TryHackMe | Zeek — Write-up

igor_sec · Follow
12 min read · Jul 12, 2023

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

Link: https://tryhackme.com/room/zeekbro

Zeek (formerly Bro) is an open-source and commercial network monitoring tool (traffic analyser).

The official description; "Zeek (formerly Bro) is the world's leading platform for network security monitoring. Flexible, open-source, and powered by defenders." "Zeek is a passive, open-source network traffic analyser. Many operators use Zeek as a network security monitor (NSM) to support suspicious or malicious activity investigations. Zeek also supports a wide range of traffic analysis tasks beyond the security domain, including performance measurement and troubleshooting."

The room aims to provide a general network monitoring overview and work with Zeek to investigate captured traffic. This room will expect you to have basic Linux familiarity and Network fundamentals (ports, protocols and traffic data). We suggest completing the "Network Fundamentals" path before starting working in this room.

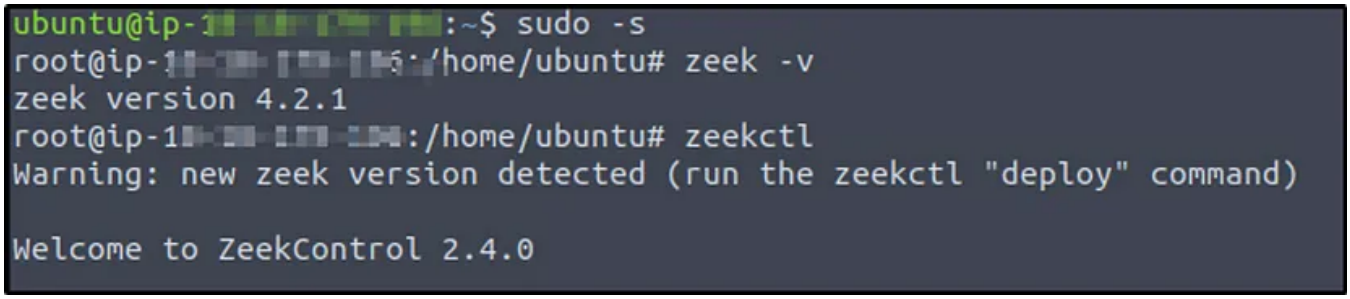
> *Task 2: Network Security Monitoring and Zeek*

Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-2

What is the installed Zeek instance version number?

Ans: 4.2.1

```
zeek -v
```



What is the version of the ZeekControl module?

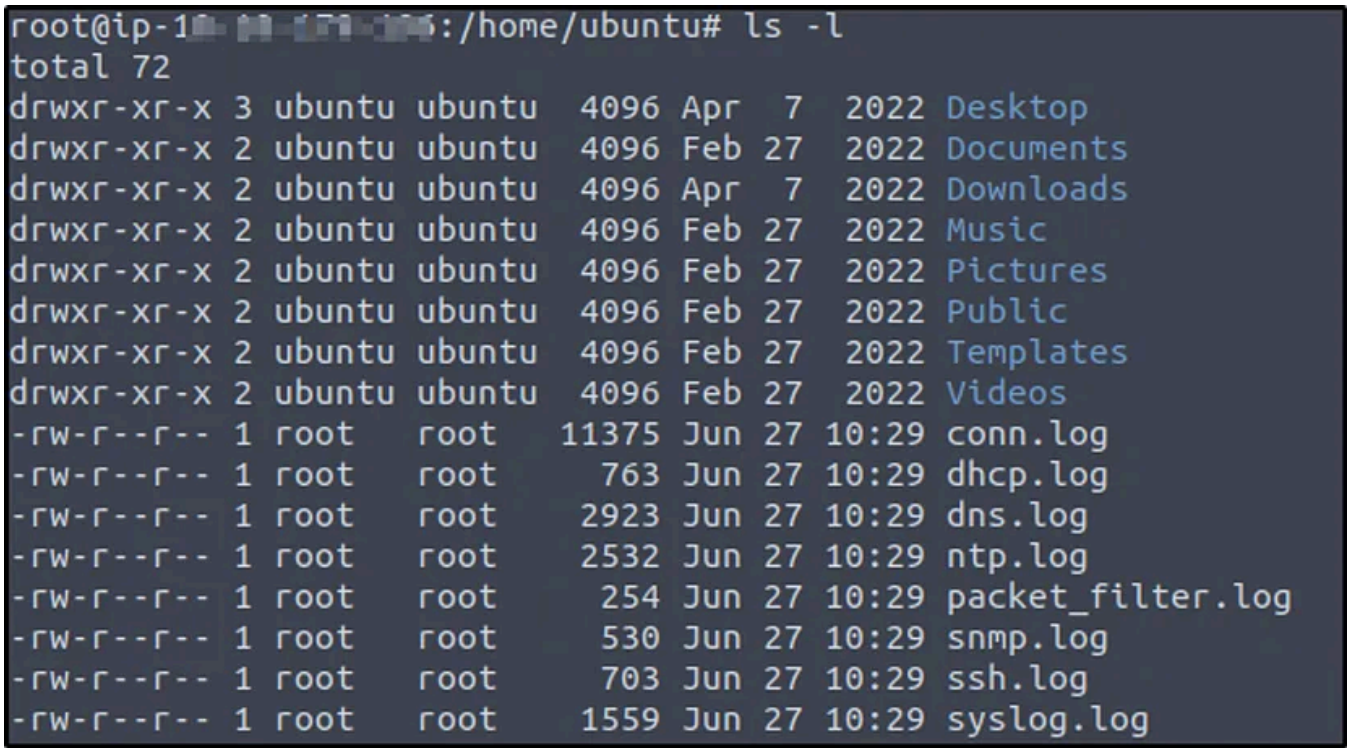
Ans: 2.4.0

We got the answer from executing the command from the previous question.

Investigate the “sample.pcap” file. What is the number of generated alert files?

Ans: 8

```
zeek -C -r Desktop/Exercise-Files/TASK-2/sample.pcap
```

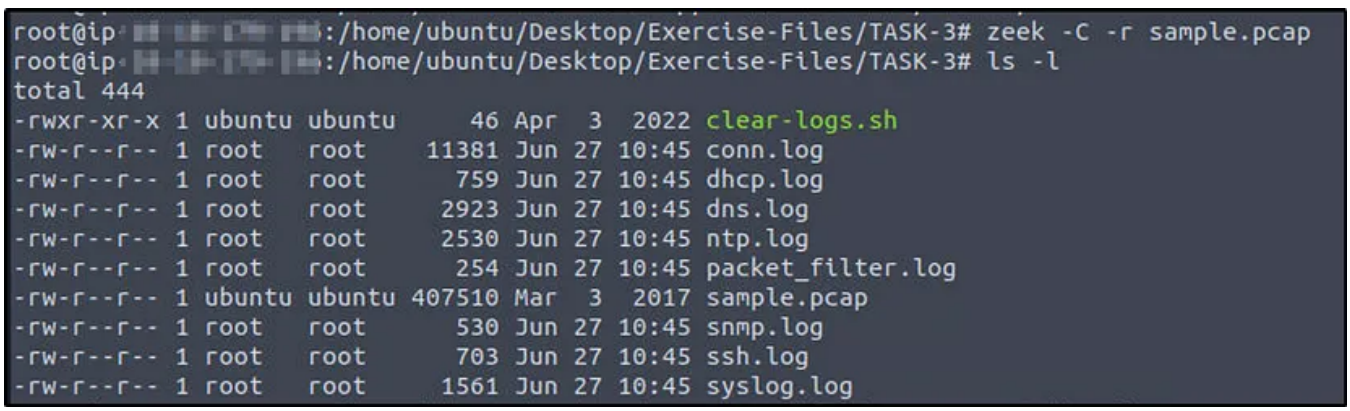


Task 3: Zeek Logs

Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-3

```
zeek -C -r sample.pcap
```

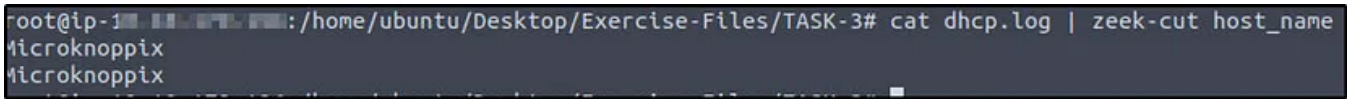
```
ls -l
```



Investigate the sample.pcap file. Investigate the dhcp.log file. What is the available hostname?

Ans: Microknoppix

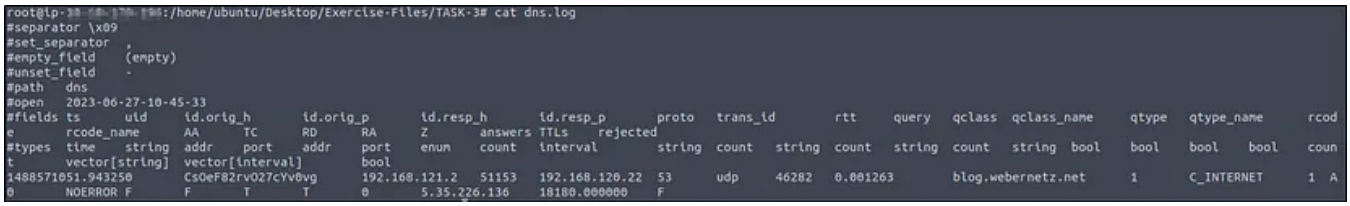
```
cat dhcp.log
cat dhcp.log | zeek-cut host_name
```



Investigate the dns.log file. What is the number of unique DNS queries?

Ans: 2

```
cat dns.log
cat dns.log  | zeek-cut query
```



```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/TASK-3# cat dns.log  | zeek-cut query
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
blog.webernetz.net
ip.webernetz.net
```

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

Ans: 332.319364

```
cat conn.log
```

We will use sort from the highest (-r) according to string numerical values (-n) then pipe the result with head command to show the first value.

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

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/TASK-3# cat conn.log | zeek-cut duration | sort -n -r | head -n 1
332.319364
```

Task 5: Zeek Signatures

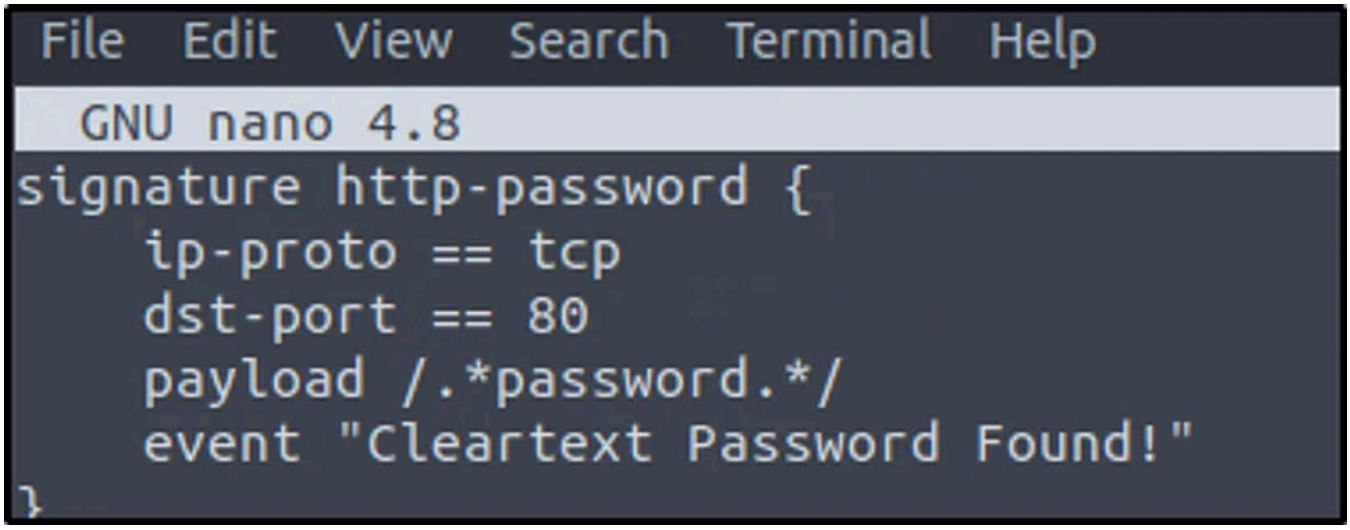
Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-5

Investigate the http.pcap file. Create the HTTP signature shown in the task and investigate the pcap. What is the source IP of the first event?

Ans: 10.10.57.178

Let's create the signature.

```
nano http-password.sig
```

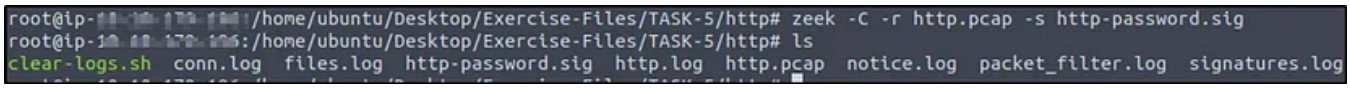


The signature will match when a "password" phrase is detected in the packet payload.

Let's read the pcap file with the signature created.

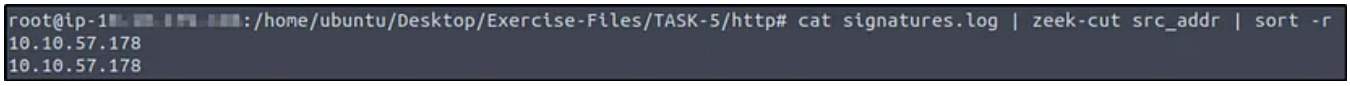
```
zeek -C -r http.pcap -s http.password.sig
```

An alert will be generated and two log files will be created, "signatures.log", and "notice.log"



From the "signatures/log", we will use "zeek-cut" to select the field "src_addr", then sort it in reverse. This will give use the first source IP address.

```
cat signatures.log | zeek-cut src_addr | sort -r
```

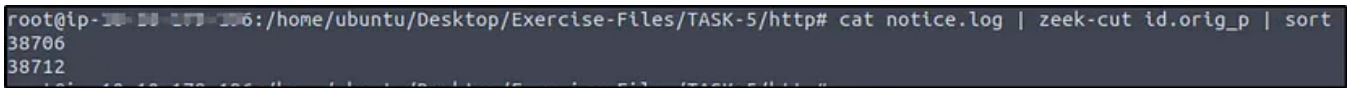


What is the source port of the second event?

Ans: 38712

We will modify the command above to select the field name “src_port”. The source port numbers will be displayed.

```
cat signatures.log | zeek-cut src_port | sort
```



Investigate the conn.log.
What is the total number of the sent and received packets from source port 38706?

Ans: 20

We will select the field names “id.orig_p” and “orig_pkts resp_pkts” and grep the source port. The two values are then added to get the total number of bytes sent and received from the source port.

```
cat conn.log
```

```
cat conn.log |zeek-cut id.orig_p orig_pkts resp_pkts | grep 38706
```

```
root@ip-1[redacted]:/home/ubuntu/Desktop/Exercise-Files/TASK-5/http# cat conn.log |zeek-cut id.orig_p orig_pkts resp_pkts | grep 38706
38706   11      9
```

Create the global rule shown in the task and investigate the ftp.pcap file.

Let's create first the global rule. The rule will match any payload when an FTP username is used to authenticate and any payload attempting to brute-force the FTP server.

```
nano ftp-bruteforce.sig
```

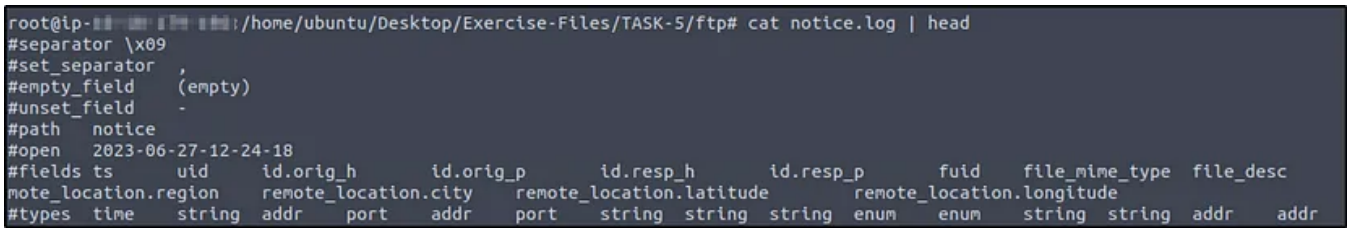
```
  GNU nano 4.8
signature ftp-username {
    ip-proto == tcp
    ftp /.*USER.*/
    event "FTP Username Input Found!"
}

signature ftp-brute {
    ip-proto == tcp
     payload /.*530.*Login.*incorrect.*/
    event "FTP Brute-force Attempt!"
}
```

Let's now read the pcap file with the signature applied.

```
zeek -C -r ftp.pcap -s ftp-brutegorce.sig
```

Similarly, an alert will be generated and two log files will be created.

```
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/TASK-5/ftp# zeek -C -r ftp.pcap -s ftp-bruteforce.sig
root@ip-[redacted]:/home/ubuntu/Desktop/Exercise-Files/TASK-5/ftp# ls
clear-logs.sh  conn.log  ftp-bruteforce.sig  ftp.pcap  notice.log  packet_filter.log  signatures.log  weird.log
root@ip-10-10-179-196:/home/ubuntu/Desktop/Exercise-Files/TASK-5/ftp#
```

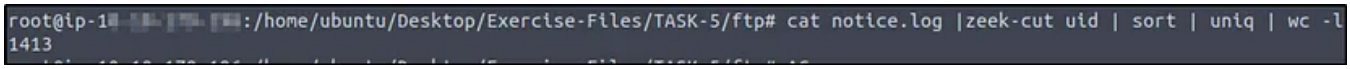


Investigate the notice.log. What is the number of unique events?

Ans:1413

We will be counting the number of unique events based off from the “uid” field.

```
cat notice.log |zeek-cut uid | sort | uniq | wc -l
```



What is the number of ftp-brute signature matches?

Ans: 1410

If you will notice, I always read, “cat”, the log files. That is simply to identify the field names correctly.

```
cat signatures.log | head -n 20
```

This time, we will investigate the “signatures.log”. From the rule we created, if there was a brute-force attempt, it will create an event, “FTP Brute-force Attempt!”, which is logged in “signatures.log”. This event will be logged as “ftp-brute”.

Notice that in the field name “sig_id”, it contains the event we are after. So we will select that field and the count the lines generated.

```
cat signatures.log | zeek-cut sig_id | grep "ftp-brute" | wc -l
```

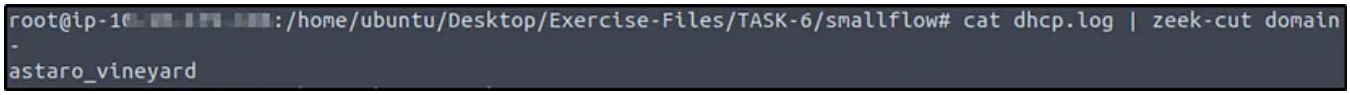
Task 6: Zeek Scripts | Fundamentals

Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-6

Investigate the smallFlows.pcap file. Investigate the dhcp.log file. What is the domain value of the “vinlap01” host?

Ans: astaro_vineyard

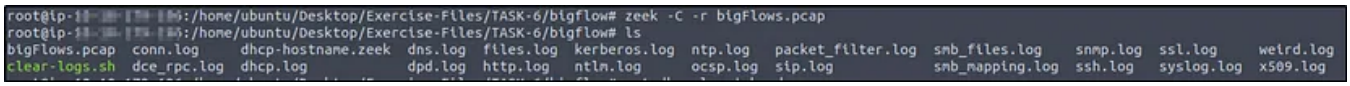
```
zeek -C -r smallFlows.pcap
cat dhcp.log
cat dhcp.log | zeek-cut domain
```

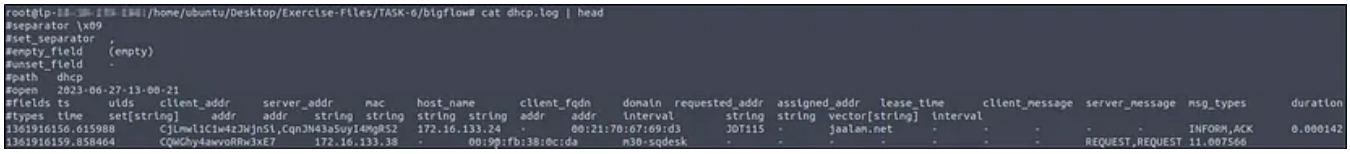


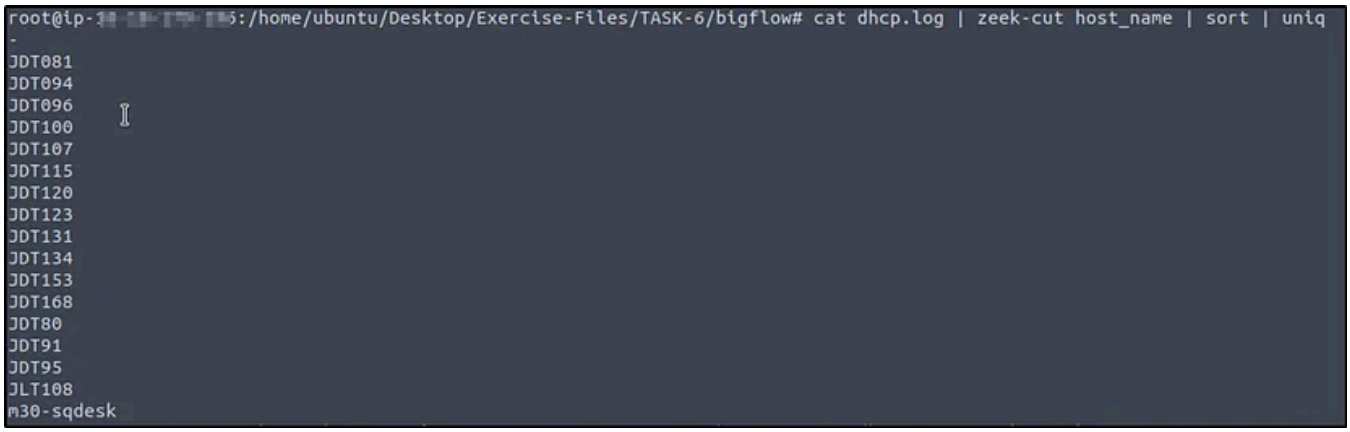
Investigate the bigFlows.pcap file. Investigate the dhcp.log file. What is the number of identified unique hostnames?

Ans: 17

```
zeek -C -r bigFlows.pcap
cat dhcp.log | zeek-cut host_name | sort | uniq
cat dhcp.log | zeek-cut host_name | sort | uniq | wc -l
```





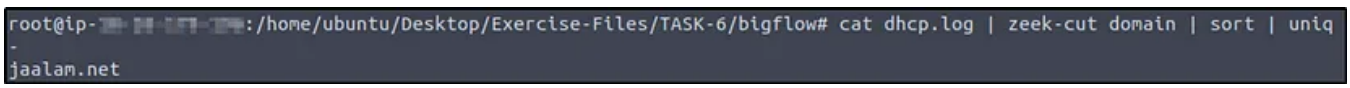


When we include the command to count the lines, the result is 18. But in the image above the first line is empty so the correct answer is 17.

Investigate the dhcp.log file. What is the identified domain value?

Ans: jaalam.net

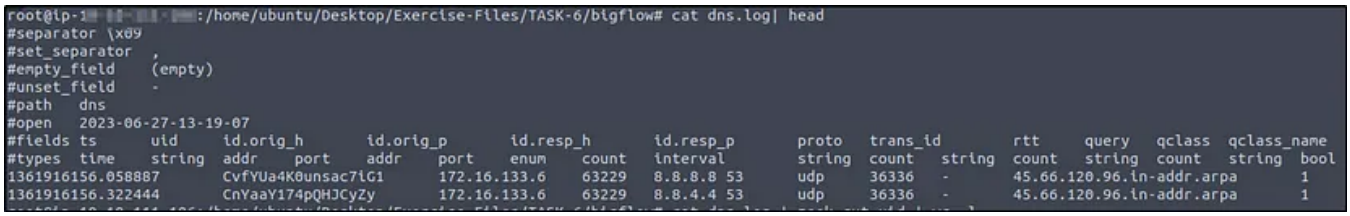
```
cat dhcp.log | zeek-cut domain | sort | uniq
```



Investigate the dns.log file. What is the number of unique queries?

Ans: 1109

```
cat dns.log| head
```



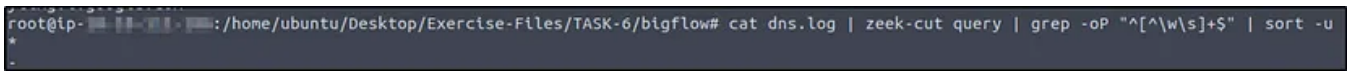
Recall from CLI Kung-fu the command `grep -v -e 'test1' -e 'test2'`, which display lines that don't match one or both "test1" and "test2" strings. The hint also provided us this, "grep -v -e '*' -e '-' ".

The hint says that there are two values that we should not include in our result. But what if in other situations there could be more values?

So to determine what are the special characters from a file or logs, we can use this command. We will read from the "dns.log" for example.

```
cat dns.log | zeek-cut query | grep -oP "^[^\w\s]+$" | sort -u
```

This command includes a grep command that only output (-o) the matched special character and '-P' option to enable Perl-compatible regular expresssions. The regex expression matches any individual special character. "sort -u" sorts the output in alphabetical order (sort) and '-u' option ensures that duplicate characters are removed, so each special character appears only once in the output.



So we know now what special characters not to include in our output.

```
cat dns.log | zeek-cut query |grep -v -e '*' -e '-' | sort | uniq| wc -l
```

Task 7: Zeek Scripts | Scripts and Signatures

Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-7

Go to folder TASK-7/101.
Investigate the sample.pcap file with 103.zeek script. Investigate the terminal output. What is the number of the detected new connections?

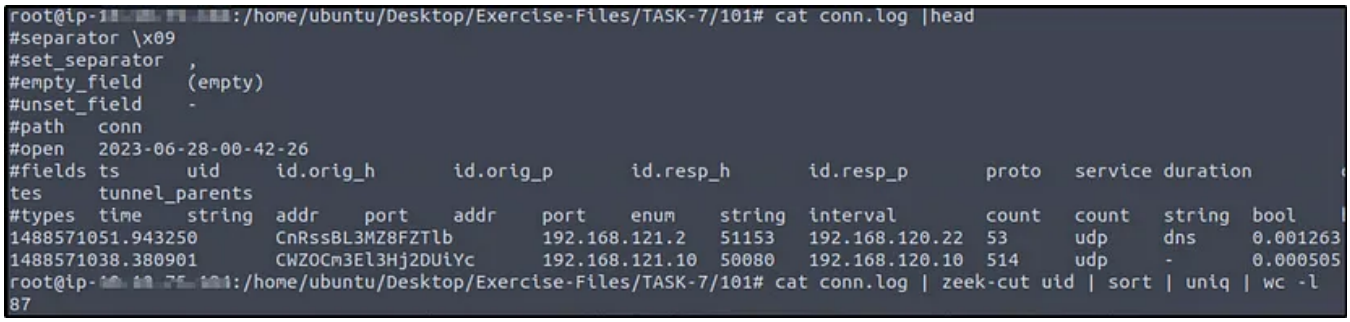
Ans: 87

let's run the sample.pcap with the script.

```
zeek -C -r sample.pcap 103.zeek
```

We will "cat" the "conn.log" then select the "uid" field, sort the results, and pipe with "uniq" to avoid duplication, and then finally count the lines.

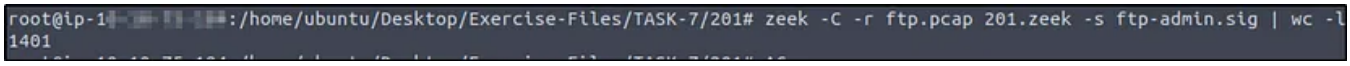
```
cat conn.log | zeek-cut uid | sort | uniq | wc -l
```



Go to folder TASK-7/201.
Investigate the ftp.pcap file with ftp-admin.sig signature and 201.zeek script. Investigate the signatures.log file. What is the number of signature hits?

Ans: 1401

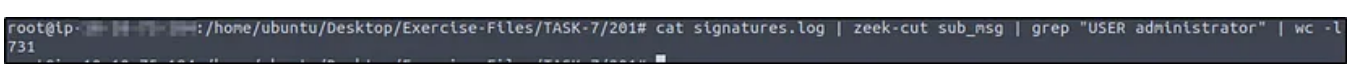
```
zeek -C -r ftp.pcap 201.zeek -s ftp-admin.sig | wc -l
```



Investigate the signatures.log file. What is the total number of "administrator" username detections?

Ans: 731

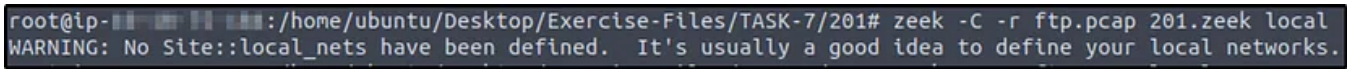
```
cat signatures.log | zeek-cut sub_msg | grep "USER administrator" | wc -l
```



Investigate the ftp.pcap file with all local scripts, and investigate the loaded_scripts.log file. What is the total number of loaded scripts?

Ans: 498

```
zeek -C -r ftp.pcap 201.zeek local
```



Don't worry if you get a warning.

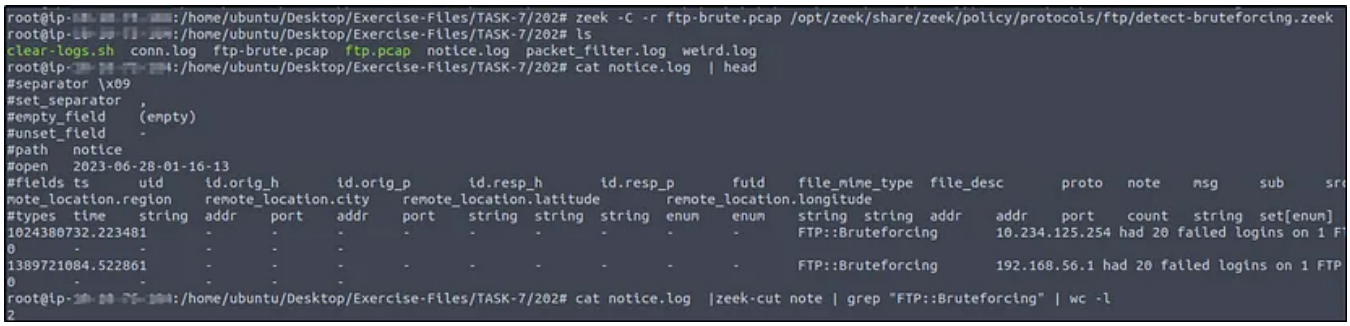
```
cat loaded_scripts.log | zeek-cut name | wc -l
```

Go to folder TASK-7/202.
Investigate the ftp-brute.pcap file with "/opt/zeek/share/zeek/policy/protocols/ftp/detect-bruteforcing.zeek" script.
Investigate the notice.log file. What is the total number of brute-force detections?

Ans: 2

```
zeek -C -r ftp-brute.pcap /opt/zeek/share/zeek/policy/protocols/ftp/detect-brut
```

```
cat notice.log  |zeek-cut note | grep "FTP::Bruteforcing" | wc -l
```



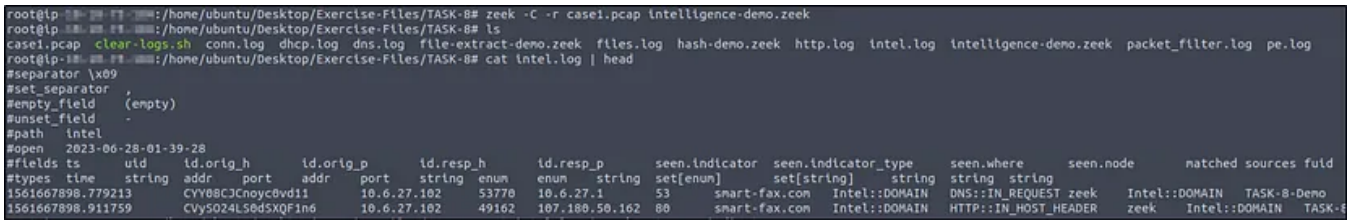
> *Task 8: Zeek Scripts | Frameworks*

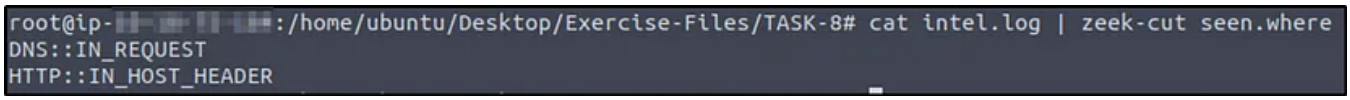
Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-8

Investigate the case1.pcap file with intelligence-demo.zeek script. Investigate the intel.log file. Look at the second finding, where was the intel info found?

Ans: IN_HOST_HEADER

```
zeek -C -r case1.pcap intelligence-demo.zeek
cat intel.log | head
cat intel.log | zeek-cut seen.where
```



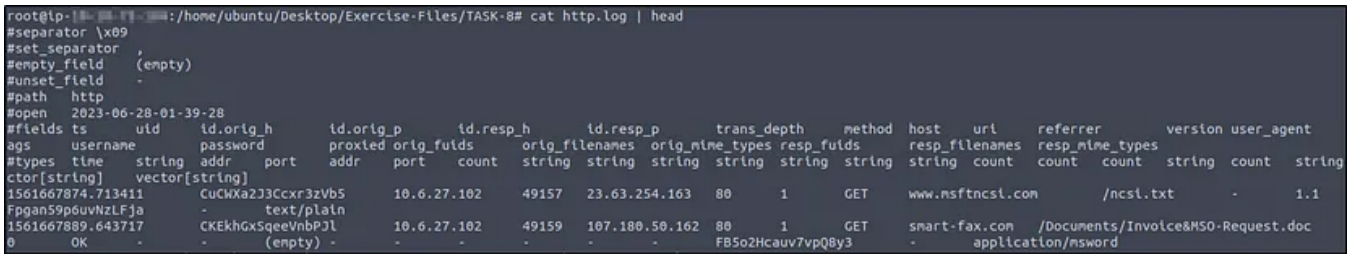


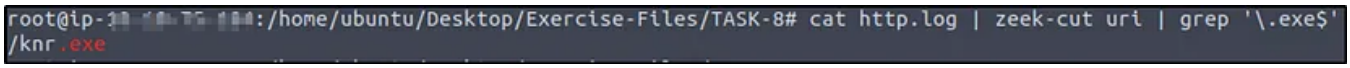
Investigate the http.log file. What is the name of the downloaded .exe file?

Ans: knr.exe

```
cat intel.log | head
cat http.log | zeek-cut uri | grep '\.exe$'
```

“grep ‘\.exe$’” searches for lines that contain the “.exe” extension at the end of the line. The backslash (\) before the dot (.) is used to escape it, so that it matches a literal dot. The “$” matches the end of the line.

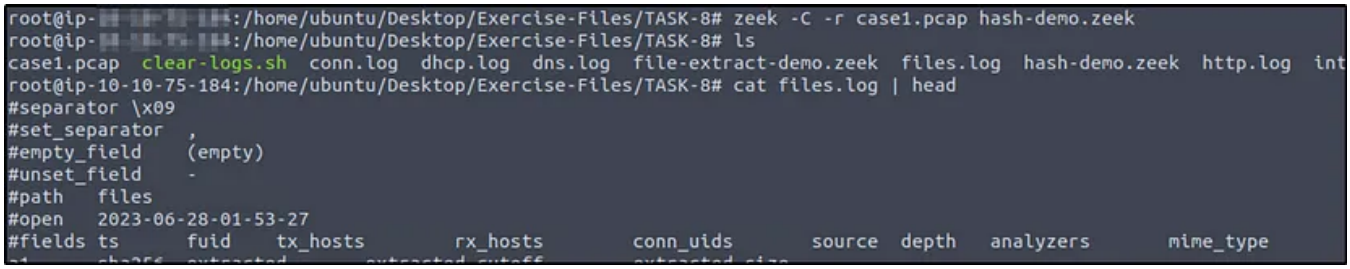




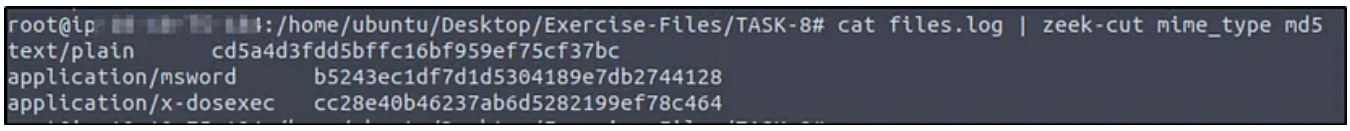
Investigate the case1.pcap file with hash-demo.zeek script. Investigate the files.log file. What is the MD5 hash of the downloaded .exe file?

Ans: cc28e40b46237ab6d5282199ef78c464

```
zeek -C -r case1.pcap hash-demo.zeek
cat files.log | head
cat files.log | zeek-cut mime_type md5
```



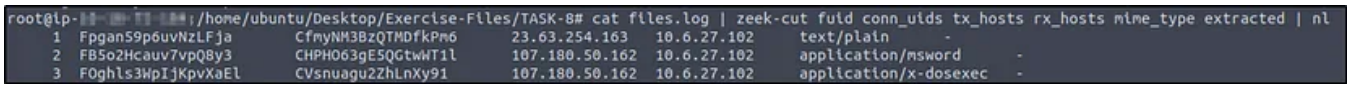
We know that it is an executable file so it should be the third md5 value.



We can also find the correlation of the “.exe” file with the other log files.

First we need a common value of the “.exe” file.

```
cat files.log | zeek-cut fuid conn_uids tx_hosts rx_hosts mime_type extracted |
```



We will choose the value of the field “conn_uids”.

So we got the “conn_uids” value of the “.exe” file. Now we will extract all values in the current directory that correlates to the “.exe” file.

```
grep -rin CVsnuagu2ZhLnXy91 * | column -t | nl | less -S
```

- `grep -rin CVsnuagu2ZhLnXy91 *` : This searches for the pattern "CVsnuagu2ZhLnXy91" recursively (`-r`) in all files (`*`) within the current directory. The `-i` option is used for case-insensitive matching, and the `-n` option displays line numbers.
- `column -t` : This formats the output into multiple columns for better readability. It assumes tabular data with whitespace as the delimiter.
- `nl` : This adds line numbers to the output.
- `less -S` : This command opens the output in the `less` pager, which allows scrolling through the content. The `-S` option disables line wrapping for better readability.

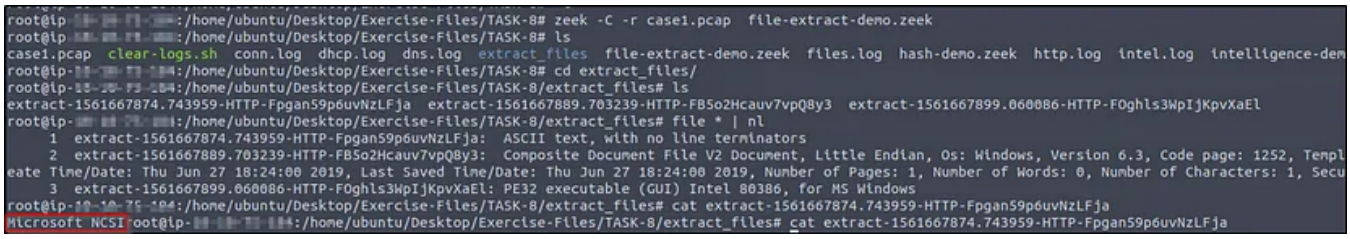
From the result, the “.exe” file is found in three logs. We see other information that relates to the file, and in “files.log” we see its MD5 value.

Investigate the case1.pcap file with file-extract-demo.zeek script. Investigate the “extract_files” folder. Review the contents of the text file. What is written in the file?

Ans: Microsoft NCSI

```
zeek -C -r case1.pcap  file-extract-demo.zeek
file * | nl
```

```
cat extract-1561667874.743959-HTTP-Fpgan59p6uvNzLFja
```



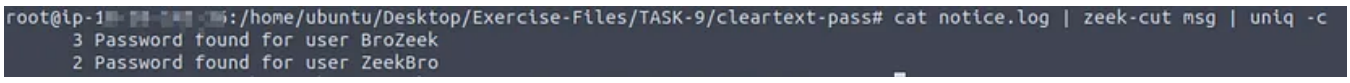
Task 9: Zeek Scripts | Packages

Each exercise has a folder. Ensure you are in the right directory to find the pcap file and accompanying files. Desktop/Exercise-Files/TASK-9

Investigate the http.pcap file with the zeek-sniffpass module. Investigate the notice.log file. Which username has more module hits?

Ans: BroZeek

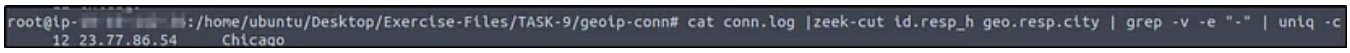
```
zeek -C -r http.pcap zeek-sniffpass
cat notice.log | head
cat notice.log | zeek-cut msg | uniq -c
# "-c"count the number of occurences for each unique value
```



Investigate the case2.pcap file with geoip-conn module. Investigate the conn.log file. What is the name of the identified City?

Ans: Chicago

```
zeek -C -r case2.pcap  geoip-conn
cat conn.log |head
cat conn.log |zeek-cut id.resp_h geo.resp.city | grep -v -e "-" | uniq -c
```



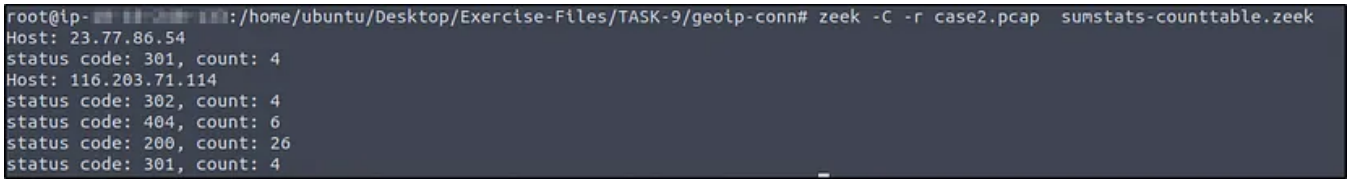
Which IP address is associated with the identified City?

Ans: 23.77.86.54

Investigate the case2.pcap file with sumstats-counttable.zeek script. How many types of status codes are there in the given traffic capture?

Ans: 4

```
zeek -C -r case2.pcap  sumstats-counttable.zeek
```



Here is a modified command.

```
zeek -C -r case2.pcap sumstats-counttable.zeek | awk '{print $3}' | grep -v -e
```

In this command, the `-v` option inverts the matching logic, causing `grep` to exclude lines that match the specified pattern. The pattern `^$` matches empty lines because `^` represents the start of a line, and `$` represents the end of a line. Thus, `^$` matches lines that contain no characters between the start and end.

Thanks for Reading.

Happy learning.

My write-up for the next Zeek Exercise is here: (coming soon)

Tryhackme | Ctf Writeup | Cybersecurity | Training | Learning

Follow

Written by igor_sec

370 Followers · 11 Following

Responses (1)

What are your thoughts?

Respond

Samar
about 2 months ago

thank you

Reply

More from igor_sec

igor_sec

CyberDefenders | Boss Of The SOC v1

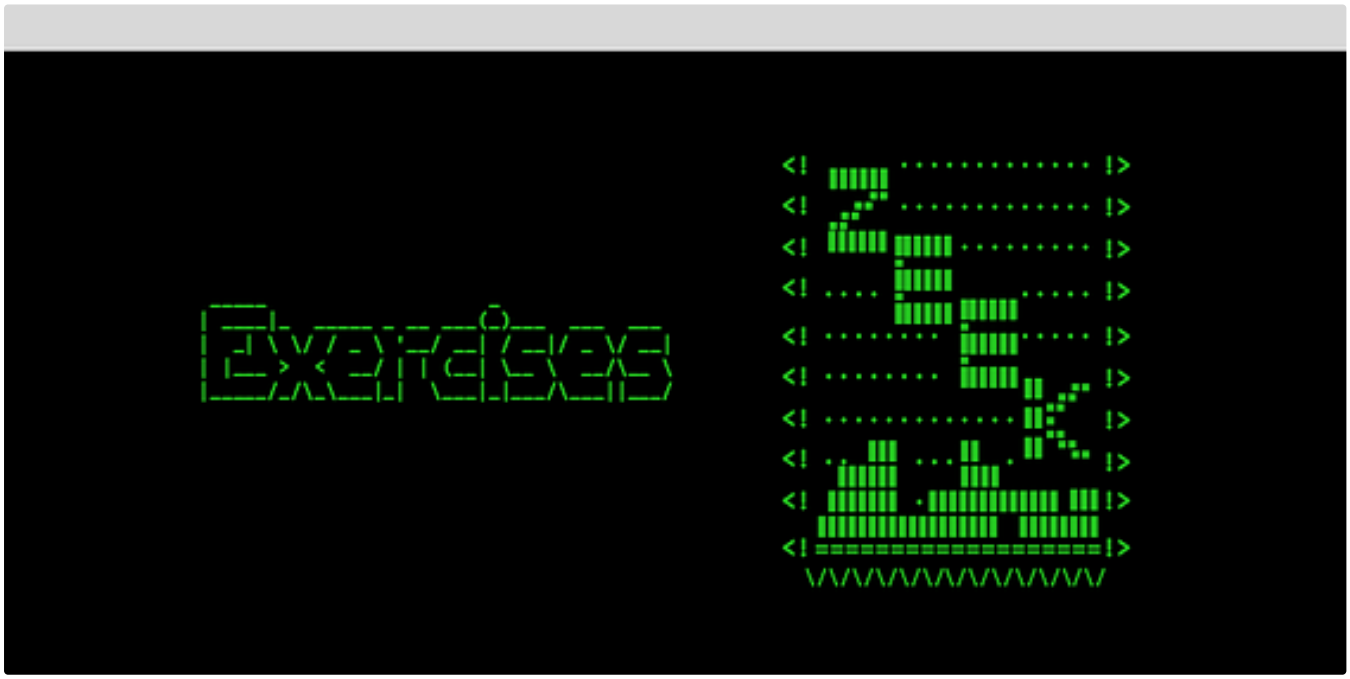
Jul 5, 2023 12

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

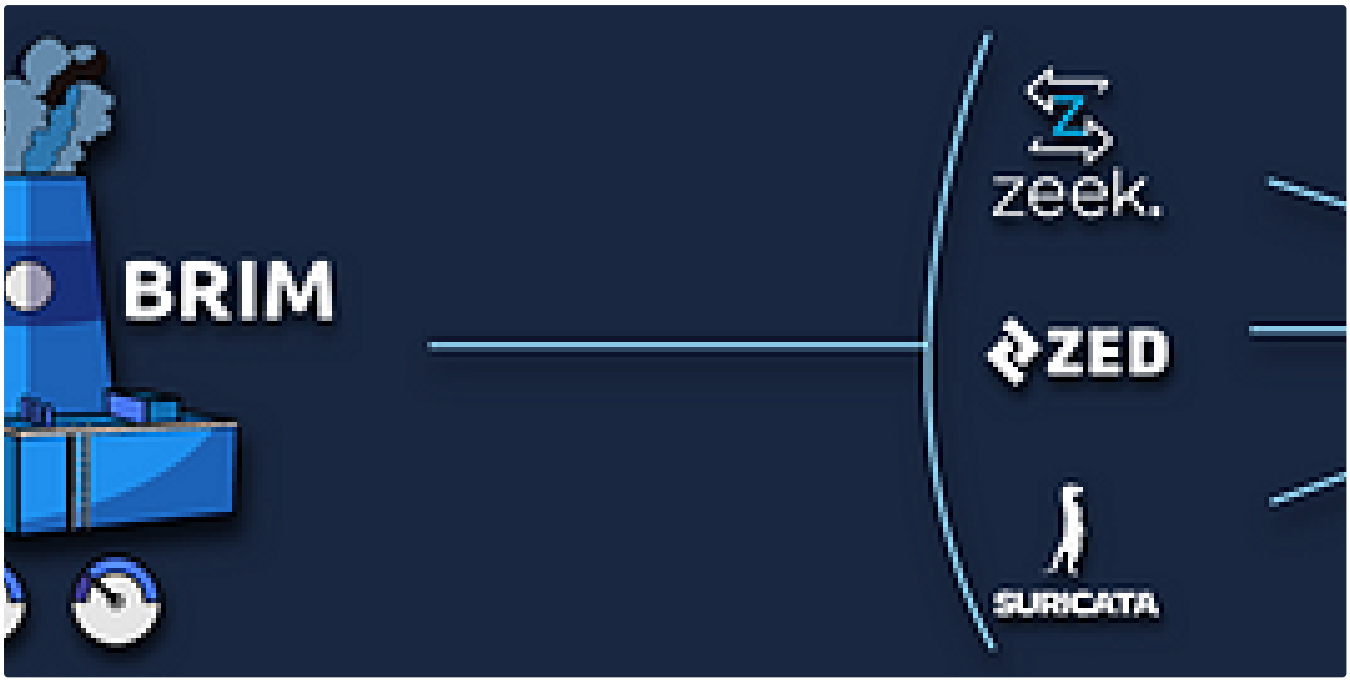


igor_sec

TryHackMe | Zeek Exercises — Write-up

Put your Zeek skills into practice and analyse network traffic.

Jul 12, 2023 38 1



igor_sec

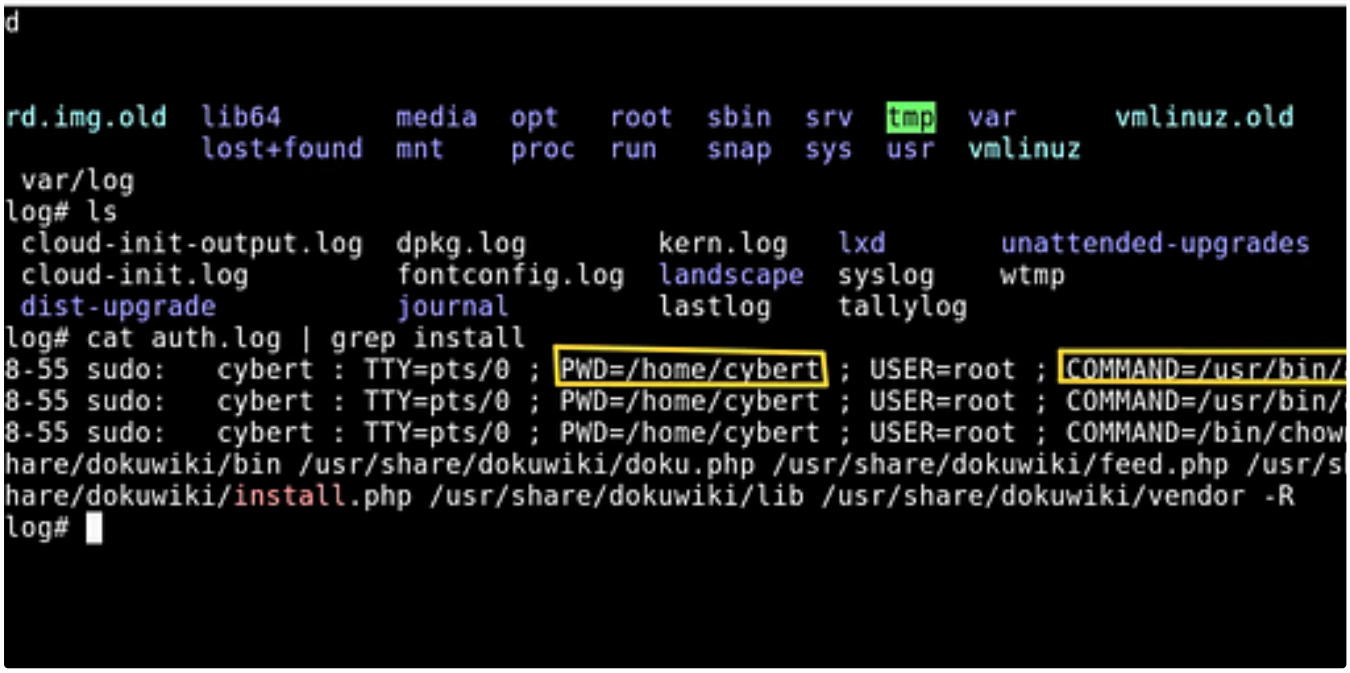
TryHackMe | Brim

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

Jul 1, 2023 3 1

See all from igor_sec

Recommended from Medium

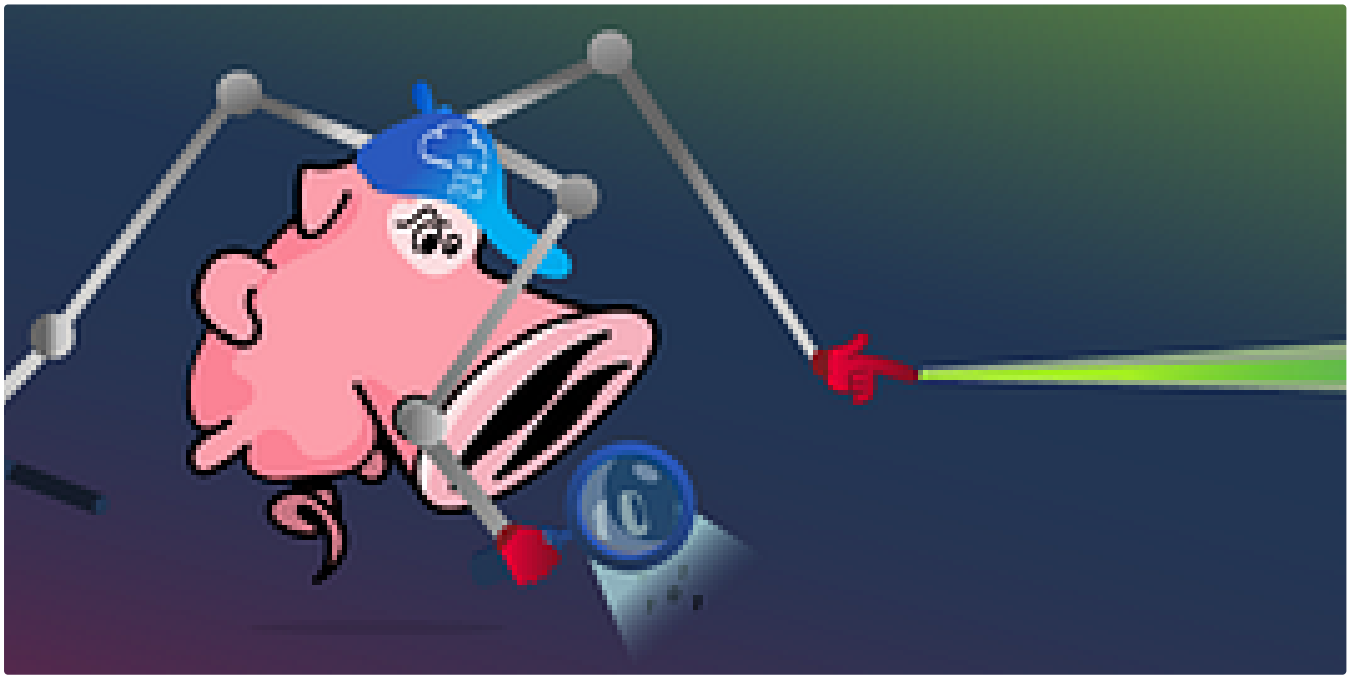


Dan Molina

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



In T3CH by Axoloth

TryHackMe | Snort Challenge—The Basics | WriteUp

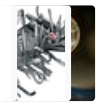
Put your snort skills into practice and write snort rules to analyse live capture network traffic

Nov 9, 2024 100

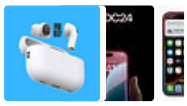
Lists

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

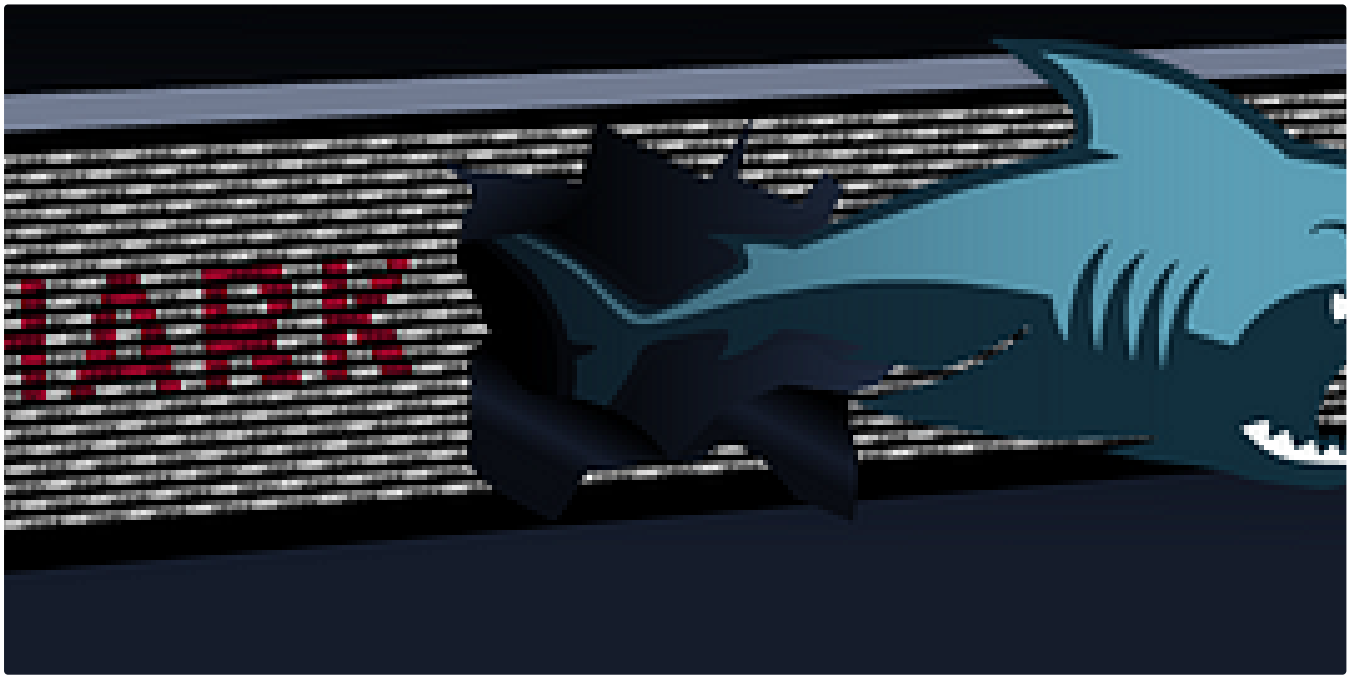
Nov 28, 2024 50

In T3CH by Axoloth

TryHackMe | Training Impact on Teams | WriteUp

Discover the impact of training on teams and organisations

Nov 5, 2024 60



MAGESH

TShark: The Basics—Tryhackme

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

Sep 3, 2024

DevSecOps

[Wi-Fi attacks] Day 11: If you’d like to WPA, press the star key!

[Wi-Fi attacks] Day 11: If you’d like to WPA, press the star key! [TryHackMe THM][Advent of Cyber AoC 2024], [Walktgrough, Write Up]

Dec 12, 2024 1

See more recommendations