SECURITY ASSESMENT REPORT

Supervisor: Eng. Omar Al-Qraini

Author: Farah Mohammed

Table of Contents

Table of Figures

Executive Summary

I have been hired by a company which is compromised recently by an attacker who has encrypted some of their important files and asked for a ransom to give them the key to decrypt these files, the company gave me permission to do anything to the server without any restriction. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against the company with the goals of:

- ✓ Find as many vulnerabilities as possible in the system and different ways the attacker might have gained access to the system.
- ✓ Try to find the footprints of the attacker, and trace his attack and how he might have gained access to the system and encrypted the files.
- ✓ Recover the files by decrypting them

I have placed great efforts into the identification and exploitation of security weaknesses that could allow a hacker to gain unauthorized access to the data. These attacks were conducted with the level of the general Internet user might have. I found that the defenses of the company were vulnerable against many vulnerabilities, remote code execution which is highly dangerous to the safety of the company.
Recommendations were made to help eliminate and mitigate these vulnerabilities. In addition to the restoration of the encrypted data.

Scope of work

The scope of the penetration test was limited to the following targets:

- IP address (35.192.180.159)
- IP address (35.192.180.159:5000)
- 4 encrypted files
- Half encrypted file
- PCAP file (Wireshark file)

Project Objectives

The project objective was to identify vulnerabilities in the company server, web application and restore encrypted data. This was to be achieved by performing a penetration test on the given IP addresses and files.

Timeline

The penetration test took place on Friday, Saturday Jan 22nd & 23rd, from 12:30 pm till 6:30 pm.

Summary of Findings

after doing Nmap and looking carefully on the Wireshark file for the attacker footprint tracing, in total, I report 3 high impacts, 3 moderate impacts, and 1 informational issue during the exam of this penetration test.

Some high impact I have found in the company server is using SSH to run malicious codes and SQL-injection to find some sensitive information, such as usernames, passwords, and cookies sessions by unauthorized login for attackers and users and that I have used to recover the company important files.

Methodology

Planning

1. **Scanning**

Through the use of port scanners and vulnerability scanners, all sources were to be tested for vulnerabilities. The results would be analyzed to determine if there any vulnerabilities that could be exploited to gain access to a target host on a network.

2. **Source code reading**

The either all of the source code or only portions identified by Scanning were being analyzed for possible vulnerabilities.

3. **Obtaining Access**

Through the use of published exploits or weaknesses found in applications, operating system and services access would then be attempted.

Attack Narrative

The attack was divided into 3 main targets:

1. Server
2. Web application
3. Restoration of encrypted data

Part (1): Server Information gathering

For the purpose of this assessment, I was provided with only an IP address **(IP address: 35.192.180.159).** and a PCAP file. In an attempt to identify the potential attack surface, I scanned the IP address in order to identify open ports and their services, for that, I have used Nmap as a port scanner. (Figure 1.1-1.2).

```
farah@Farah: ~/Downloads
farah@Farah: ~/D...   farah@Farah: ~/D...   farah@Farah: ~/D...   farah@Farah: ~/D...
farah@Farah:~/Downloads$ nmap -A 35.192.180.159
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-22 12:33 EET
Nmap scan report for 159.180.192.35.bc.googleusercontent.com (35.192.180.159)
Host is up (0.17s latency).
Not shown: 995 filtered ports
PORT     STATE  SERVICE        VERSION
22/tcp   open   ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|    2048 f9:2c:eb:f7:88:75:16:ef:06:db:04:ac:3f:8c:28:7f (RSA)
|    256 a4:20:c4:1d:d5:be:68:78:4d:1a:03:2e:3e:8e:e2:8b (ECDSA)
|_   256 24:05:e4:84:38:66:70:b7:9e:6b:32:8c:7c:3c:9a:80 (ED25519)
80/tcp   open   http           Apache httpd 2.4.29 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/@dm1n1strator , /devnotes
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-title: Login Page
|_Requested resource was http://159.180.192.35.bc.googleusercontent.com/login
443/tcp  closed https
3389/tcp closed ms-wbt-server
5000/tcp open   upnp?
| fingerprint-strings:
|    DNSStatusRequestTCP, DNSVersionBindReqTCP, GenericLines, GetRequest, HTTPOptions, Help, NULL, RP
CCheck, RTSPRequest, SMBProgNeg, SSLSessionReq, ZendJavaBridge:
|       Welcome to online calculator
|       typing into this calculator ex: 1+1
|_      https://github.com/primaloptimum/calculator/tree/main
1 service unrecognized despite returning data. If you know the service/version, please submit the fo
llowing fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5000-TCP:V=7.91%I=7%D=1/22%Time=600AAA04%P=x86_64-pc-linux-gnu%r(NU
SF:LL,7C,"Welcome\x20to\x20online\x20calculator\x20\nTry\x20typing\x20into
SF:\x20this\x20calculator\x20ex:\x201\+1\x20\nhttps://github\.com/primalop
SF:timum/calculator/tree/main")%r(GenericLines,7C,"Welcome\x20to\x20online
SF:\x20calculator\x20\nTry\x20typing\x20into\x20this\x20calculator\x20ex:\
SF:x201\+1\x20\nhttps://github\.com/primaloptimum/calculator/tree/main")%r
SF:(GetRequest,7C,"Welcome\x20to\x20online\x20calculator\x20\nTry\x20typin
SF:g\x20into\x20this\x20calculator\x20ex:\x201\+1\x20\nhttps://github\.com
SF:/primaloptimum/calculator/tree/main")%r(RTSPRequest,7C,"Welcome\x20to\x
SF:20online\x20calculator\x20\nTry\x20typing\x20into\x20this\x20calculator
SF:\x20ex:\x201\+1\x20\nhttps://github\.com/primaloptimum/calculator/tree/
SF:main")%r(DNSVersionBindReqTCP,7C,"Welcome\x20to\x20online\x20calculator
```

Figure 1.1 - Information gathering using (Nmap)reveals available ports.

```
farah@Farah:~/Downloads$ nmap -A -T2 35.192.180.159
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-22 15:19 EET
Stats: 0:01:19 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 7.45% done; ETC: 15:35 (0:14:42 remaining)
Stats: 0:04:17 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 27.75% done; ETC: 15:34 (0:10:51 remaining)
Stats: 0:09:58 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 66.40% done; ETC: 15:34 (0:04:59 remaining)
Stats: 0:12:14 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 81.75% done; ETC: 15:34 (0:02:42 remaining)
Stats: 0:13:20 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 89.25% done; ETC: 15:34 (0:01:36 remaining)
Stats: 0:14:36 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 97.85% done; ETC: 15:34 (0:00:19 remaining)
Stats: 0:15:09 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.64% done; ETC: 15:34 (0:00:00 remaining)
Nmap scan report for 159.180.192.35.bc.googleusercontent.com (35.192.180.159)
Host is up (0.17s latency).
Not shown: 996 filtered ports
PORT     STATE  SERVICE        VERSION
22/tcp   open   ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|    2048 f9:2c:eb:f7:88:75:16:ef:06:db:04:ac:3f:8c:28:7f (RSA)
|    256 a4:20:c4:1d:d5:be:68:78:4d:1a:03:2e:3e:8e:e2:8b (ECDSA)
|_   256 24:05:e4:84:38:66:70:b7:9e:6b:32:8c:7c:3c:9a:80 (ED25519)
80/tcp   open   http           Apache httpd 2.4.29 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/@dm1n1strator , /devnotes
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-title: Login Page
|_Requested resource was http://159.180.192.35.bc.googleusercontent.com/login
443/tcp  closed https
3389/tcp closed ms-wbt-server
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 912.38 seconds
farah@Farah:~/Downloads$
```

Figure 1.2 - Information gathering using (Nmap -T2) reveals more available ports.

After identifying the ports, now I have 3 open ports, HTTP on port 80 and SSH on port 22 and there is port 5000 For unknown service. For that I will go through the webserver then I will go to the server again.

SSH version has a vulnerability **username Enumeration** So I have tried to exploit using Metasploit but I still needing more evidence to continue.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set USERNAME htu
USERNAME => htu
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS 35.192.180.159
RHOSTS => 35.192.180.159
msf6 auxiliary(scanner/ssh/ssh_enumusers) > exploit

[*] 35.192.180.159:22 - SSH - Using malformed packet technique
[*] 35.192.180.159:22 - SSH - Starting scan
[+] 35.192.180.159:22 - SSH - User 'htu' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 1.3 - SSH exploit using Metasploit

Part (2): Web Application

Information gathering

For the purpose of this assessment, I was provided with just an IP address **(IP address: 35.192.180.159).** In an attempt to identify the potential attack surface, I have scanned the IP address in order to identify any possible paths or existing vulnerabilities, for that I've used a gobuster as a web scanner.

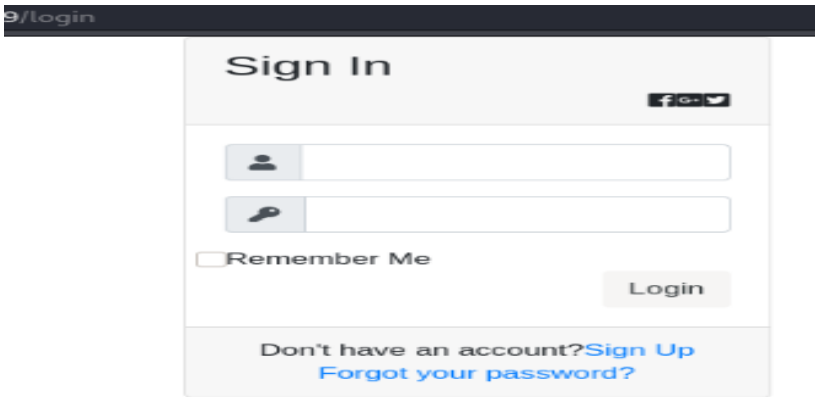


Figure 2.1 - Web Application-Login Page (the main page).

gobuster scan results not useful at all.

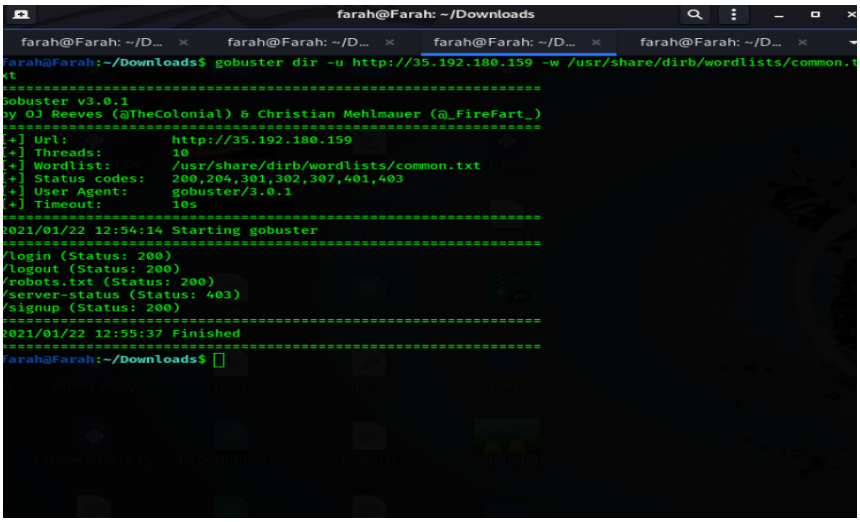


Figure 2.2 – Gobuster results.

So, I considered moving into the web page and started examining it using BurpSuite I have tried many things to have unauthorized access to the webpage but with no luck so I considered making an account and signup using [**username**: farah2 & **password**:farah123] and start checking the website and examine it using BurpSuite. XSS-Injection worked in the create post field but it wasn't that useful then I have tried to HTML-Injection it's worked perfectly but again nothing useful so I have used a BurpSuite to SQL-Injection on the searchById field and it's successfully worked.

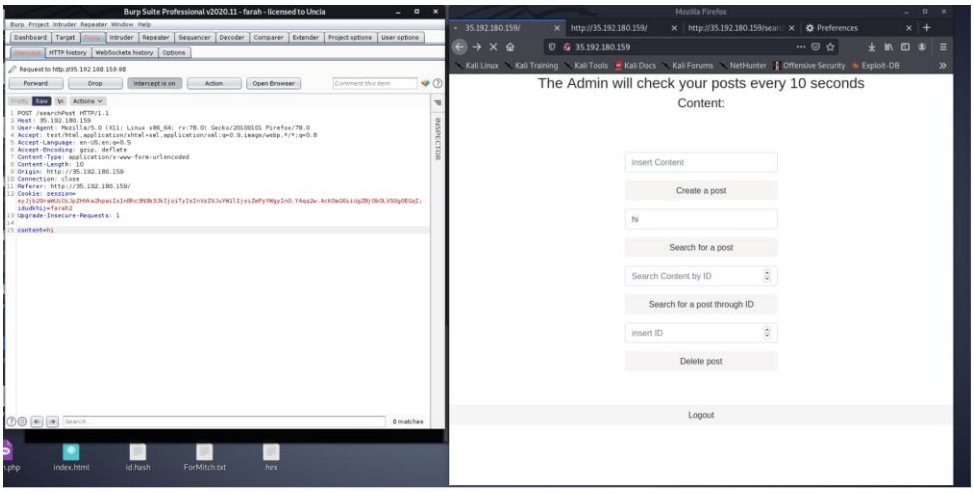


Figure 2.3 – Post Page.

I have used **10” Union select * from users /*** on the BurpSuite now I have all users and their data!

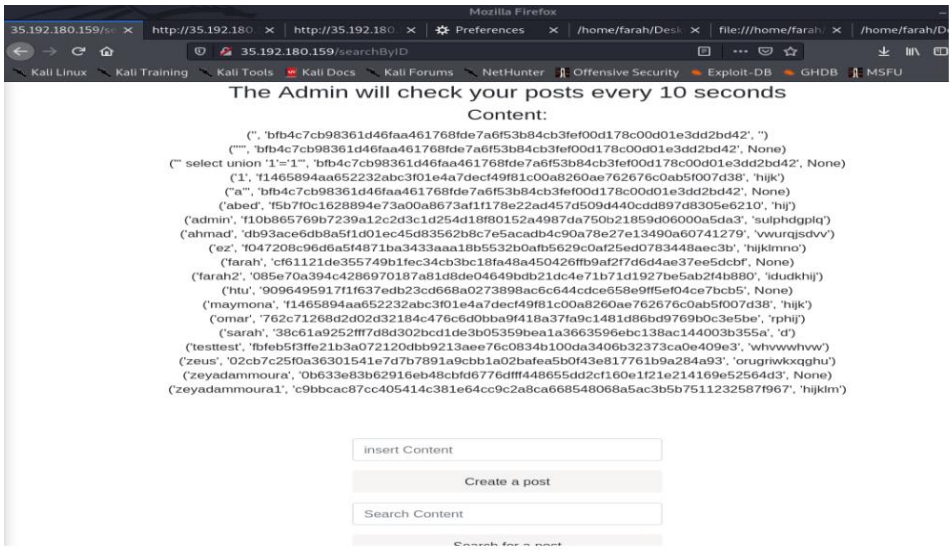


Figure 2.4 – First Attack SQL-Injection.

Now I have all users, their passwords, and their cookies.

After a long time of thinking and cracking the passwords and many failed attempts finally I have found the solution I have two accounts with two different passwords, in addition I have two users with their real passwords I have taken them from the Wireshark file (username=zeus & password=lordofthunder)(username=ahmad & password=strongpass) I have taken a look at the cookies they are somehow stranger for me so I have dug into them, I found the cookies is a **cipher shift by 29** of the passwords now I am able to login as Admin in the website.

I have written a python code cipher shift to get the password for “admin”.

[Username for webserver: admin **password**: primeadmin]

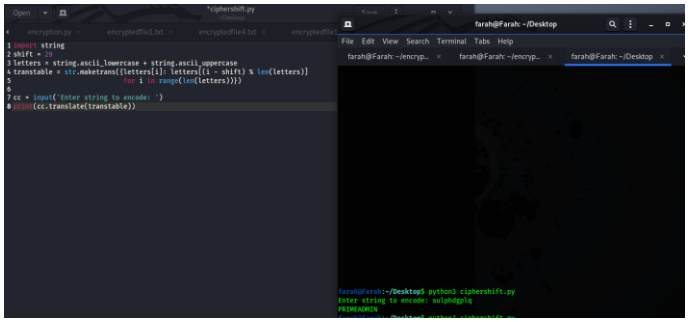


Figure 2.5 –python code to Find the admin Password.

Now I have special directory which I found on robots.txt on the website which I can login just As Administrator it's called "@dm1n1strator" so I have gone to it I have to change the User Agent to Uniquebot and I can access to this page.

I am also found the /devnotes.

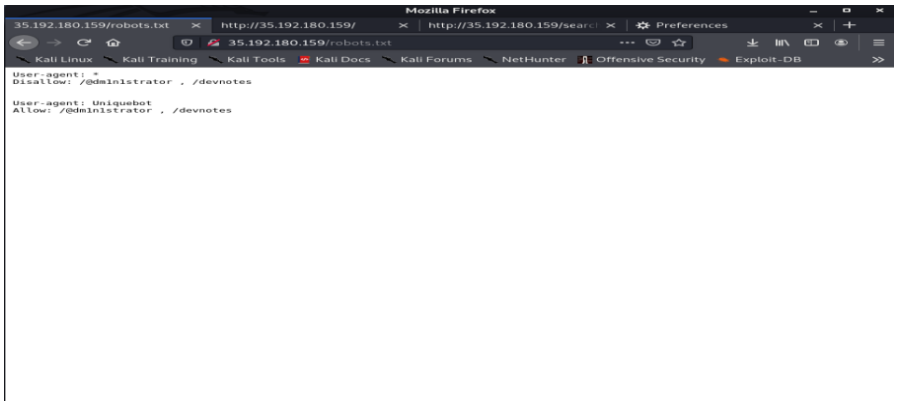


Figure 2.6 – hidden uncommon directories in robots.txt.

Till now nothing helped me to login to the SSH and log in as user or root, now I will go to the server again.

Part (1)-Step (2): Server

Port 5000 digging.

Port 5000 was open I have tried to open a reverse-shell through the website but doesn't work so I have a hint here "open NC throw the open port" that what exactly I have done. There is something called "Python-Flask template injection attack SSTI (python sandbox escape)" It's to use python functions for hacking so I have read about it. I can use it here because it replies to me when writing 1+1 it will give me 2. If I write msg it will resend msg to me so what if I send `__import__('os').popen('ls').read()`

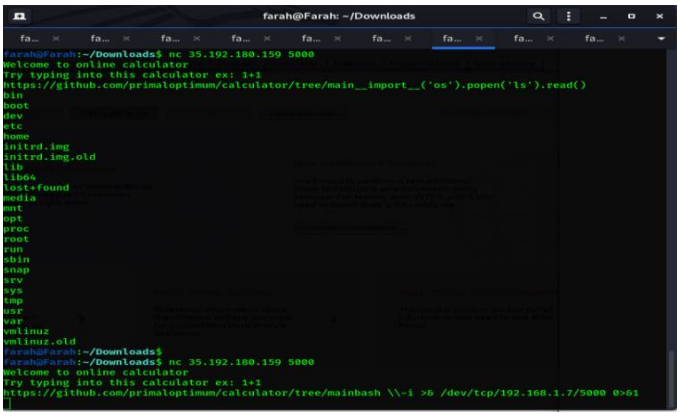


Figure 1.2.1 -using python command to show unwanted data.

so, if I write _import_('os').popen('cd home;ls').read() all users on the system will be shown **(htu, omars, ubuntu, webadmin, calculator ,gke-…..) now** my target is **htu** the root user so I've tried to login SSH as htu with "primeadmin" that's not correct so I've used webadmin user and primeadmin as password finally I am in.

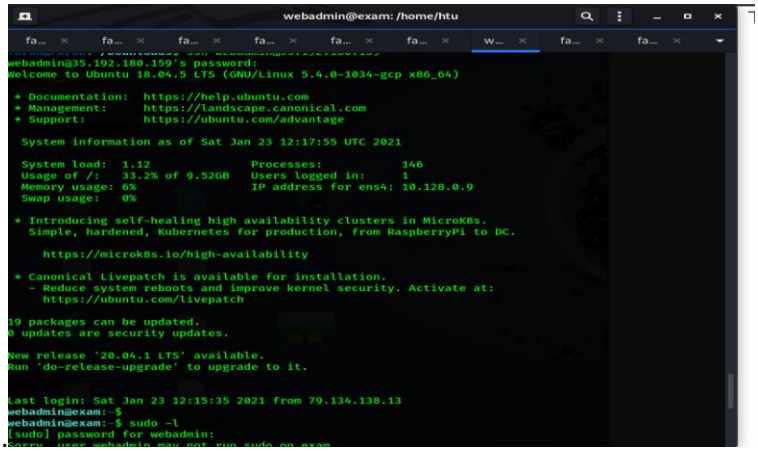


Figure 1.2.2 – Webadmin SSH successfully logged in.

I was checked what can I run as sudo on this user "**webadmin may not run sudo on exam**"

I need to privilege escalation so I have search if webadmin can write on any folder but nothing helpful.

So, I will dig for any note or something to run SSH as a root(htu). Finally, I've found something interesting, **"noteToHTU"**

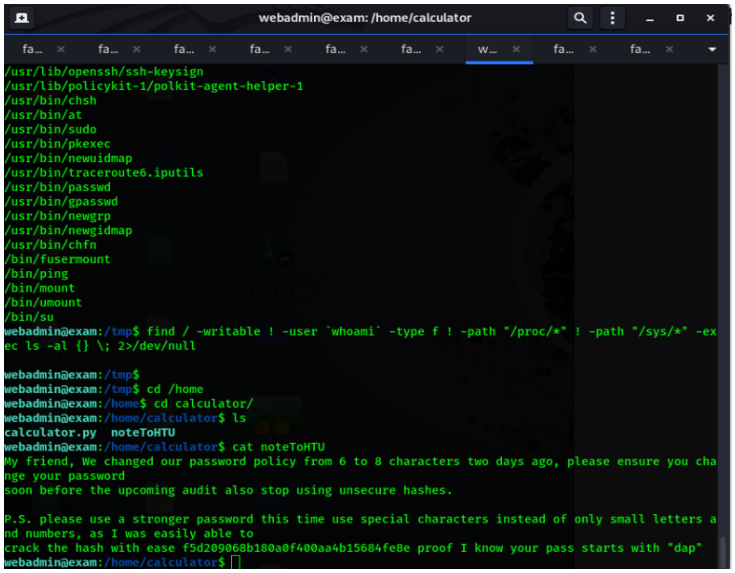


Figure 1.2.3 – Webadmin SSH noteToHTU.

Here I have found an interesting message, here there is hashed password for htu so it's time for hash Identifier.

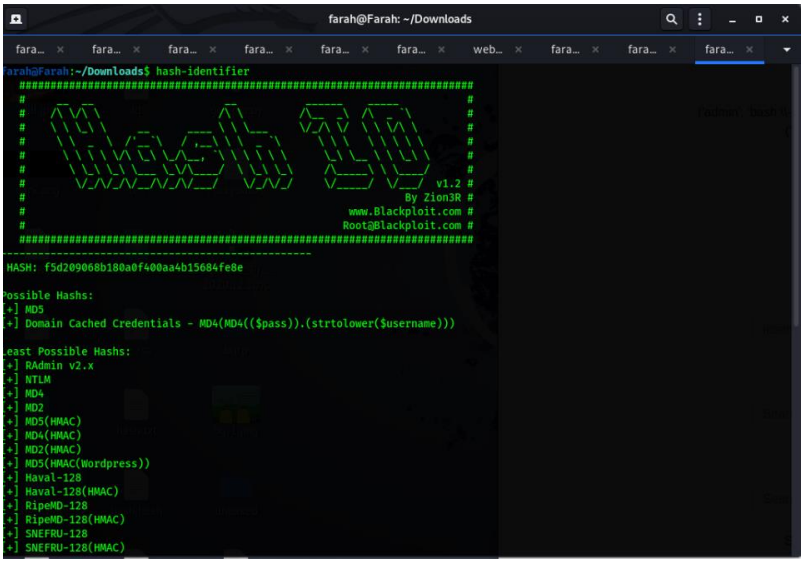


Figure 1.2.4 – identify the hashed password for htu.

It's md5 hash and I know that the first letter of the password is dap I have the rockyou.txt and the dap and hashcat the password is: **dapa55**.

Yes! Now I am the root(htu).

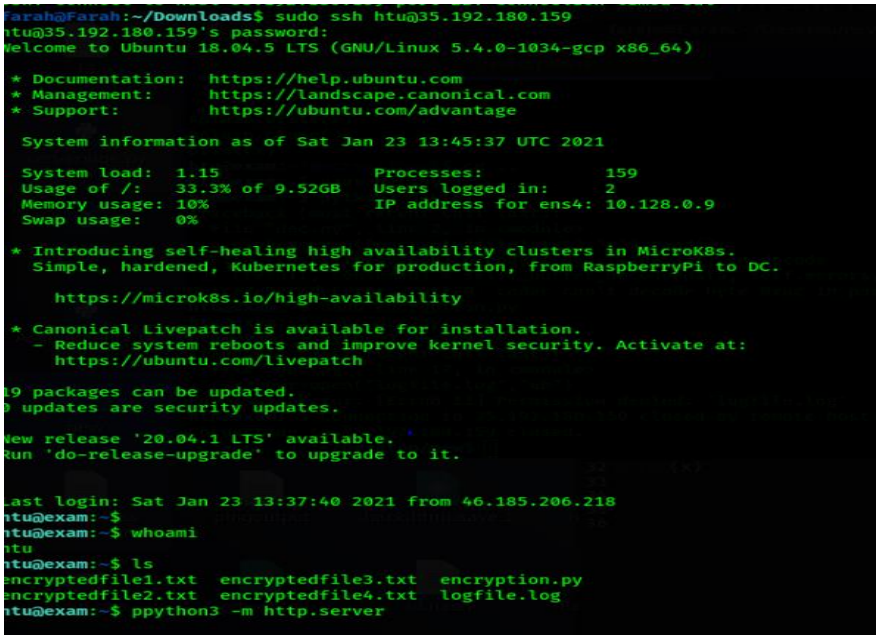


Figure 1.2.5 – htu SSH successfully logged in.

I have taken the important files that I should decrypt and find the key to help me decryption.

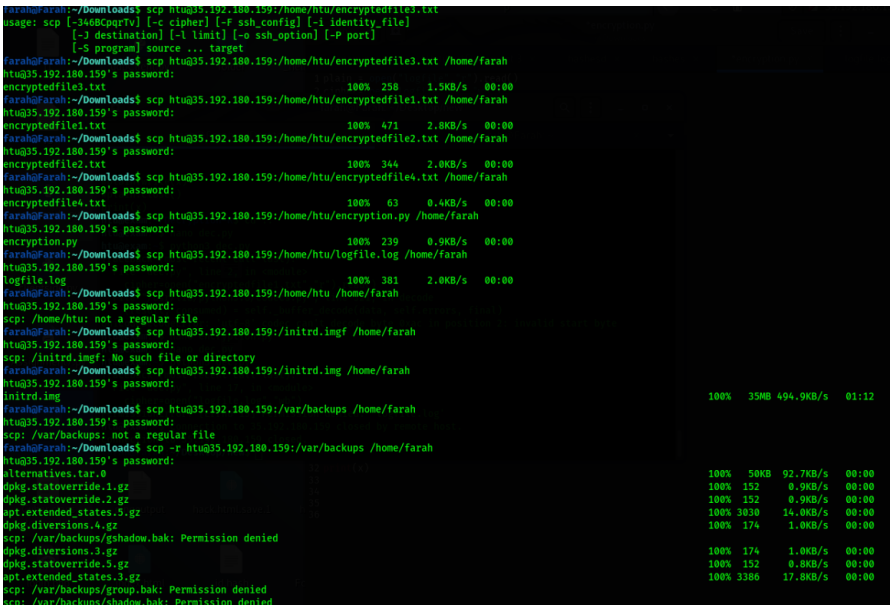


Figure 1.2.6 – transfer files from htu to my local machine.

Part (3): Restoration of Encrypted Data

Now I have 4 encrypted files 1 half encrypted file and 1 encryption code

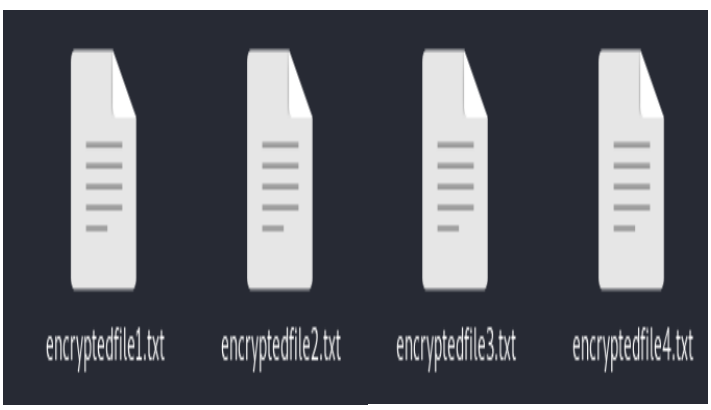


Figure 3.1-3.2 – 4 encrypted files and the python encryption code.

I have understood the python code well and known the way of cipher it's called the Caesar Cipher algorithm it has an encryption key and the key keeps rotate till the text in the plain text file end.

Before decrypting files, I have to write a python code to find the key because I have a logfile.log and it's half encrypted file. I have manually rewritten the expected original file and subtracted them.

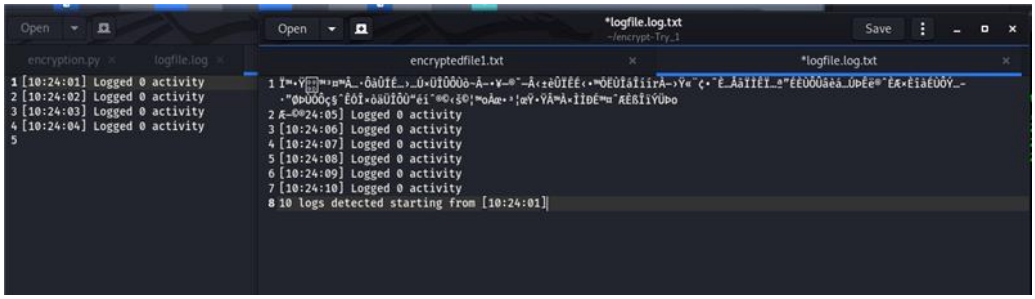
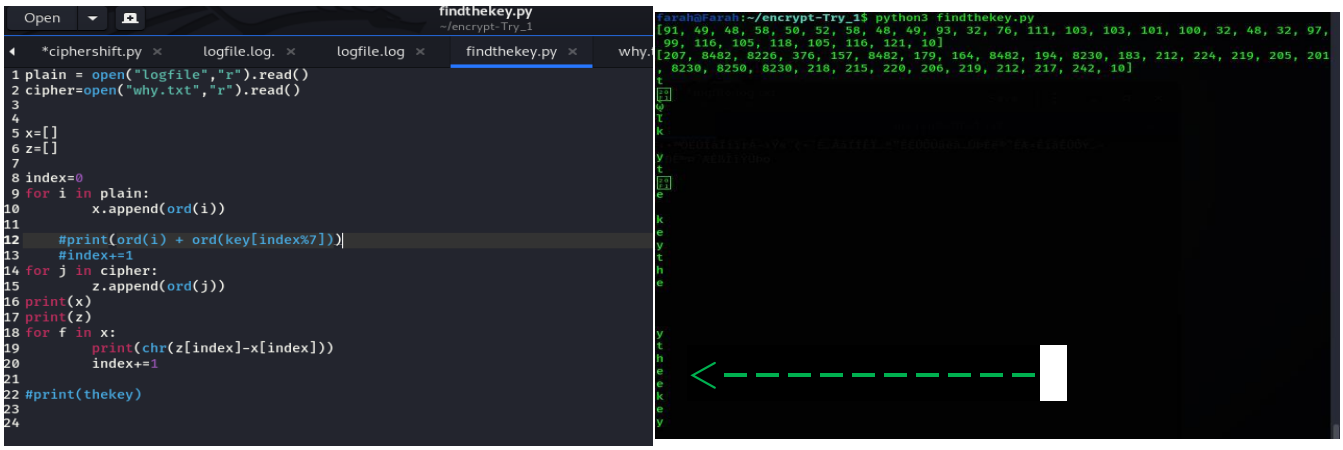


Figure 3.3 – half encrypted file and the expected value for the encryption data.

The key for decryption contained 7 characters and it's "theekey".



1Figure 3.4-3.5—find the key python code and successfully found the key.

Because **plain + key = cipher** **so** **cipher – key = plain** we can easily decrypting files using this code for decryption, in addition because I need a values in range of 0-256 so I need to module them to this number and the abs value give me a perfect result with no errors because chr function need a positive values and in that range

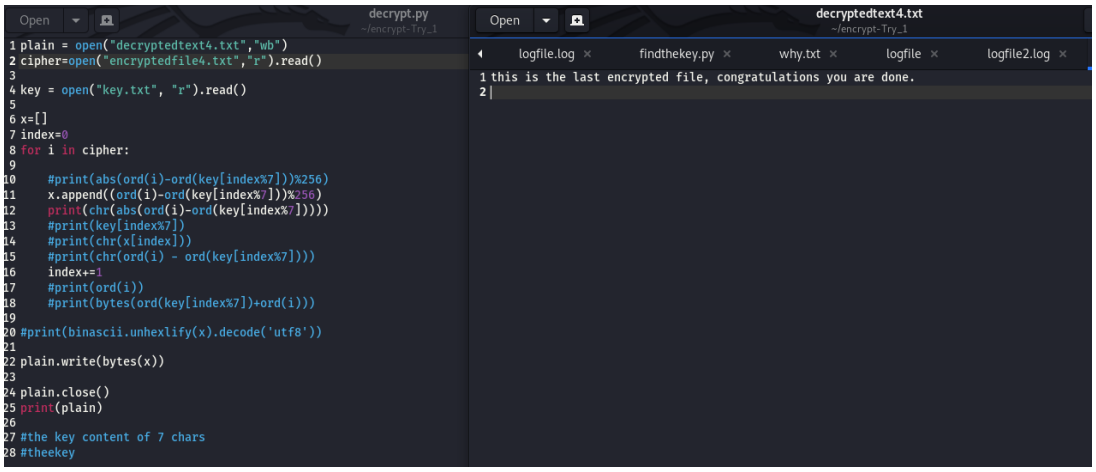


Figure 3.6 – python code to decrypt and successfully decrypted.

And by that, I've successfully restored these four important files to the company.

Attacker footprint:

The attacker with IP address(IP: 10.128.0.9) used an SQL-injection to get the user's info then he gets the admin page and he exploited the loophole on the google cloud platform environment to exploit the server he can get the machine data using it then get the user SSH and run malicious code on the target machine and encrypting files.[you can see the references page to read more about it]

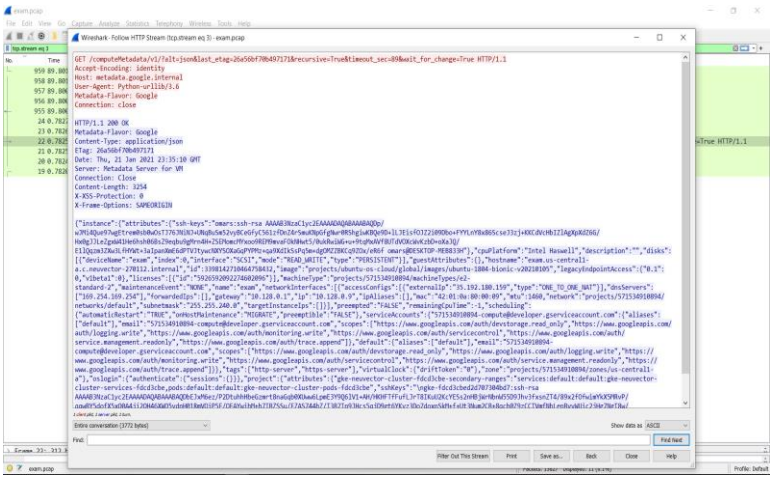


Figure 4.1 – loophole the attacker used to attack the machine.

Conclusion

The company suffered from a series of control failures, which led to a complete compromise of critical company assets. These failures could make a great effect on the company operations if a malicious attacker had exploited them.

- ✓ The specific goals of the penetration test were stated as:
- ✓ Identifying if a remote attacker could penetrate the company
- ✓ defenses.
- ✓ detect the attacker and show his attacking footprints via PCAP file.

- ✓ Restoring encrypted data
- ✓ Determining the impact of a security breach on:
    1. Confidentiality of the company's private data.
    2. Internal infrastructure and availability of the company information systems.

These goals were met successfully. Multiple issues that would typically be considered minor or not used anymore were leveraged in concern, resulting in a total compromise of the company information systems. Appropriate efforts should be undertaken to introduce effective network protection, which will help mitigate the effect of these vulnerabilities on this company.

Recommendations

Due to the impact on the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement. I recommend the following:

Because of the impact that might to the company which is uncovered by this penetration test, the company must put more effort to protect its system and accomplished it in a timely manner. So, I have some recommendations:

1- make sure to use strong passwords credentials, the system of the company highly impacted by the use of weak passwords.

2- make sure to close unnecessary open ports that lead to unexpected attacks.

3- Conduct regular vulnerability assessments: As part of an effective organizational risk management strategy, vulnerability assessments should be conducted on a regular basis. Doing so will allow the organization to determine if the installed security controls are properly installed, operating as intended, and producing the desired outcome. Please consult NIST SP 800-3011 for guidelines on operating an effective risk management program.

Risk Rating

The overall risk identified to the company as a result of the penetration test. A direct path from external attacker to full system compromise was discovered. It is reasonable to believe that a malicious entity would be able to successfully execute an attack against this company through targeted attacks.

Vulnerability Detail and Mitigation.

OpenSSH 7.6p1 ubuntu 4ubuntu0.3

Table 1: openSSH vulnerability

Rating	**Informational**
Description	Remotely observable behavior in auth-gss2.c in OpenSSH through 7.6p1 could be used by remote attackers to the detect existence of users on a target system when GSS2 is in use.
Impact	An attacker can bypass access restrictions to data via Username Enumeration of OpenSSH, in order to obtain sensitive information.
Remediation	Upgrade open-SSH to 7.8 and above

Apache 2.4.29

Table 2: Apache version 2.4.29 vulnerability

Rating	**Medium**
Description	Apache HTTP Server versions 2.4.20 to 2.4.43 A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards. Configuring the HTTP/2 feature via "H2Push off" will mitigate this vulnerability for unpatched servers
Impact	• Metasploit exploit • SQL-injection • XSS-Injection • HTML-Injection
Remediation	Upgrade the version

Broken Authentication

Table 3: Broken Authentication

Rating	**High**
Description	Broken authentication occurs when the application mismanages session related information such that the user's identity gets compromised. The information can be in the form of session cookies, passwords, secret keys etc.
Impact	The aim here is to either get into someone else's session or use a session which has been ended by the user or steal session related information.
Remediation	• Use of multifactor authentication. • Session isolation. • Idle session timeouts. • Using secured cookies.

Weak Password Policy

Table 4: Weak Password Policy

Rating	**Medium**
Description	The passwords are not that complex and there's no restrictions on making them complex.

Impact	• Lack of thought in creating password policies increases the chances of unauthorized access or compromised data. • The time it takes for an attacker to crack or brute force the password will reduce significantly.
Remediation	Creating a strong password policy.

Information Exposure Through Debug Information

Table 5: backend misconfiguration

Rating	**High**
Description	The application contains misconfiguration in the debugging code.
Impact	Exposure of sensitive information to untrusted parties.
Remediation	• Do not leave debug statements that could be executed in the source code. Assure that all debug information is eradicated before releasing the software. • Compartmentalize the system to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area.

Open ports

Table 6: open ports vulnerability

Rating	**High**
Description	Confidentiality: Open ports (actually the programs listening and responding at them) may reveal information about the system or network architecture. They can leak banners, software versions, content, the fact a system is there at all (instead of dropping the packet) and what type of system it is (for example, nmap can fingerprint systems). Rook's answer got me thinking about this. Integrity: Without open port controls, software can open any candidate port and immediately communicate unhindered. This is often relied upon by games, chat programs and other useful software, but is undesirable for malware. Availability: The network stack and the programs at open ports, even if the requests are invalid, still process incoming traffic. Even if electricity isn't an issue, technological solutions still have limited resources: degraded or denial of service results from finding a way to commit a port, network stack, computer, its hardware, network, or the people so they can't do much else.
Impact	An open port is an attack surface. The daemon that is listing on a port, could be vulnerable to a buffer overflow, or another remotely exploitable vulnerability.
Remediation	• An important principle in security is reducing your attack surface, and ensure that servers have the minimum number of exposed services.

Unencrypted communications

Table 6: Unencrypted communication

Rating	**Medium**
Description	The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.
Impact	To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.
Remediation	• Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection. • Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

References

1. Vulnerability OpenSSH via Username Enumeration
2. Storing and retrieving instance metadata
3. Linux - Finding a backdoor on a server
4. Apache Http Server 2.4.29 : Related security vulnerabilities
5. CVE - CVE-2020-9490
6. CVE-2020-9490 | Tenable®
7. privilege escalation and post exploitation tactics in Google Cloud Platform environments
8. NIST SP 800-30
9. CVE - CVE-2004-2172