

PSP0201

Week 3

Writeup

Group Name: Haxon

Members

ID	Name	Role
1211102370	Lau Zi Thao	Leader
1211102797	Teng Wei Joe	Member
1211103142	Wong Khai King	Member
1211101029	Garrison Goh Zen Ken	Member

Day 6: Web Exploitation – Be careful with what you wish on a Christmas night

Tools used: Kali Linux, Firefox, OWASP Zap

Solution/walkthrough:

Question 1

Examine the OWASP Cheat Sheet. Match the input validation level with the correct description.

Syntactic validation should enforce correct syntax of structured fields (e.g. SSN, date, currency symbol).

Semantic validation should enforce correctness of their *values* in the specific business context (e.g. start date is before end date, price is within expected range).

	Syntactic	Semantic
enforce correctness of their values in the specific business context	<input type="radio"/>	<input checked="" type="radio"/>
enforce correct syntax of structured fields	<input checked="" type="radio"/>	<input type="radio"/>

Question 2

Examine the OWASP Cheat Sheet. What is the regular expression used to validate a US Zip code?

Validating a U.S. Zip Code (5 digits plus optional -4)

```
^\d{5}(-\d{4})?$
```

Question 3

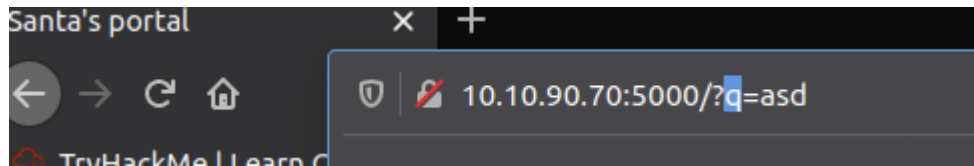
What vulnerability type was used to exploit the application?

Stored XSS works when a certain malicious JavaScript is submitted and later on stored directly on the website. For example, comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other words, **in any content that persistently exists on the website and can be viewed by victims.**

A: stored crosssite scripting

Question 4

What query string can be abused to craft a reflected XSS?

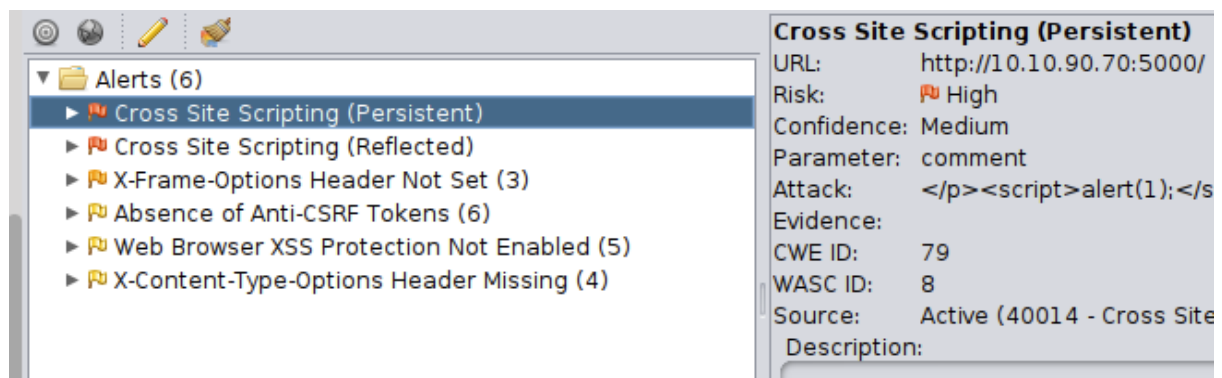


By typing in any word in the “Search query” bar and clicking enter, the query string can be found in the URL.

A: q

Question 5

Run a ZAP (zapoxy) automated scan on the target. How many XSS alerts of high priority are in the scan?

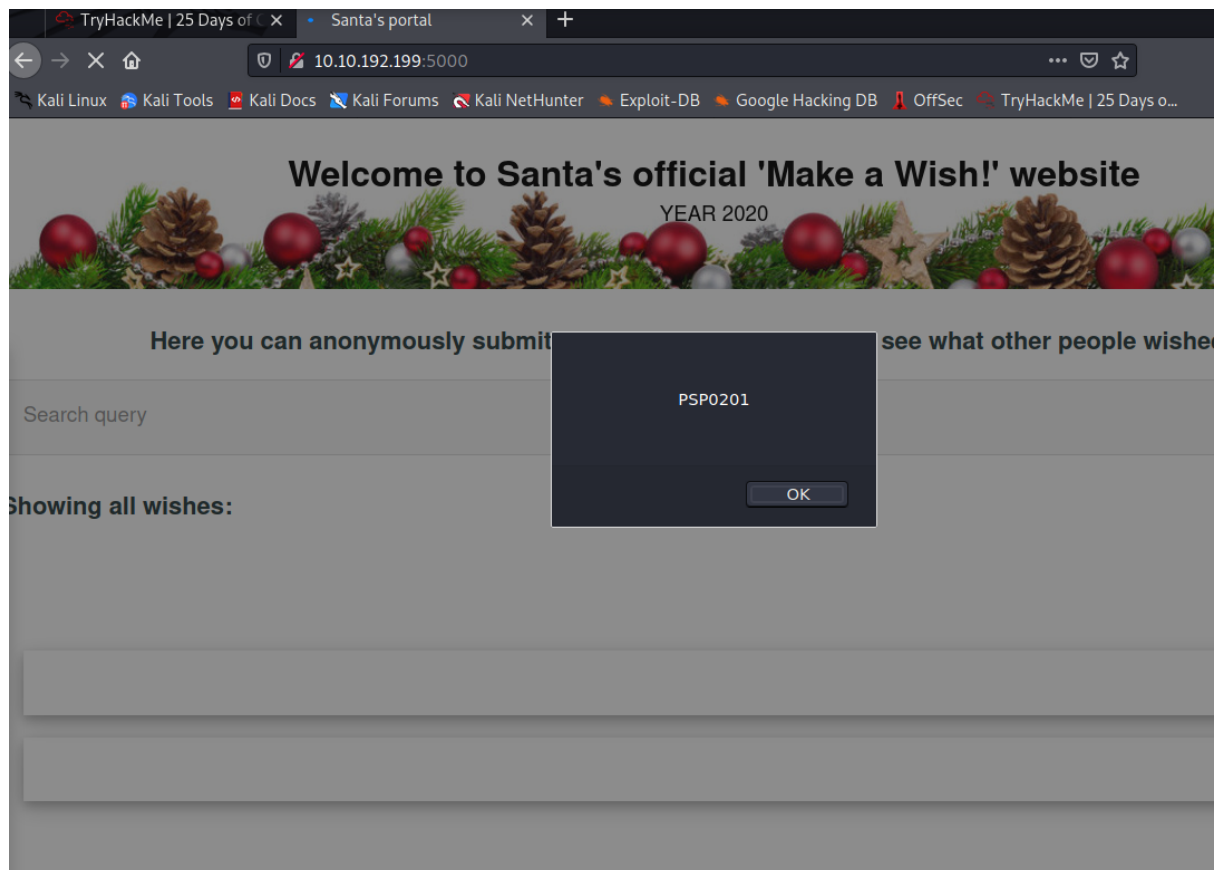
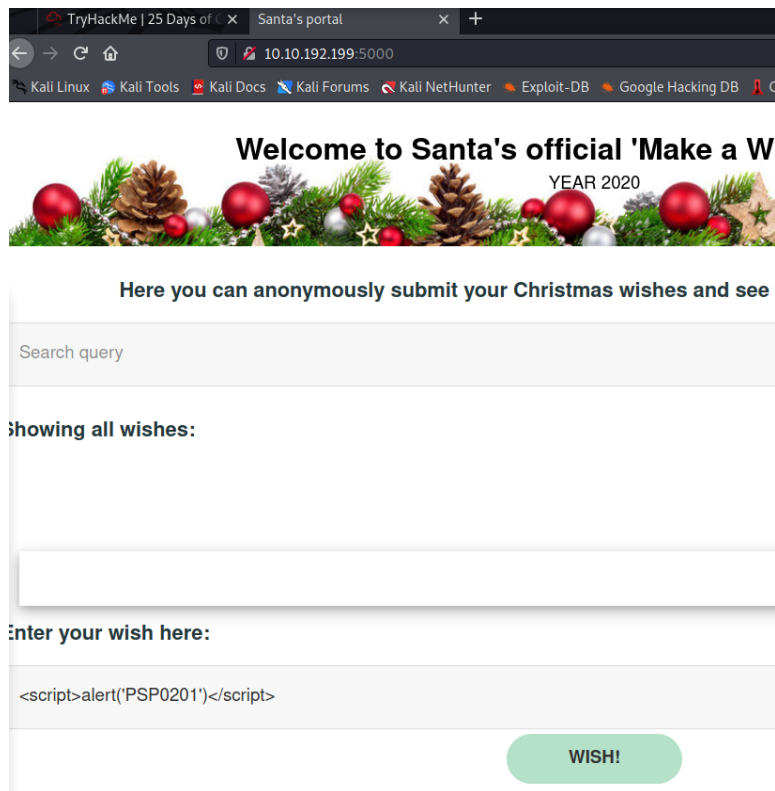


A: 2

2 high priority XSS alerts, being “Cross Site Scripting (Persistent)”, and “Cross Site Scripting (Reflected)”

Question 6

What Javascript code should you put in the wish text box if you want to show an alert saying "PSP0201"?

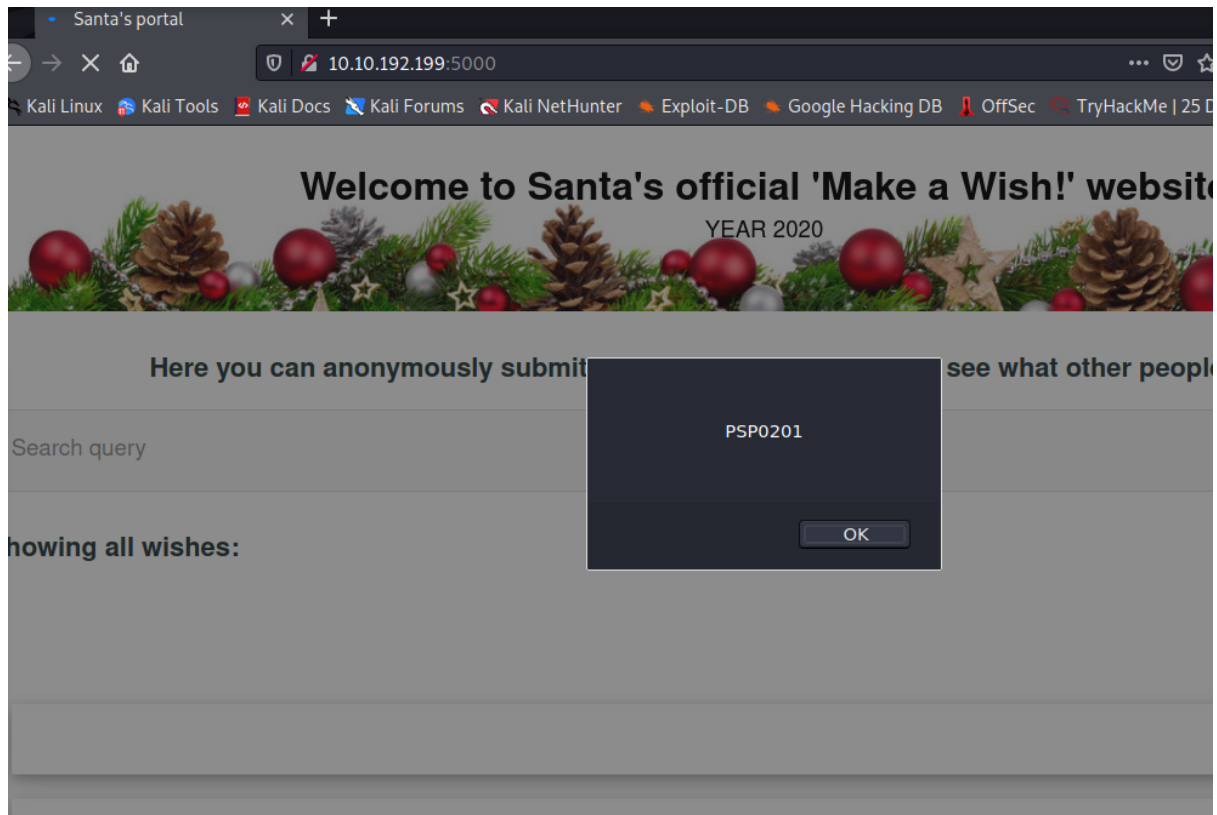


By entering “`<script>alert('PSP0201')</script>`” and clicking wish, the alert box immediately pops up saying “PSP0201”

A: `<script>alert('PSP0201')</script>`

Question 7

Close your browser and revisit the site MACHINE-IP:5000 again. Does your XSS attack persist?



After closing the browser and revisiting the address, the alert box pops up with the message that I uploaded earlier.

Thought Process/Methodology:

After starting the machine, type in "MACHINE_IP:5000" into the browser search bar to access Santa's portal. Typing any string in "Search query" shows us all wishes that have "search query" in the name, as well as the query string showing in the URL. Running a ZAP (zaproxy) automated scan on the target returns 6 alerts with 2 high priority, 1 moderate, and 3 low priority alerts. There are 2 high priority XSS alerts, being "Cross Site Scripting (Persistent)", and "Cross Site Scripting (Reflected)". By entering "<script>alert('PSP0201')</script>" and clicking wish, an alert box immediately pops up saying "PSP0201", which shows that the XSS exploit worked. After closing the browser and revisiting the address, the alert box pops up with the message that I uploaded earlier, showing that it is a stored XSS.

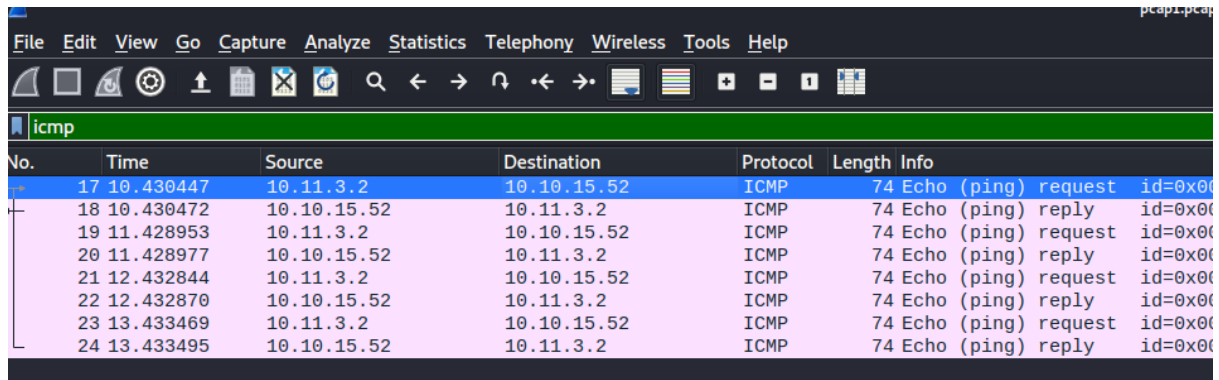
Day 7: Networking – The Grinch Really Did Steal Christmas

Tools used: Kali Linux, Firefox, Wireshark

Solution/walkthrough:

Question 1

Open "pcap1.pcap" in Wireshark. What is the IP address that initiates an ICMP/ping?



No.	Time	Source	Destination	Protocol	Length	Info
17	10.430447	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request id=0x00
18	10.430472	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply id=0x00
19	11.428953	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request id=0x00
20	11.428977	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply id=0x00
21	12.432844	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request id=0x00
22	12.432870	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply id=0x00
23	13.433469	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request id=0x00
24	13.433495	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply id=0x00

A: 10.11.3.2

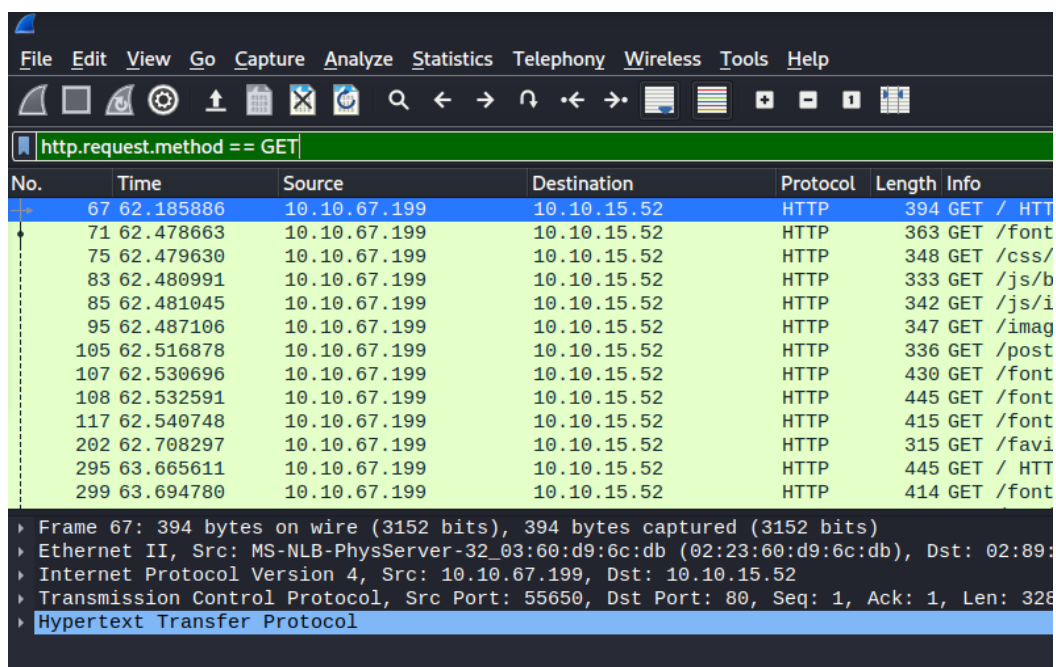
Question 2

If we only wanted to see HTTP GET requests in our "pcap1.pcap" file, what filter would we use?

tcp/udp.port Show all packets that are sent via the protocol and port specified

protocol.request.method Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a GET and POST to retrieve and submit data accordingly.

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 I wish to search for (`145.254.160.237`). We'll quickly explore the use of these operators in the next section.



No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /font
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/b
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/i
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /imag
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	GET /font
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /font
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /font
202	62.708297	10.10.67.199	10.10.15.52	HTTP	315	GET /favi
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /font

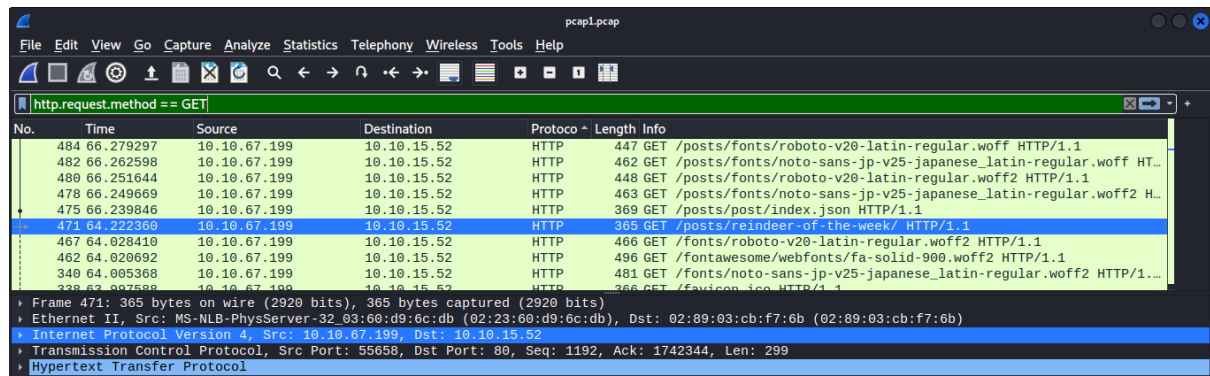
Frame 67: 394 bytes on wire (3152 bits), 394 bytes captured (3152 bits)
Ethernet II, Src: MS-NLB-PhysServer-32_03:60:d9:6c:db (02:23:60:d9:6c:db), Dst: 02:89:
Internet Protocol Version 4, Src: 10.10.67.199, Dst: 10.10.15.52
Transmission Control Protocol, Src Port: 55650, Dst Port: 80, Seq: 1, Ack: 1, Len: 328
Hypertext Transfer Protocol

A: `http.request.method == GET`

The answer is found in THM

Question 3

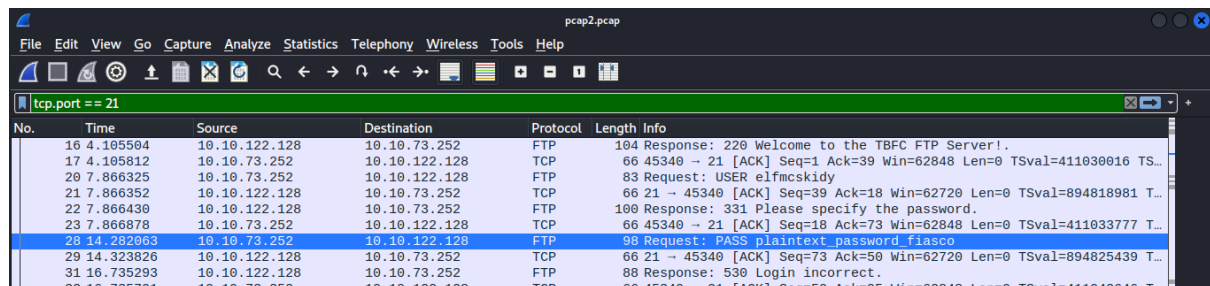
Now apply this filter to "pcap1.pcap" in Wireshark, what is the name of the article that the IP address "10.10.67.199" visited?



By applying the filter earlier (`http.request.method == GET`), there is a result that shows the name of the article in the posts directory which is visited by 10.10.67.199.

Question 4

Let's begin analysing "pcap2.pcap". Look at the captured FTP traffic; what password was leaked during the login process?



By filtering with "`tcp.port == 21`" or "`FTP`", we can see the keyword PASS on one of the info.

Question 5

Continuing with our analysis of "pcap2.pcap", what is the name of the protocol that is encrypted?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.122.128	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
2	0.000004	10.10.122.128	10.11.3.2	SSH	150	Server: Encrypted packet (len=96)
3	0.000016	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=49 Win=1024 Len=0
4	0.101317	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=145 Win=1029 Len=0
5	1.127866	10.10.122.128	91.189.92.40	TCP	74	33400 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSv...
6	2.549894	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
7	2.549999	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
8	2.550011	10.10.122.128	10.10.73.252	TCP	66	21 → 45332 [FIN, ACK] Seq=15 Ack=7 Win=490 Len=0 TSval=894813665...
9	2.555520	10.10.73.252	10.10.122.128	TCP	66	45332 → 21 [ACK] Seq=7 Ack=15 Win=491 Len=0 TSval=411028463 TSec...
10	2.555528	10.10.73.252	10.10.122.128	TCP	66	45332 → 21 [FIN, ACK] Seq=7 Ack=15 Win=491 Len=0 TSval=411028463...

The encrypted packets show up with the SSH protocol.

Question 6

Examine the ARP communications. Who has 10.10.122.128? Tell 10.10.10.1. Answer: 10.10.122.128 is at

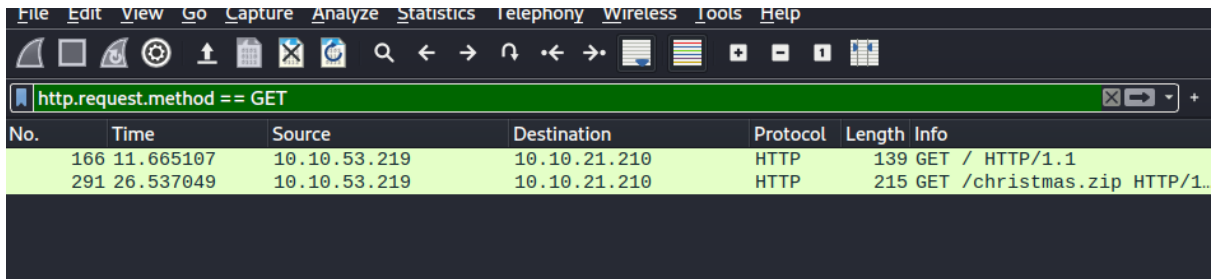
No.	Time	Source	Destination	Protocol	Length	Info
46	19.785010	02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56	who has 10.10.122.128? Tell 10.10.0.1
47	19.785024	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	10.10.122.128 is at 02:c0:56:51:8a:51
77	26.727854	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	who has 10.10.0.1? Tell 10.10.122.128
78	26.727968	02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56	10.10.0.1 is at 02:c8:85:b5:5a:aa
84	32.388846	02:c8:85:b5:5a:aa	Broadcast	ARP	56	who has 10.10.122.128? Tell 10.10.0.1
85	32.388861	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	10.10.122.128 is at 02:c0:56:51:8a:51
137	53.095851	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	who has 10.10.0.1? Tell 10.10.122.128
138	53.095990	02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56	10.10.0.1 is at 02:c8:85:b5:5a:aa

Filtering with the keyword “ARP” shows us the packets with ARP protocols, and we are able to find the line with “10.10.122.128 is at” and copy the answer that follows.

Question 7

Analyse "pcap3.pcap" and recover Christmas! What is on Elf McSkidy's wishlist that will be used to replace Elf McEager?

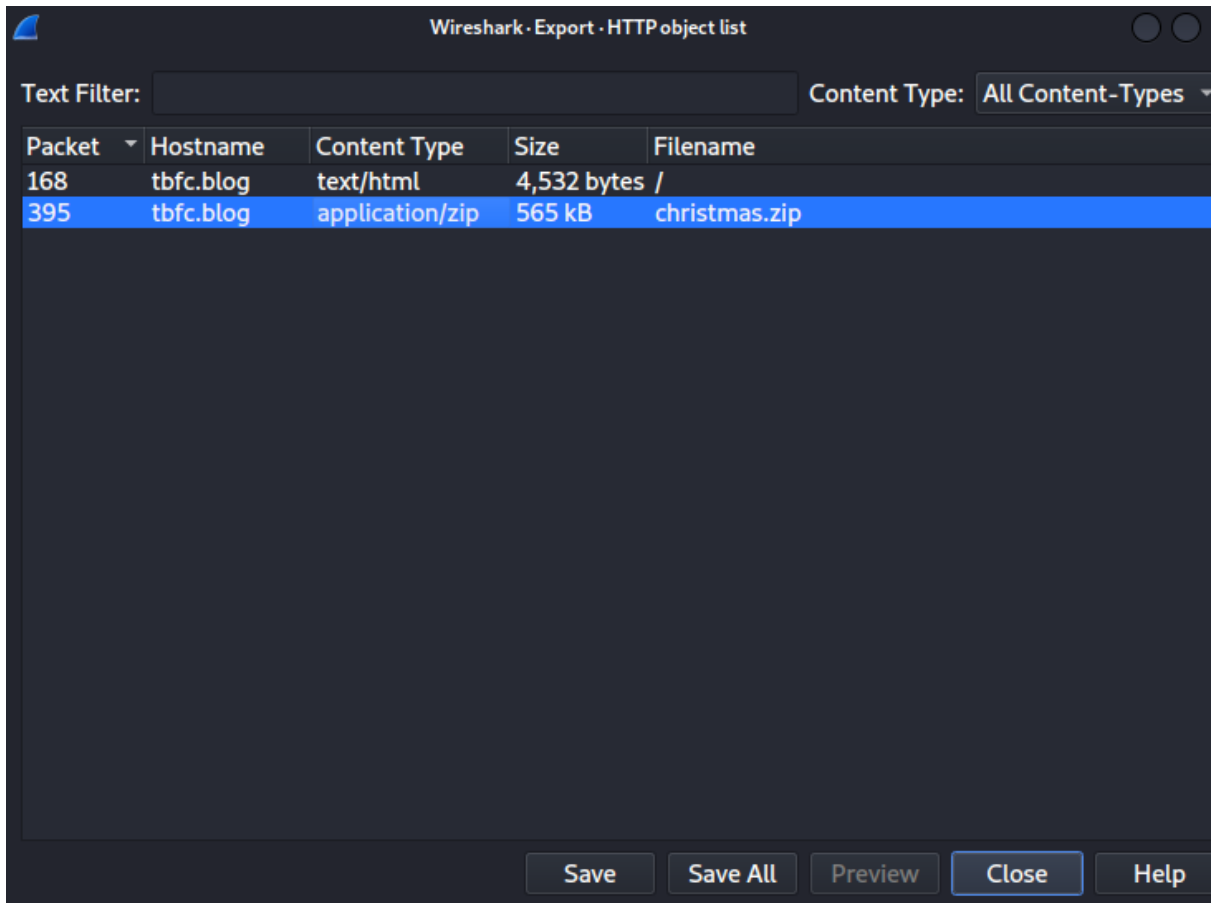
No.	Time	Source	Destination	Protocol	Length	Info
166	11.665107	10.10.53.219	10.10.21.210	HTTP	139	GET / HTTP/1.1
168	11.665723	10.10.21.210	10.10.53.219	HTTP	4852	HTTP/1.1 200 OK (text/html)
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	GET /christmas.zip HTTP/1.1
395	26.542475	10.10.21.210	10.10.53.219	HTTP	10388	HTTP/1.1 200 OK (application/zip)



The screenshot shows the Wireshark main window with the filter bar set to `http.request.method == GET`. The packet list table below shows three packets, with the third packet (No. 291) highlighted in green, indicating it contains a GET request for a file.

No.	Time	Source	Destination	Protocol	Length	Info
166	11.665107	10.10.53.219	10.10.21.210	HTTP	139	GET / HTTP/1.1
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	GET /christmas.zip HTTP/1.1

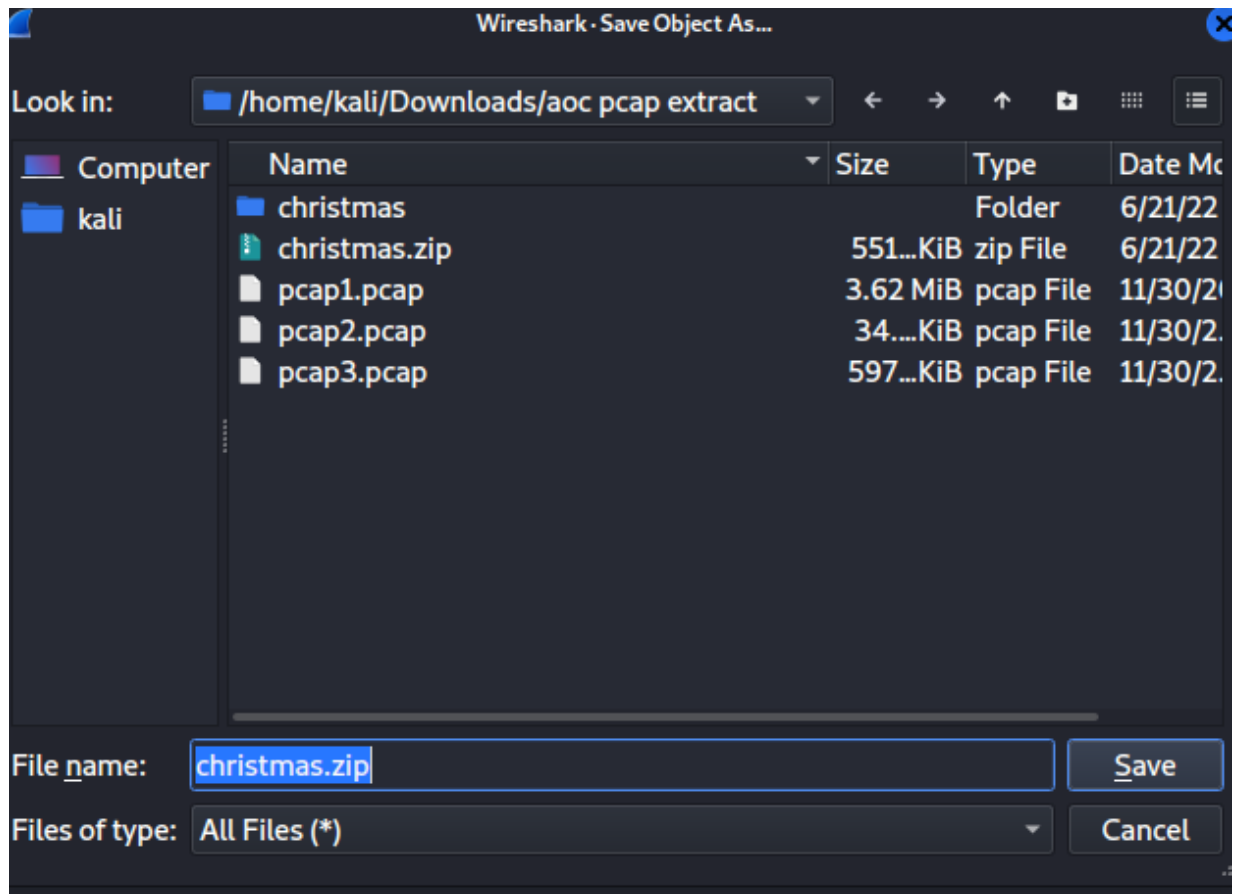
Filtering with “http” or “http.request.method == GET” shows a few packets, with one of them containing a zip file.



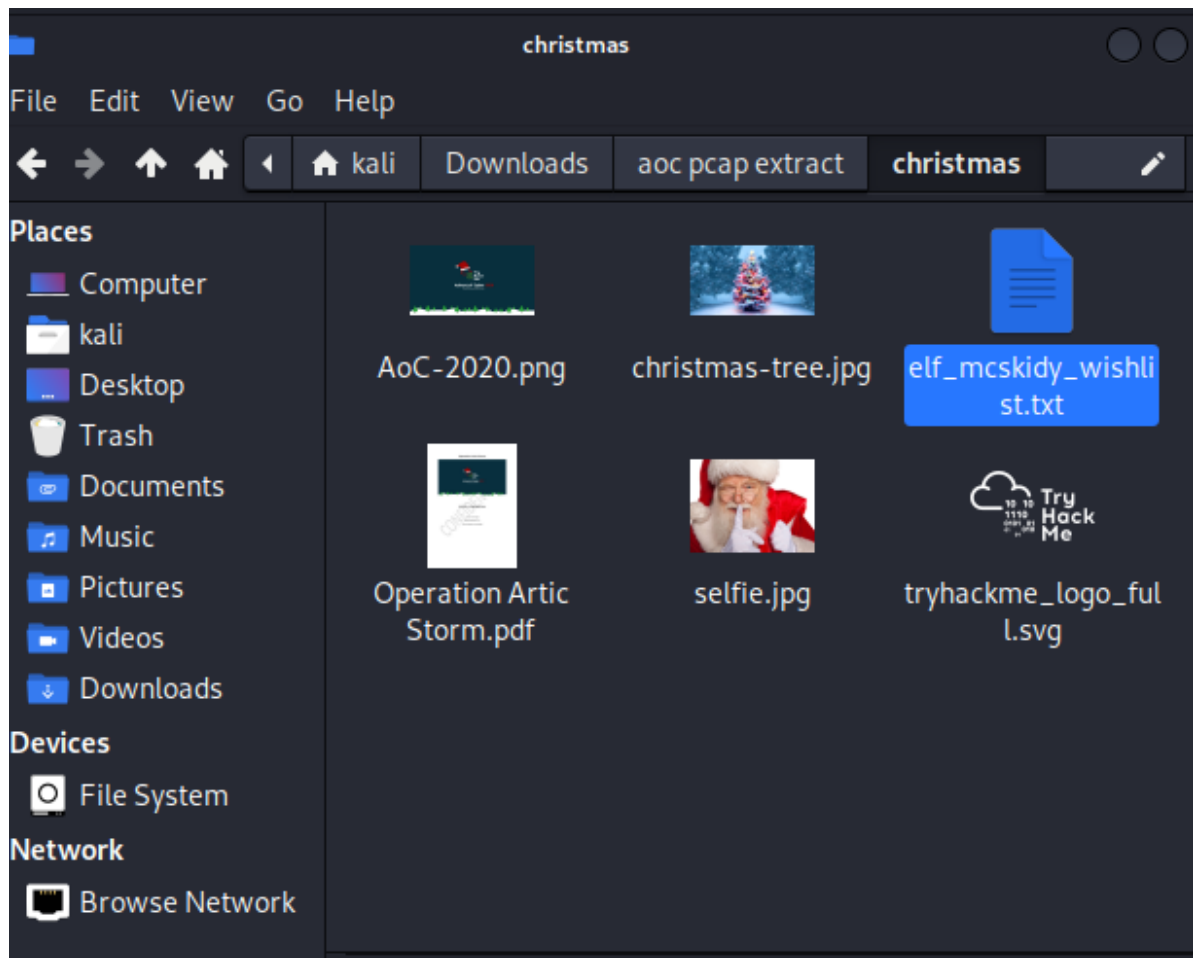
The screenshot shows the 'Wireshark - Export - HTTP object list' dialog box. It has a 'Text Filter' field and a 'Content Type' dropdown set to 'All Content-Types'. The table below lists two HTTP objects, with the second one (Packet 395) highlighted in blue, indicating it is a zip file.

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

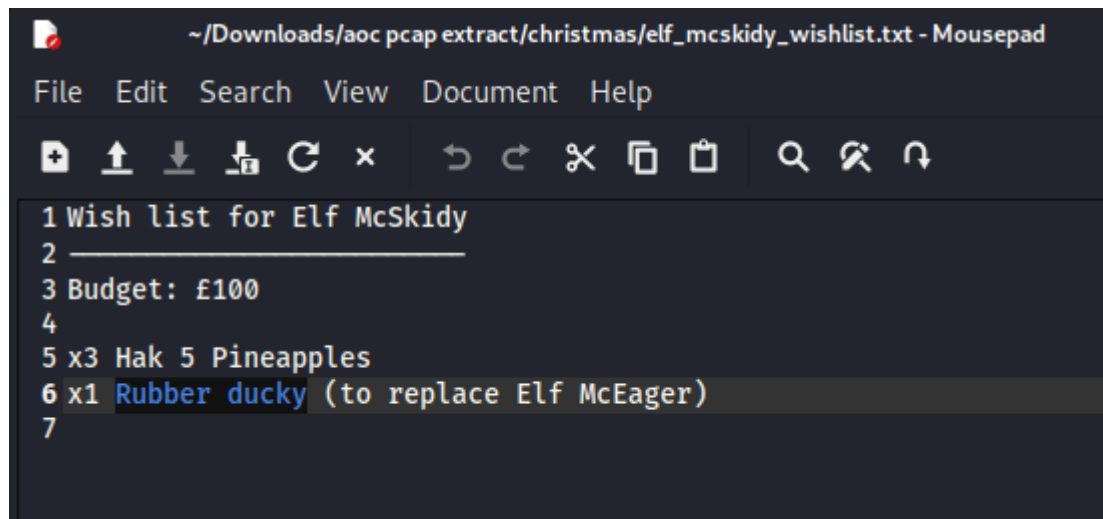
At the bottom of the dialog, there are buttons for 'Save', 'Save All', 'Preview', 'Close', and 'Help'.



We then export the zip file as http, and save it.



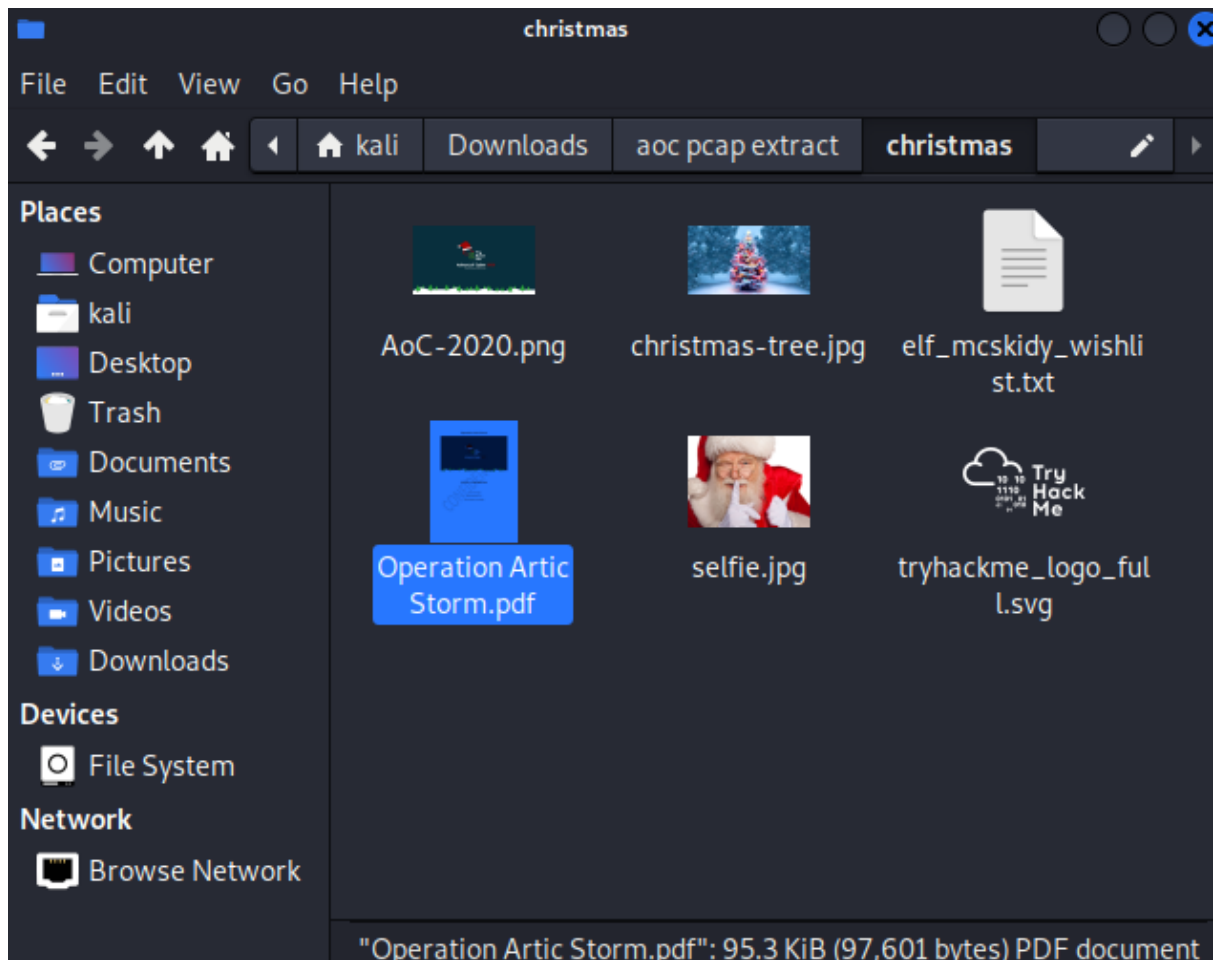
After extracting the zip file, we see the wishlist.txt.



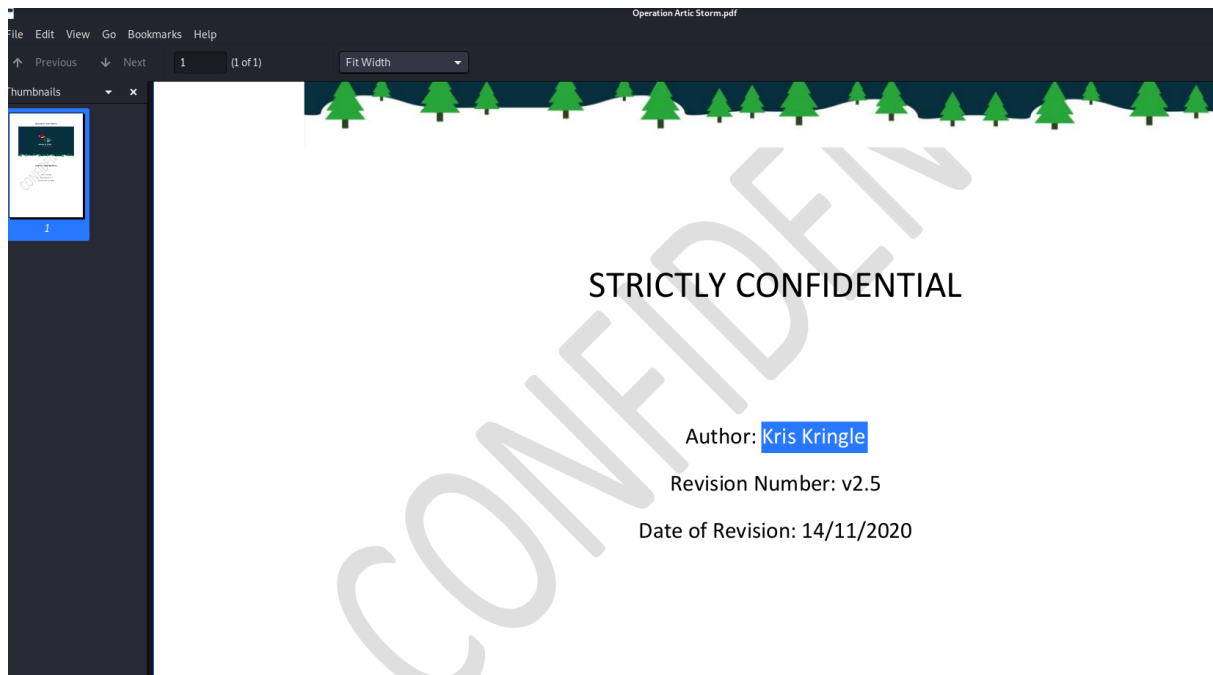
Inside is the answer to what is used to replace elf mceager.

Question 8

Who is the author of Operation Arctic Storm?



The operation arctic storm pdf is found in the christmas extracted zip file.



The author of operation arctic storm is found in the pdf.

Thought Process/Methodology:

We opened the pcap1.pcap file through Wireshark. We looked through the ICMP protocol and found that the source IP that initiates a ping is 10.11.3.2. To find all the GET requests in the file, we used the filter `http.request.method == GET` to see all of it in which one of them shows the name of the article (reindeer-of-the-week). After that, we opened the pcap2.pcap file. By filtering out only FTP protocols, we were able to find the password that was leaked in one of the results (plaintext_password_fiasco).

While looking through, we noticed that the SSH protocols of the pcap2.pcap file contained encrypted packets. After that, we began looking through the ARP communications by filtering it out. The lines were able to tell us who had 10.10.122.128. After analysing pcap2.pcap, we move on to pcap3.pcap. While looking through the file, we used the `http.request.method == GET` filter and managed to get a result containing a zipped folder (christmas.zip). We then proceeded to export and download the file to view its contents. The folder consisted of many different files. In elf_mcskidys_wishlist.txt, we can see what is on Elf McSkidy's wishlist that will be used to replace Elf McEager. Not only that, there is also another file named Operation Artic Storm.pdf file which contains the name of the author, Kris Kringle.

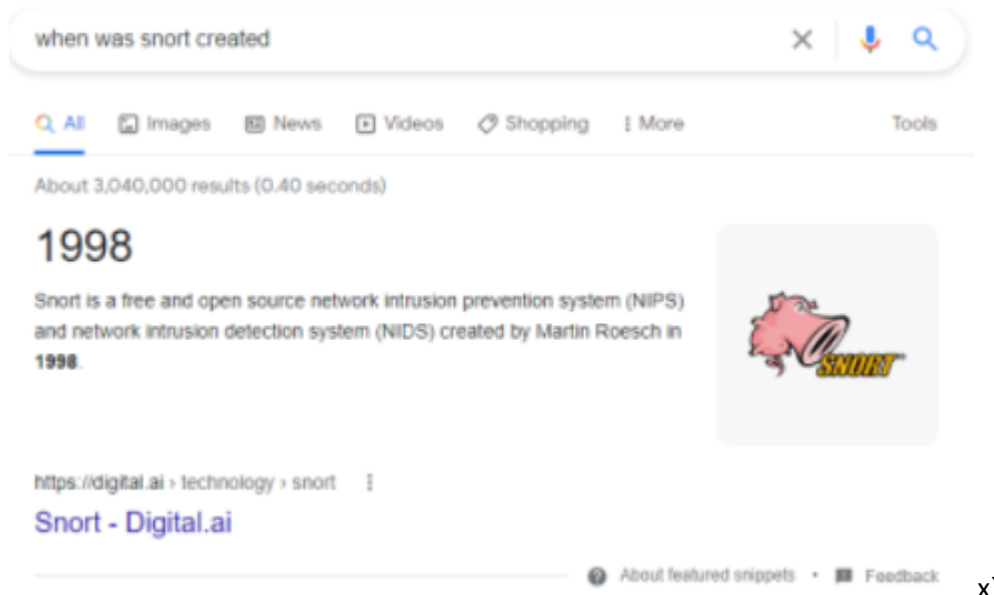
Day 8: Networking – What's Under the Christmas Tree?

Tools used: Kali Linux, nmap

Solution/walkthrough:

Question 1

When was Snort created?



A quick google search reveals the answer.

Question 2

Using Nmap on MACHINE_IP , what are the port numbers of the three services running?

```
root@ip-10-10-97-180:~# nmap -sS 10.10.28.33

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 04:53 BST
Nmap scan report for ip-10-10-28-33.eu-west-1.compute.internal (10.10.28.33)
Host is up (0.00089s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
MAC Address: 02:DB:5B:BC:7E:C1 (Unknown)
```

The ports 80,2222,3389 are shown along with their respective state and service name

Question 3

Use Nmap to determine the name of the Linux distribution that is running, what is reported as the most likely distribution to be running?

```

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 05:01 BST
Nmap scan report for ip-10-10-28-33.eu-west-1.compute.internal (10.10.28.33)
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:DB:5B:BC:7E:C1 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .

```

Ubuntu shows up multiple times after running “nmap -A MACHINE_IP” and “nmap -sV MACHINE_IP”.

Question 4

What is the version of Apache?

```

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 05:01 BST
Nmap scan report for ip-10-10-28-33.eu-west-1.compute.internal (10.10.28.33)
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:DB:5B:BC:7E:C1 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .

```

By running “nmap -A MACHINE_IP” or “nmap -sV MACHINE_IP”, the version of apache can be clearly seen.

Question 5

What is running on port 2222?

```

root@ip-10-10-97-180:~# nmap -A 10.10.28.33

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 05:40 BST
Nmap scan report for ip-10-10-28-33.eu-west-1.compute.internal (10.10.28.33)
Host is up (0.00043s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC&#39;s Internal Blog
2222/tcp  open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|   2048 cf:c9:99:d0:5c:09:27:cd:a1:a8:1b:c2:b1:d5:ef:a6 (RSA)
|   256  4c:d4:f9:20:6b:ce:fc:62:99:54:7d:c2:b4:b2:f2:b2 (ECDSA)
|   256  d0:e6:72:18:b5:20:89:75:d5:69:74:ac:cc:b8:3b:9b (EdDSA)
3389/tcp  open  ms-wbt-server  xrdp
MAC Address: 02:DB:5B:BC:7E:C1 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=6/22%OT=80%CT=1%CU=36668%PV=Y%DS=1%DC=D%C=Y%M=02DB5B%T

```

By running “nmap -A MACHINE_IP”, the answer can be found beside port 2222 info.

Question 6

Use Nmap's Network Scripting Engine (NSE) to retrieve the "HTTP-TITLE" of the webserver. Based on the value returned, what do we think this website might be used for?

```

root@ip-10-10-97-180:~# nmap --script http-title 10.10.28.33

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 05:37 BST
Nmap scan report for ip-10-10-28-33.eu-west-1.compute.internal (10.10.28.33)
Host is up (0.00067s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
|_http-title: TBFC&#39;s Internal Blog
2222/tcp  open  EthernetIP-1
3389/tcp  open  ms-wbt-server
MAC Address: 02:DB:5B:BC:7E:C1 (Unknown)

```

Judging from the “http-title” line, a keyword “blog” can be seen, thus being the most likely answer.

Thought Process/Methodology:

We first ran a nmap -sS scan, which is a SYN scan, it shows the ports and its respective services, as well as a MAC address. It showed us three different services running under three different ports, which were 80, 2222 and 3389. We found that “nmap -sV MACHINE_IP” is the best way to find out the linux distribution that is running, and their versions. The -A, -sS, and -O scan types were also useful in this task. After running a few scans, we found out that the linux distribution that was running was Ubuntu. The version of Apache was also shown by the scan result (2.4.29). By looking at the port 2222 and the results beside it, we can see that port 2222 is running on SSH. Lastly, by

looking at the value of the “http-title” returned by the scan, it is stated that it is an internal blog, thus showing us what the website might be used for (blogging).

Day 9: Networking – Anyone can be Santa!

Tools used: Kali Linux, ftp, nano

Solution/walkthrough:

Question 1

What are the directories you found on the FTP site?

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 0          0              4096 Nov 16  2020 backups
drwxr-xr-x  2 0          0              4096 Nov 16  2020 elf_workshops
drwxr-xr-x  2 0          0              4096 Nov 16  2020 human_resources
drwxrwxrwx  2 65534     65534           4096 Nov 16  2020 public
226 Directory send OK.
ftp> █
```

After logging into the tbfc ftp server with “anonymous” as name, we find these 4 directories.

Question 2

Name the directory on the FTP server that has data accessible by the "anonymous" user

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 0          0              4096 Nov 16  2020 backups
drwxr-xr-x  2 0          0              4096 Nov 16  2020 elf_workshops
drwxr-xr-x  2 0          0              4096 Nov 16  2020 human_resources
drwxrwxrwx  2 65534     65534           4096 Nov 16  2020 public
226 Directory send OK.
ftp> █
```

“public” is the only directory that contains files in the server when viewed as “anonymous”

Question 3

What script gets executed within this directory?

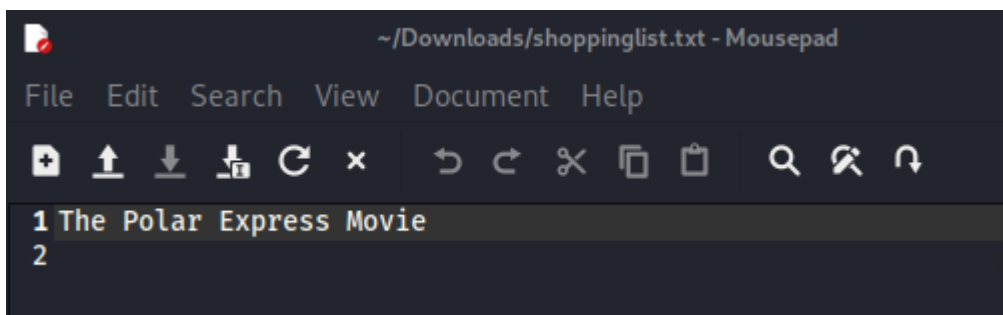
```
ftp> cd public
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxr-xr-x  1 111      113      341 Nov 16  2020 backup.sh
-rw-rw-rw-  1 111      113      24 Nov 16  2020 shoppinglist.txt
226 Directory send OK.
ftp> █
```

There is a shell file in the directory which contains a script which gets executed.

Question 4

What movie did Santa have on his Christmas shopping list?

```
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
226 Transfer complete.
24 bytes received in 0.00 secs (600.9615 kB/s)
ftp> █
```



The “public” directory contains a shoppinglist.txt, which upon downloading and viewing, reveals the name of the movie.

Question 5

Re-upload this script to contain malicious data (just like we did in section 9.6. Output the contents of /root/flag.txt!

```
root@ip-10-10-216-95: ~
File Edit View Search Terminal Tabs Help
root@ip-10-10-216-95: ~ x root@ip-10-10-216-95: ~ x
root@ip-10-10-216-95:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.140.14 55676 received!
bash: cannot set terminal process group (1254): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# ls
ls
flag.txt
root@tbfc-ftp-01:~# cat flag.txt
cat flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#
```

Thought Process/Methodology:

Firstly, we logged in to the FTP server by using the ftp command. We used the identity “anonymous” and managed to get through. We then listed all the contents of the FTP server using the ls command and found 4 directories. It turns out the public directory is the only one accessible with our current identity. We navigated into the public directory and listed the content in it again. There were 2 files, one named backup.sh and shoppinglist.txt. We used the get command to obtain those 2 files. The backup.sh file was a shell script and the shoppinglist.txt contained the name of a movie. We then opened up a text editor (nano) to edit the contents of the shell script. We commented out all of its contents and put in a reverse shell command line. After that, we logged back into the FTP server and navigated to the public directory and uploaded the reverse shell there using the put command. After that, we opened up a netcat listener for the reverse shell. After waiting for some time, we are able to receive a connection and we are able to get into the root directory of the FTP server. We used the ls command to list the contents of the directory and found a flag.txt. We then used the cat command to read the contents of flag.txt and we managed to get the flag.

Day 10: Networking – Don't be sElfish!

Tools used: Kali Linux, enum4linux

Solution/walkthrough:

Question 1

Examine the help options for enum4linux. Match the following flags with the descriptions.

```
(kali㉿kali)-[/usr/share/enum4linux]
└─$ ./enum4linux.pl -h
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
-U get userlist
-M get machine list*
-S get sharelist
-P get password policy information
-G get group and member list
-d be detailed, applies to -U and -S
-u user specify username to use (default "")
-p pass specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f open file, you press

Additional options:
-a Do all simple enumeration (-U -S -G -P -r -o -n -i).
  This option is enabled if you don't provide any other options.
-h Display this help message and exit
-r enumerate users via RID cycling
-R range RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
-K n Keep searching RIDs until n consecutive RIDs don't correspond to
  a username. Impies RID range ends at 999999. Useful
  against DCs.
-l Get some (limited) info via LDAP 389/TCP (for DCs only)
-s file brute force guessing for share names
-k user User(s) that exists on remote system (default: administrator,guest,krbtgt,
  domain admins,root,bin,none)
  Used to get sid with "lookupsid known_username"
  Use commas to try several users: "-k admin,user1,user2"
-o Get OS information
-i Get printer information
-w wrkg Specify workgroup manually (usually found automatically)
-n Do an nmblookup (similar to nbtstat)
-v Verbose. Shows full commands being run (net, rpcclient, etc.)
-A Aggressive. Do write checks on shares etc
```

Using “./enum4linux.pl -h”, all the flags with their description is shown.

Question 2

Using enum4linux, how many users are there on the Samba server?

```
( Users on 10.10.188.115 )

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:      Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager    Name: elfmcea
ger      Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson Name:      Desc:
Places
user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Sat Jun 25 12:06:31 2022
```

We can see the list of users using the command “./enum4linux.pl -U <machine_ip>”

Question 3

Now how many "shares" are there on the Samba server?

```
Sharename      Type      Comment
-----
tbfc-hr        Disk      tbfc-hr
tbfc-it        Disk      tbfc-it
tbfc-santa     Disk      tbfc-santa
IPC$           IPC       IPC Service (tbfc-smb server (Samba, Ubuntu
))
Reconnecting with SMB1 for workgroup listing.
Server         Comment
```

The list of shares can be revealed with “./enum4linux.pl -S <machine_ip>”

Question 4

Use smbclient to try to login to the shares on the Samba server. What share doesn't require a password?

```
(kali㉿kali)-[~]
$ smbclient //10.10.188.115/tbfc-hr
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
$ smbclient //10.10.188.115/tbfc-it
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
$ smbclient //10.10.188.115/tbfc-santa
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \>
```

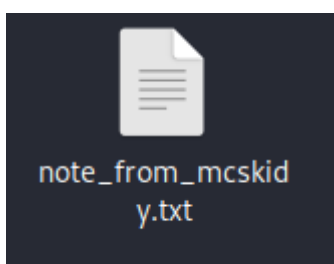
We login with the command “smbclient //machine_ip/sharename” .After trying to login on all the shares, “tbfc-santa” can be logged in without a password.

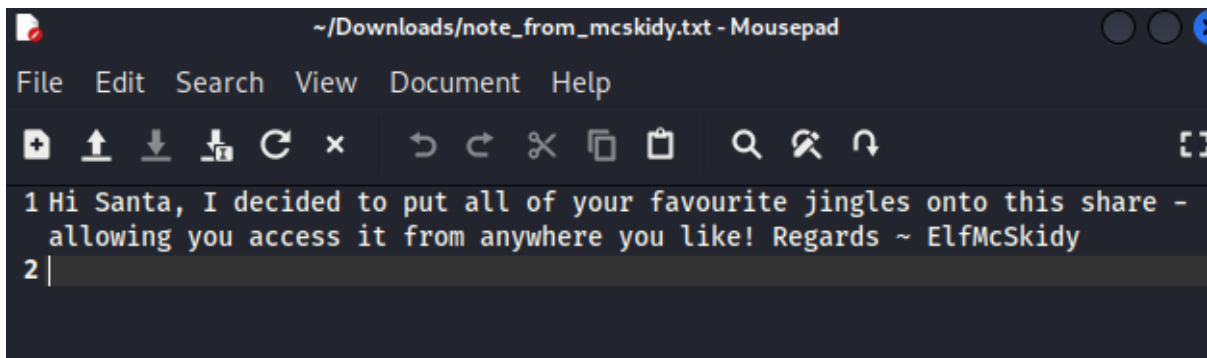
Question 5

Log in to this share, what directory did ElfMcSkidy leave for Santa?

```
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt

10252564 blocks of size 1024. 5369076 blocks available
smb: \> get note_from_mcskidy.txt
getting file \note_from_mcskidy.txt of size 143 as note_from_mcskidy.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
smb: \>
```





After logging into the “tbfc-santa” share, a “jingle-tunes” directory can be seen along with a .txt note by ElfMcSkidy.

Thought Process/Methodology:

We first installed enum4linux on the kali linux vm using the command “sudo apt install enum4linux”, then the enum4linux.pl file can be found in /usr/share/enum4linux. We first ran “./enum4linux.pl -h”, which shows the help message. To find the amount of users, running “./enum4linux.pl -U <machine_ip>” can show the user list and their usernames. The list of shares can be revealed with “./enum4linux.pl -S <machine_ip>”. We tried logging into each share with the command “smbclient //machine_ip/sharename”. We tried logging in without passwords for each of the accounts as suggested by THM. The sharename “tbfc-santa” can be logged in without a password. Inside, there is a “jingle-tunes” directory and a note from ElfMcSkidy to santa.