

PSP0201

Week 3

Writeup

Group Name: ikun no 1

Members

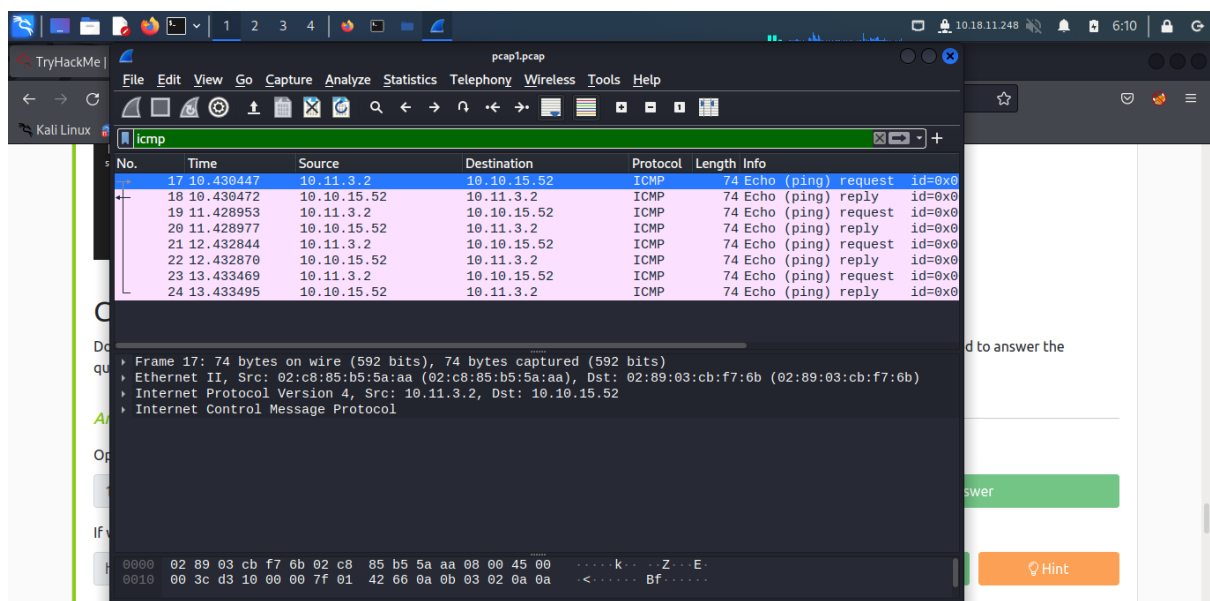
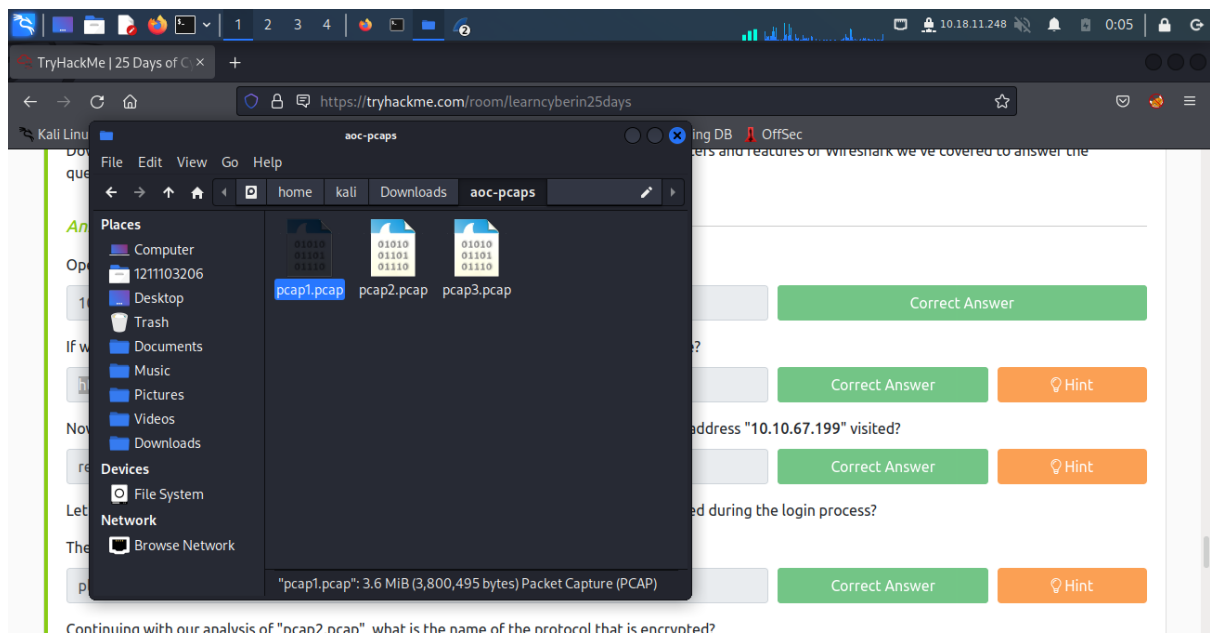
ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

Day 7: Networking The Grinch Really Did Steal Christmas

Tool used: Kali-Linux, WireShark

Solution/walkthrough:

Question 1



Download the zip file that has given and unzip it. Open pcap1 and type icmp and the ip address shown.

Question 2

TryHackMe | 25 Days of C x

https://tryhackme.com/room/learnycyberin25days

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Red Hat Linux / Fedora Packages

Waiting for ntp.msn.com...

Networks are, however, rather noisy...Wireshark captured 2,648 packets after a single minute on my machine. This makes analysing very hard. Thankfully, we can use filters to narrow down the results. We can filter by many things, but we'll only cover a couple of important ones in the table below. Note that all the examples below use the `==` operator to see if the filter exactly matches the value we give it.

Filter	Description	Example
<code>ip.src</code>	Show all packets that originate from the specified IP address	<code>ip.src == 192.168.1.1</code>
<code>ip.dst</code>	Show all packets that are destined to the specified IP address	<code>ip.dst == 192.168.1.1</code>
<code>tcp/udp.port</code>	Show all packets that are sent via the protocol and port specified	<code>tcp.port == 22 / udp.port == 67</code>
<code>protocol.request.method</code>	Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a <code>GET</code> and <code>POST</code> to retrieve and submit data accordingly.	<code>http.request.method == GET / POST</code>

In the screenshot below, I used the filter `ip.src` to list all the packets that were explicitly sent from a specific address, using the `==` operator to define what host I wish to search for (`145.254.160.237`). We'll quickly explore the use of these operators in the next section.

`ip.src == 145.254.160.237`

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	145.254.160.237	65.208.228.223	TCP		

Learn from the THM.

Question 3

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

`http.request.method == GET`

No.	Time	Source	Destination	Protocol	Length	Info
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP/1.1
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1
316	63.698177	10.10.67.199	10.10.15.52	HTTP	393	GET /js/instantpage.min.js HTTP/1.1
320	63.701373	10.10.67.199	10.10.15.52	HTTP	398	GET /images/icon.png HTTP/1.1
335	63.987281	10.10.67.199	10.10.15.52	HTTP	387	GET /post/index.json HTTP/1.1
338	63.997588	10.10.67.199	10.10.15.52	HTTP	366	GET /favicon.ico HTTP/1.1
340	64.005368	10.10.67.199	10.10.15.52	HTTP	481	GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2...
462	64.020692	10.10.67.199	10.10.15.52	HTTP	496	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
471	64.222360	10.10.67.199	10.10.15.52	HTTP	365	GET /posts/reindeer-of-the-week/ HTTP/1.1
475	66.239846	10.10.67.199	10.10.15.52	HTTP	369	GET /posts/post/index.json HTTP/1.1
478	66.249669	10.10.67.199	10.10.15.52	HTTP	463	GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular...
480	66.251644	10.10.67.199	10.10.15.52	HTTP	448	GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1

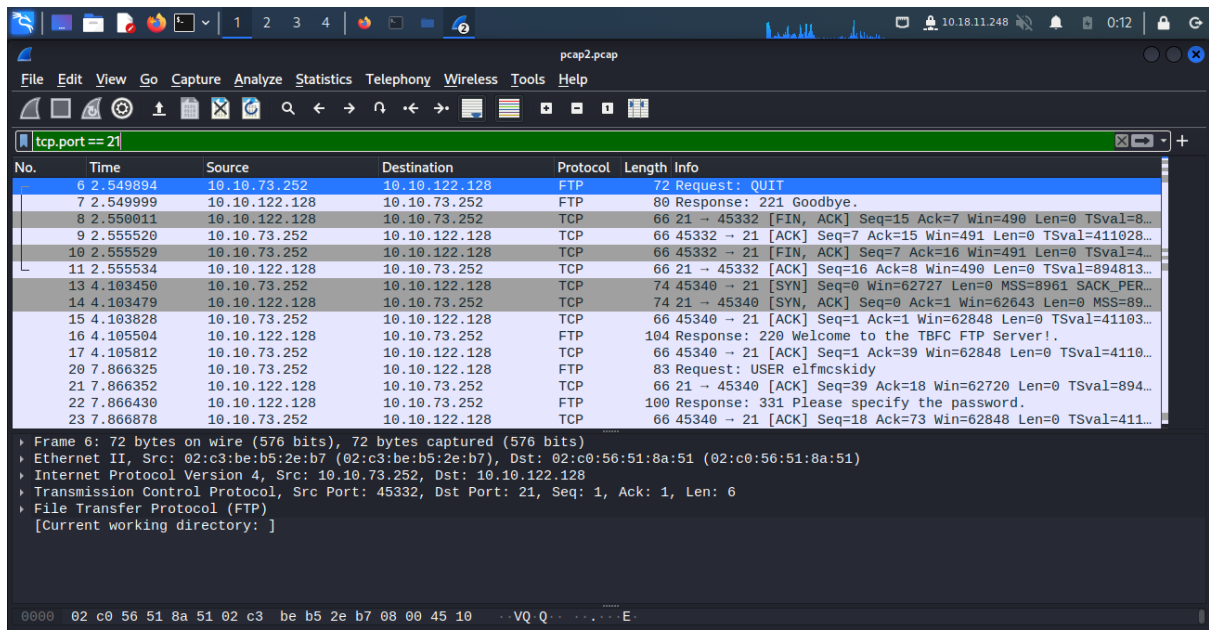
Frame 471: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits)

- Ethernet II, Src: MS-NLB-PhysServer-32_03:60:d9:6c:db (02:23:60:d9:6c:db), Dst: 02:89:03:cb:f7:6b (02:89:03:cb:f7:6b)
- Internet Protocol Version 4, Src: 10.10.67.199, Dst: 10.10.15.52
- Transmission Control Protocol, Src Port: 55658, Dst Port: 80, Seq: 1192, Ack: 1742344, Len: 299
- Hypertext Transfer Protocol

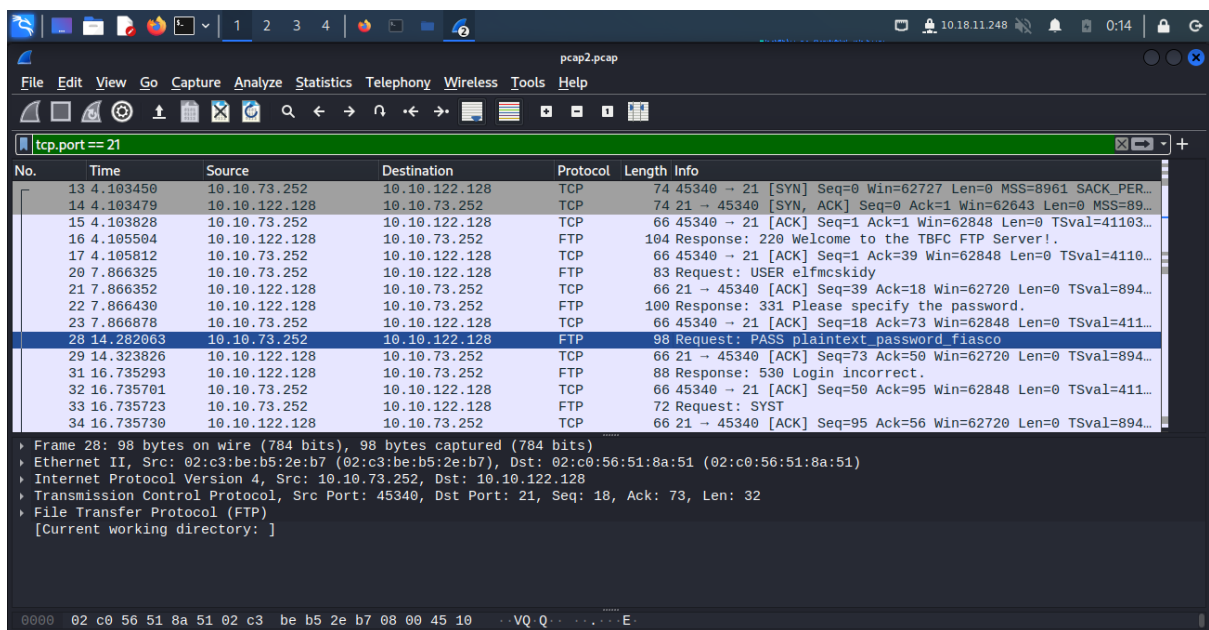
0000 02 89 03 cb f7 6b 02 23 60 d9 6c db 08 00 45 00k#`l...E

Copy the HTTP GET filter and paste on the search bar and find the answer of article with `/posts/`.

Question 4

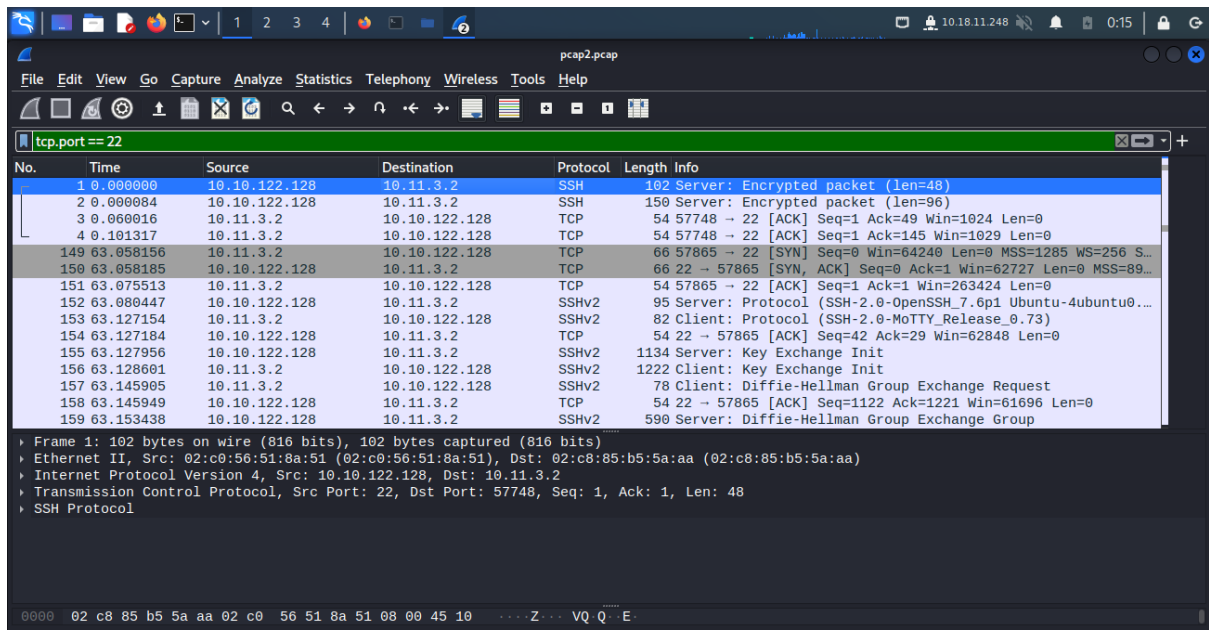


Open pcap2 and use tcp.port == 21 to filter it.



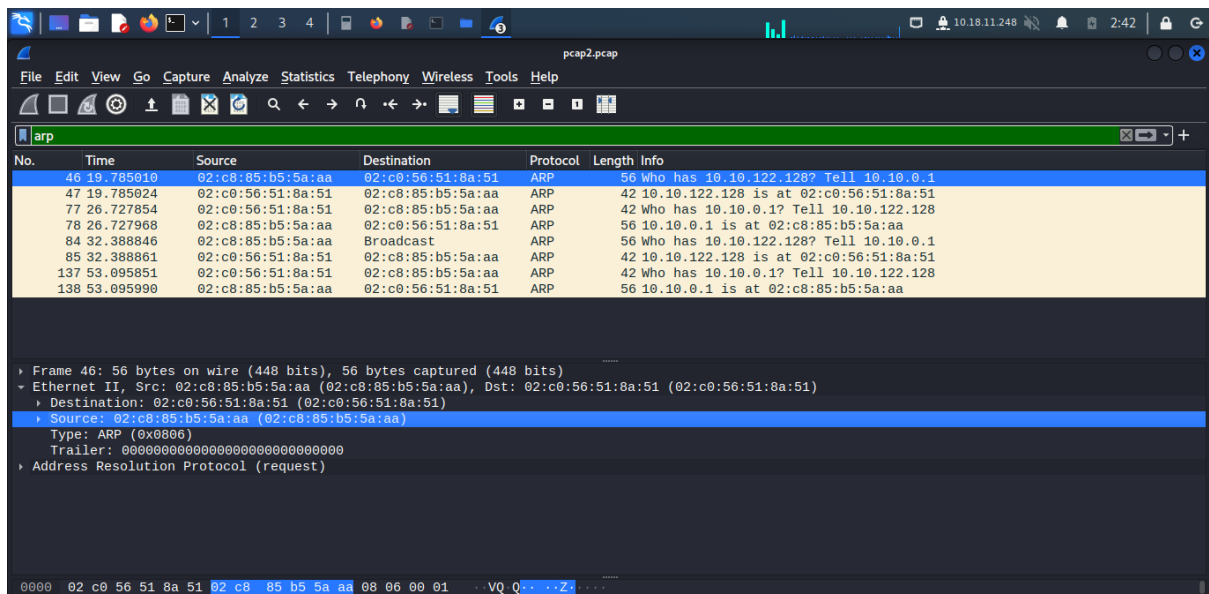
Next, look for the answer. The password shown.

Question 5



Continue using `tcp.port == 22` this filter, change the number of 22 into other number to find the protocol that has been encrypted. Lastly, SSH shown in number 21.

Question 6



Type `arp` on the bar and the answer shown.

Question 7

Wireshark interface showing packet list and packet details for pcap3.pcap. The packet list displays various SSH and TCP packets. The packet details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
112	7.738357	10.10.53.219	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
113	7.792911	10.11.3.2	10.10.53.219	TCP	54	60319 → 22 [ACK] Seq=1745 Ack=1857 Win=1024 Len=0
114	7.847741	10.11.3.2	10.10.53.219	SSH	102	Client: Encrypted packet (len=48)
115	7.848033	10.10.53.219	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
116	7.904570	10.11.3.2	10.10.53.219	TCP	54	60319 → 22 [ACK] Seq=1793 Ack=1905 Win=1024 Len=0
117	8.003704	10.11.3.2	10.10.53.219	SSH	102	Client: Encrypted packet (len=48)
118	8.004026	10.10.53.219	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
119	8.061454	10.11.3.2	10.10.53.219	TCP	54	60319 → 22 [ACK] Seq=1841 Ack=1953 Win=1024 Len=0
120	8.072616	10.11.3.2	10.10.53.219	SSH	102	Client: Encrypted packet (len=48)
121	8.072897	10.10.53.219	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
122	8.130365	10.11.3.2	10.10.53.219	TCP	54	60319 → 22 [ACK] Seq=1889 Ack=2001 Win=1024 Len=0
123	8.326129	10.11.3.2	10.10.53.219	SSH	102	Client: Encrypted packet (len=48)
124	8.326457	10.10.53.219	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
125	8.383856	10.11.3.2	10.10.53.219	TCP	54	60319 → 22 [ACK] Seq=1937 Ack=2049 Win=1029 Len=0
126	8.495841	10.11.3.2	10.10.53.219	SSH	102	Client: Encrypted packet (len=48)

Frame 168: 4852 bytes on wire (38816 bits), 4852 bytes captured (38816 bits)
Ethernet II, Src: MS-NLB-PhysServer-07:7b:6f:c0:01 (02:07:7b:6f:c0:01), Dst: 02:cd:4e:c8:87:f1 (02:cd:4e:c8:87:f1)
Internet Protocol Version 4, Src: 10.10.21.210, Dst: 10.10.53.219
Transmission Control Protocol, Src Port: 80, Dst Port: 38454, Seq: 1, Ack: 74, Len: 4786
Hypertext Transfer Protocol
Line-based text data: text/html (212 lines)

0000 02 cd 4e c8 87 f1 02 07 7b 6f c0 01 08 00 45 00 ... N ... { 0 ... E

Wireshark interface showing the 'Export - HTTP object list' dialog box. The dialog displays a list of HTTP objects with columns for Packet, Hostname, Content Type, Size, and Filename. The list shows two objects: packet 168 (text/html, 4,532 bytes) and packet 395 (application/zip, 565 kB).

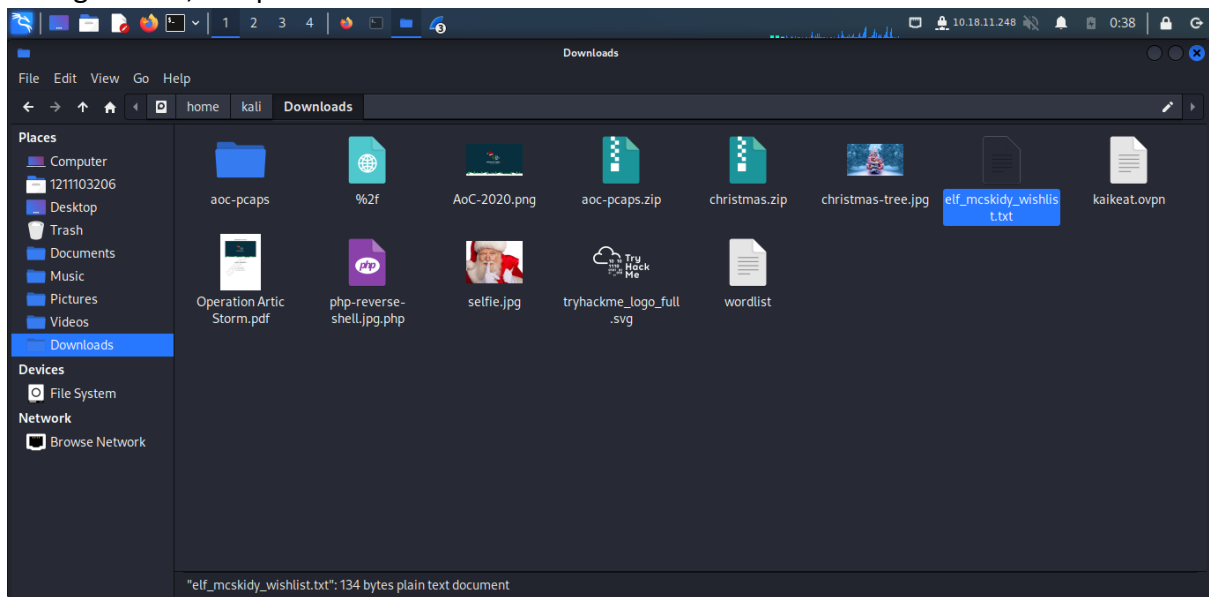
Packet	Hostname	Content Type	Size	Filename
168	tbfc.blog	text/html	4,532 bytes	
395	tbfc.blog	application/zip	565 kB	christmas.zip

Text Filter: Content Type: All Content-Types

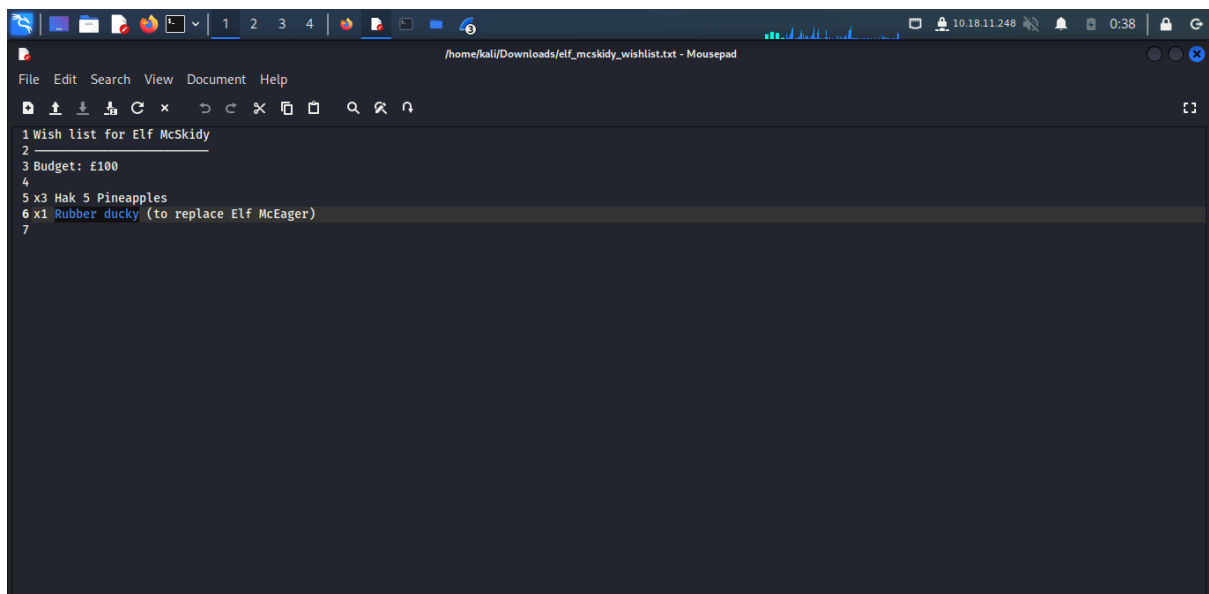
Save Save All Preview Close Help

0000 02 cd 4e c8 87 f1 02 07 7b 6f c0 01 08 00 45 00 ... N ... { 0 ... E

Open pcap3 and click export object of this file, there was 2 file shown and save it. After saving the file, unzip the file.

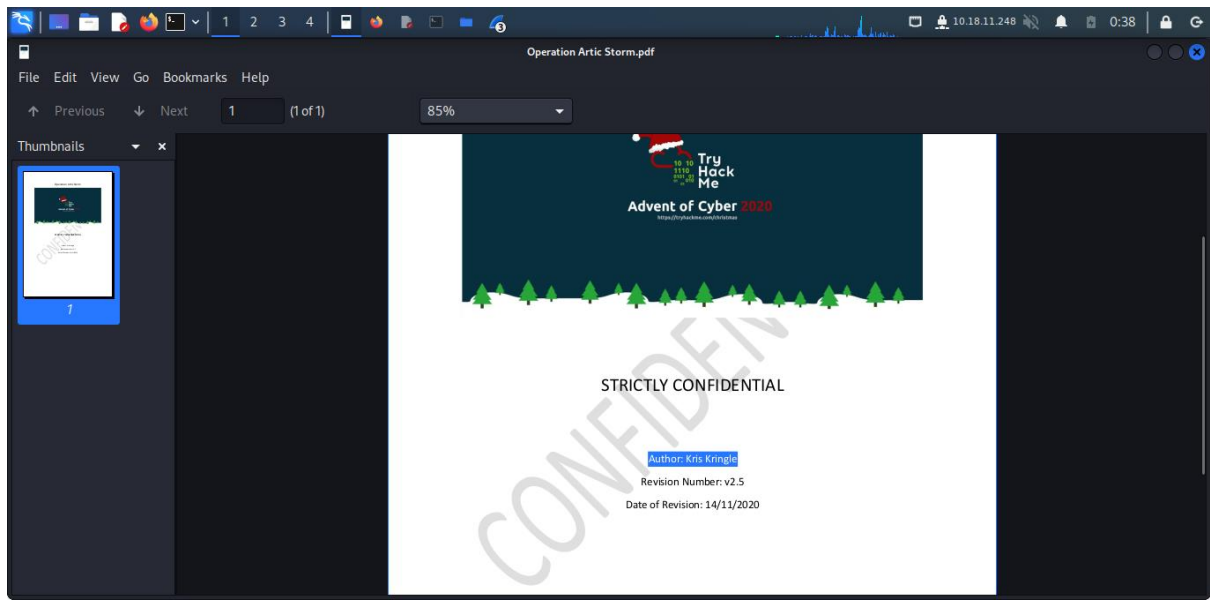


Next, open the txt file.



The answer shown.

Question 8



Open the pdf file that has been unzip, the name of the author shown.