```
alert tcp any any -> any any (msg:"SYN FIN Scan"; flags: SF;sid:9000000;)
alert tcp any any -> any any (msg:"FIN Scan"; flags: F;sid:9000001;)
alert tcp any any -> any any (msg:"NULL Scan"; flags: 0;sid:9000002;)
alert tcp any any -> any any (msg:"XMAS Scan"; flags: FPU;sid:9000003;)
alert tcp any any -> any any (msg:"Full XMAS Scan"; flags: SRAFPU;sid:9000004;)
alert tcp any any -> any any (msg:"URG Scan"; flags: U;sid:9000005;)
alert tcp any any -> any any (msg:"URG FIN Scan"; flags: FU;sid:9000006;)
alert tcp any any -> any any (msg:"PUSH FIN Scan"; flags: FP;sid:9000007;)
alert tcp any any -> any any (msg:"URG PUSH Scan"; flags: PU;sid:9000008;)
alert tcp any any -> any any (flags: A; ack: 0; msg:"NMAP TCP ping!";sid:9000009;)
```

ping sweep:

```
alert icmp any any -> any any (msg:"ICMP Packet found";sid:9000000;)
alert icmp any any -> any any (itype: 0; msg: "ICMP Echo Reply";sid:9000001;)
alert icmp any any -> any any (itype: 3; msg: "ICMP Destination Unreachable";sid:9000002;)
alert icmp any any -> any any (itype: 4; msg: "ICMP Source Quench Message received";sid:9000003;)
alert icmp any any -> any any (itype: 5; msg: "ICMP Redirect message";sid:9000004;)
alert icmp any any -> any any (itype: 8; msg: "ICMP Echo Request";sid:9000005;)
alert icmp any any -> any any (itype: 11; msg: "ICMP Time Exceeded";sid:9000006;)
```

Note you may have to add the following for the stream analysis:

```
preprocessor stream5_global: track_tcp yes, \
   track_udp yes, \
   track_icmp no, \
   max_tcp 262144, \
   max_udp 131072, \
   max_active_responses 2, \
   min_response_seconds 5
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \
   overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
    ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137 139 143 \
        161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 \
        7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \
    ports both 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465 563 591 593 631 636 901 989 992 993 994 995 1220 1414 1830 2301 2381 2809 \
       3037 3057 3128 3443 3702 4343 4848 5250 6080 6988 7907 7000 7001 7144 7145 7510 7802 7777 7779 \
       7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914 7915 7916 \
        7917 7918 7919 7920 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8222 8243 8280 8300 8500 8800 8888 8899 9000 9060 9080 9090 \
        9091 9443 9999 10000 11371 34443 34444 41080 50000 50002 55555
preprocessor stream5_udp: timeout 180
```