Final Project
Penetration Test Report

Prepared by Kimmie DeBolt

December 12th, 2024

Table of Contents

Executive Summary

Kimmie DeBolt of DeBolt Security, LLC was contracted by No Security Corp to conduct a penetration test. The goal of this penetration test is to provide a detailed and comprehensive examination of the security controls in place. This test will assess the security stature of No Security Corp against a targeted attack and/or data breach and is intended to simulate the actions of a malicious actor. This test was performed in accordance with DeBolt Security's Penetration Testing Method. All testing was completed with permission from No Security Corp to:

- Test the FTP server, which is used to create and reload systems on No Corp Security's intranet.
- Ensure no classified or sensitive information resides on this FTP server.
- Ensure confidentiality of client information.

Priority was placed towards identifying vulnerabilities that would aid threat actors to gain access to classified or sensitive information should it exist in an accessible location.

Please find an appendix of relevant terms and attack details following the Execution and Results section.

Overall Posture

The overall security risk level of No Security Corp can be considered high to critical regarding a **CVE** score.

There is a medium to high probability of attack as the user information needed to successfully propagate an attack was easily accessible on the ftp server webpage.

Given the data classification of sensitive and confidential, in the event of a successful attack No Security Corp could face High to Severe Damage.

The CVE score rating is 7-10.

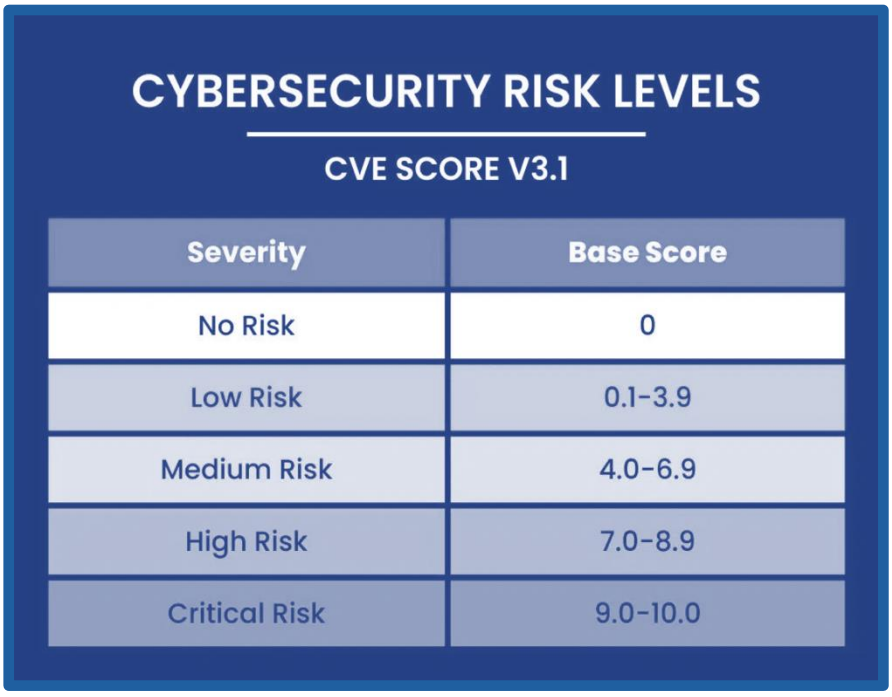


Figure 1. CVE Score Chart (Bocchino, 2022)

Summary of Results

Through network scanning and host discovery the FTP server's IP address was acquired as well as the services available. Further enumeration of the intranet led to the disclosure of user information. A vulnerability scan and review of the services running led to the discovery of a vulnerability that impacts the FTP protocol. Utilizing the information gathered thus far, the FTP server was breached, and system files were extracted.

Via the decryption of these system files confidential user information had become exposed. This information was then used to establish a user connection to the server via a communication protocol. Using administrator credentials, privilege escalation was achieved, and sensitive files were located, decrypted, and analyzed to reveal sensitive and confidential customer data.

Information Gathering

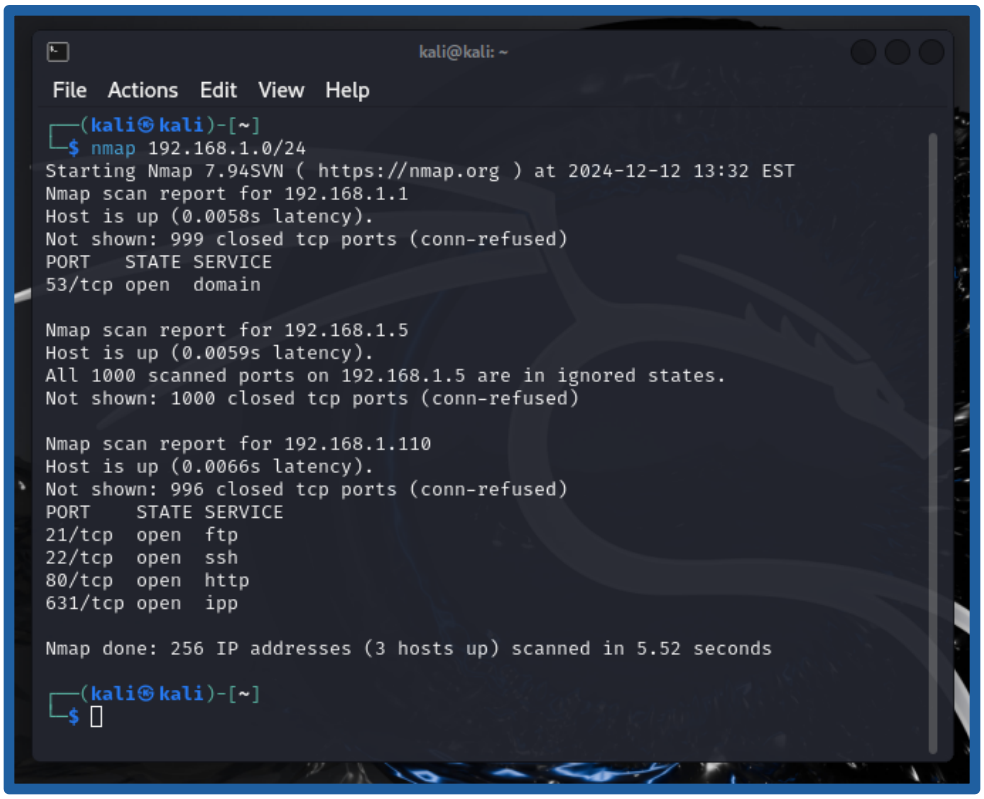


Figure 2. Network Scan

Utilizing **Nmap**, the network address of 192.168.1.0/24 was scanned. This revealed the host 192.168.1.110.

For further examination a multitude of Nmap scans were conducted. These included a **TCP**, **UDP**, service detection, and operating system detection scan were conducted.

This revealed the protocols being employed by the 192.168.1.110 address.

These protocols included FTP on port 21, SSH on port 22, HTTP on port 80, IPP on port 631.

The version of these services included: FTP - vsftpd 2.0.4, HTTP – Apache 2.2.4, and IPP CUPS 1.1

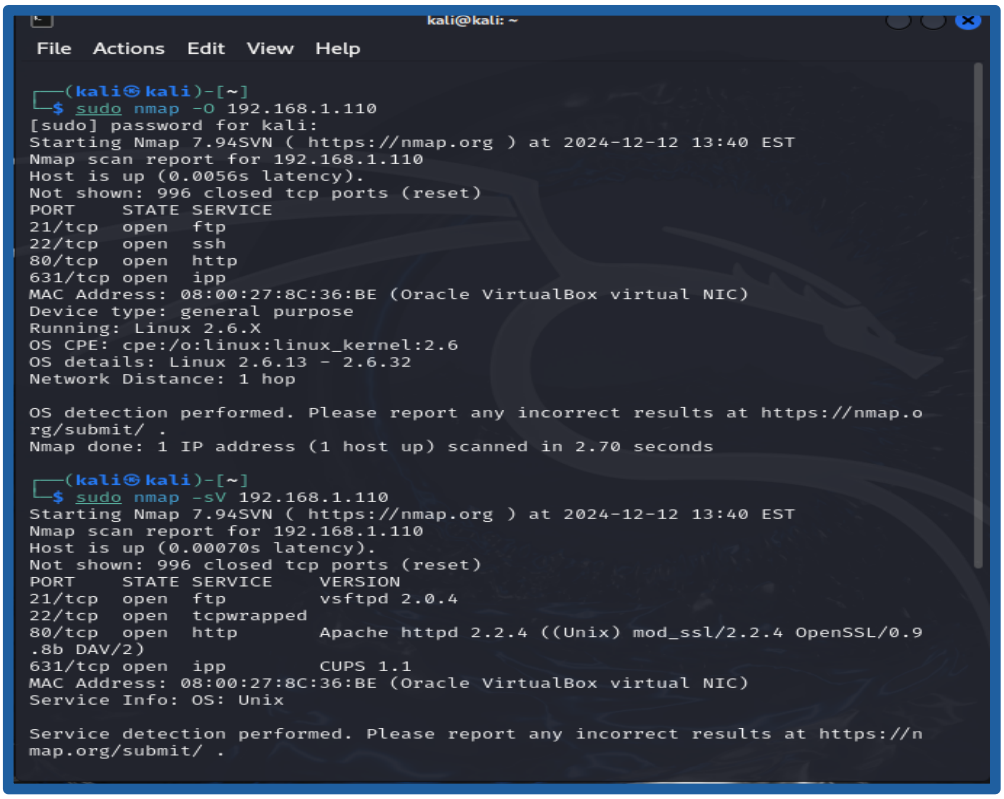


Figure 3. OS and Version Scan

Attack Narrative

Figure 4. No Security Corp.'s FTP Page

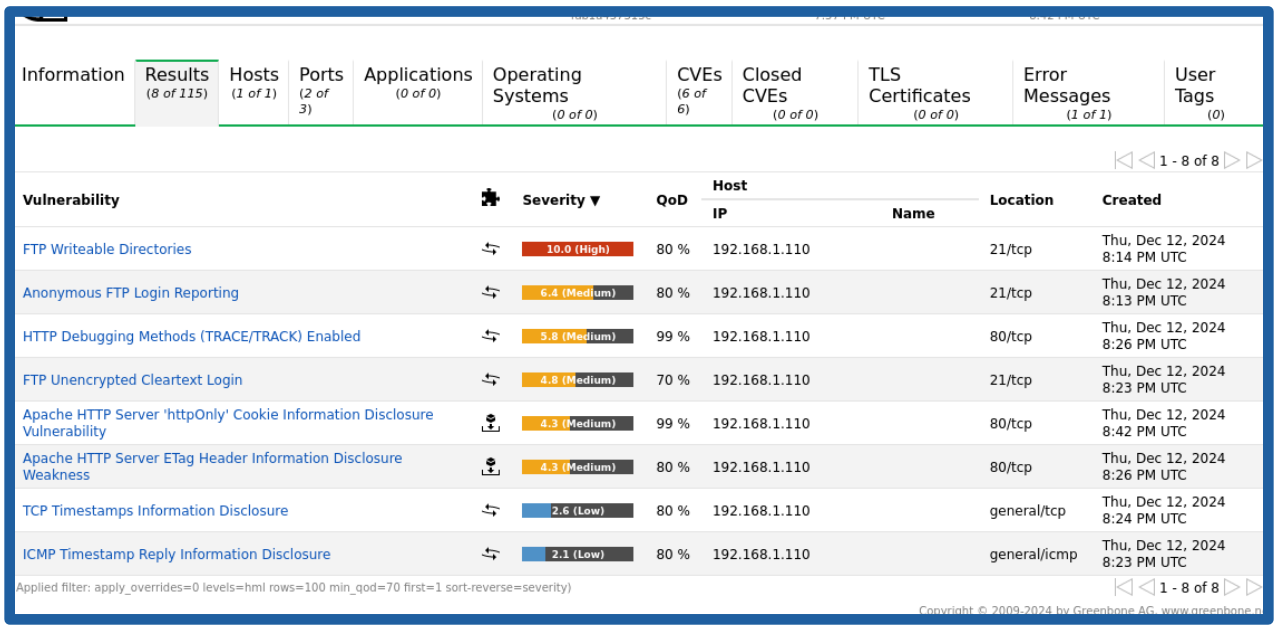
The IP address that was discovered was then input into FireFox for domain enumeration. Here we locate an informative page detailing a short description of the FTP server purpose and some contact information. However, this page reveals user information within the contact information it lists. From the descriptions of the employees, one may assume that not only do they have administrator privileges within the system, but also that they likely have access to sensitive information.

One can also deduct possible usernames from the email addresses listed above or at least some variation of them as listed below:

adamsa or aadams
banter or bbanter
coffeec or ccoffee
admin
user
guest

For good measure and for the sake of conducting a through penetration test, the emails were checked against the "haveibeenpwned.com" database to verify if they were exposed in prior data breaches. There were no results. The robots.txt and sitemap.xml were input as well; however, these pages did not exist, or they were not accessible to an outside actor.

For additional information, the OpenVAS Vulnerability Scanner was deployed. Visible here are some of the vulnerabilities the FTP server is at risk to, as well as their level of priority. Please notice the FTP vulnerabilities specifically, as they will be attempted to be employed and exploited during the attack phase.



Information | Results (8 of 115) | Hosts (1 of 1) | Ports (2 of 3) | Applications (0 of 0) | Operating Systems (0 of 0) | CVEs (6 of 6) | Closed CVEs (0 of 0) | TLS Certificates (0 of 0) | Error Messages (1 of 1) | User Tags (0)

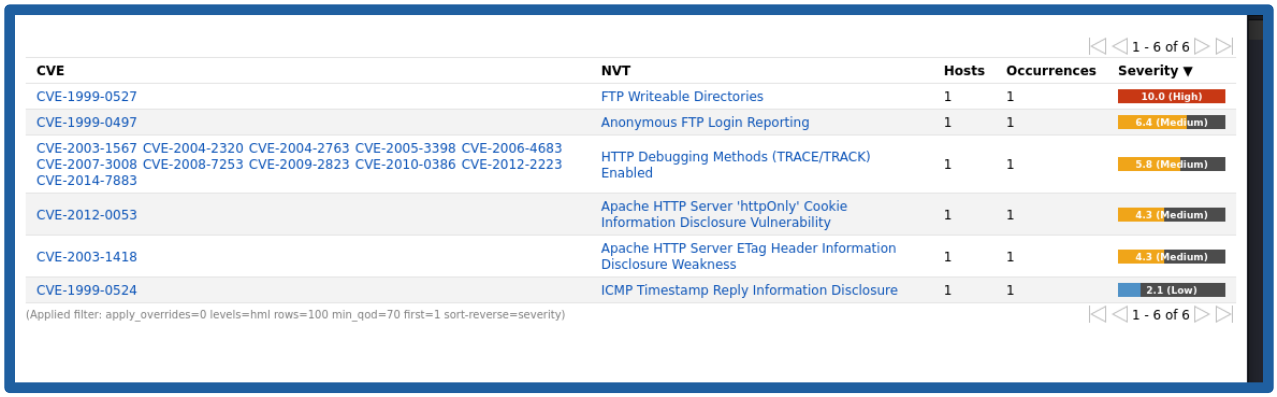
1 - 8 of 8

Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
FTP Writeable Directories	10.0 (High)	80 %	192.168.1.110		21/tcp	Thu, Dec 12, 2024 8:14 PM UTC
Anonymous FTP Login Reporting	6.4 (Medium)	80 %	192.168.1.110		21/tcp	Thu, Dec 12, 2024 8:13 PM UTC
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	192.168.1.110		80/tcp	Thu, Dec 12, 2024 8:26 PM UTC
FTP Unencrypted Cleartext Login	4.8 (Medium)	70 %	192.168.1.110		21/tcp	Thu, Dec 12, 2024 8:23 PM UTC
Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	4.3 (Medium)	99 %	192.168.1.110		80/tcp	Thu, Dec 12, 2024 8:42 PM UTC
Apache HTTP Server ETag Header Information Disclosure Weakness	4.3 (Medium)	80 %	192.168.1.110		80/tcp	Thu, Dec 12, 2024 8:26 PM UTC
TCP Timestamps Information Disclosure	2.6 (Low)	80 %	192.168.1.110		general/tcp	Thu, Dec 12, 2024 8:24 PM UTC
ICMP Timestamp Reply Information Disclosure	2.1 (Low)	80 %	192.168.1.110		general/icmp	Thu, Dec 12, 2024 8:23 PM UTC

Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 8 of 8

Figure 5. OpenVAS Report



1 - 6 of 6

CVE	NVT	Hosts	Occurrences	Severity ▼
CVE-1999-0527	FTP Writeable Directories	1	1	10.0 (High)
CVE-1999-0497	Anonymous FTP Login Reporting	1	1	6.4 (Medium)
CVE-2003-1567 CVE-2004-2320 CVE-2004-2763 CVE-2005-3398 CVE-2006-4683 CVE-2007-3008 CVE-2008-7253 CVE-2009-2823 CVE-2010-0386 CVE-2012-2223 CVE-2014-7883	HTTP Debugging Methods (TRACE/TRACK) Enabled	1	1	5.8 (Medium)
CVE-2012-0053	Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	1	1	4.3 (Medium)
CVE-2003-1418	Apache HTTP Server ETag Header Information Disclosure Weakness	1	1	4.3 (Medium)
CVE-1999-0524	ICMP Timestamp Reply Information Disclosure	1	1	2.1 (Low)

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 6 of 6

Figure 6. OpenVAS CVE List

Given the nature of the server, the FTP vulnerabilities were given priority and will be tested primarily before other methods.

CVE-1999-0527 (X-Force Vulnerability Report, 1999) allows for file manipulation by anonymous users. The writable directories with FTP servers can be utilized by threat actors to store manipulation or manipulate manipulation within the server.

CVE-1999-0497 (X-Force Vulnerability Report, 1993) allows for an anonymous user to access and ftp server. When utilizing the login: anonymous, the user only need to press enter when prompted for the password to receive access. Although the severity rating given here is medium, given other factors with this penetration test, including any sensitive data at risk, the severity is more likely to be high.

Execution

Now that possible usernames and the services being run by the domain have been discovered, it is pertinent to explore what vulnerabilities that can leveraged against them.

First, the anonymous ftp vulnerability will be employed to gain anonymous access to the server. Once the server is breached, the files will be traversed and examined until valuable information is discovered as shown below.

The valuable information in this case is within the **shadow** file, the hosts file and the **core** file which were acquired using the FTP get command.

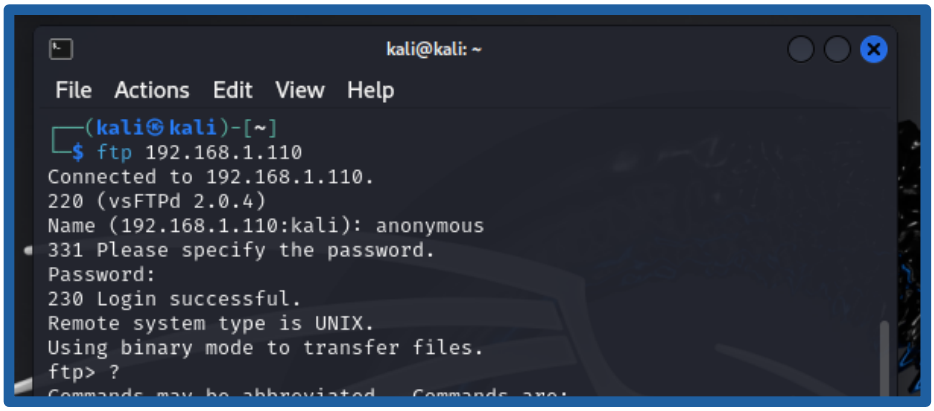


Figure 7. FTP Server Breached

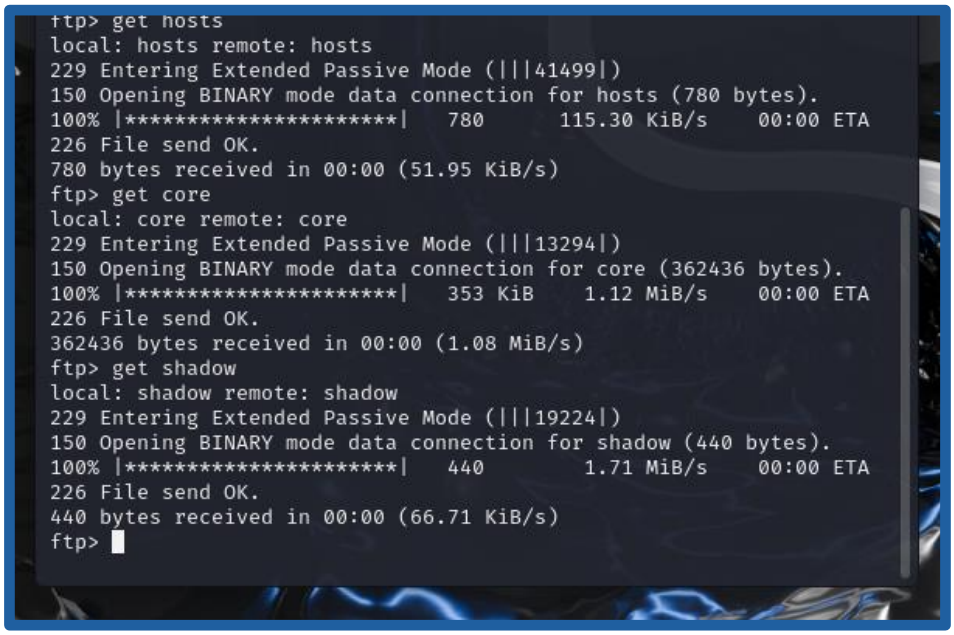


Figure 8. FTP Files Transferred to Remote Host

Using the cat command, the file contents were displayed. The core file was obfuscated; however, the strings command can be used to extract any information hidden inside. It is often a tool utilized in malware analysis to remove hidden information.

Within the shadow and core files, user password **hashes** were discovered These hashes included those for the admin users that were mentioned previously. There was no especially relevant information in the user file. The particularly important data is highlighted below.

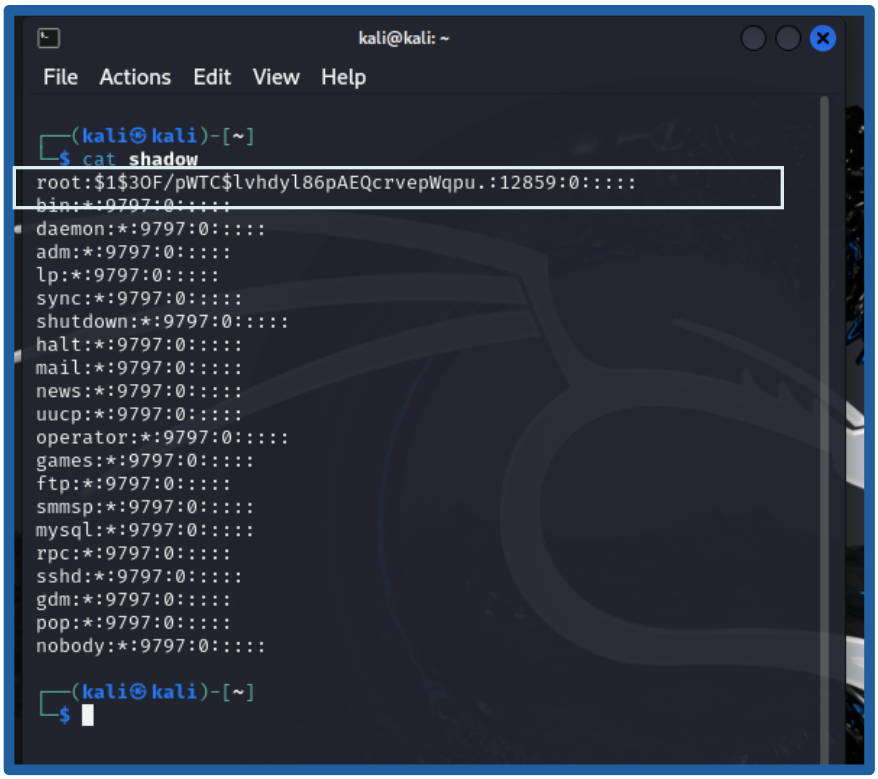


Figure 9. FTP Shadow File's Contents

Figure 10. FTP Core File's Contents

The contents of these files was extracted and placed into a file name Hashes.txt. In total, 5 hashes were discovered including:

1 root hash from the shadow file

1 root hash from the core file
3 admin user hashes from the core file

Now, using a decrypting tool, the hashes will be reverted to their plaintext password form. If successful, this will be employed to gain access to an admin account (as discovered in the information gathering phase). Below the tool utilized is an online tool (hashes.com) that uses a wide array of algorithms to decrypt encrypted passwords/hashes and identifies the hashing algorithm used. Not all hashing algorithms are created the same way or equally, so they are categorized and mathematically deconstructed. Below is a list of hashing algorithms as well as the output of the decryption tool.

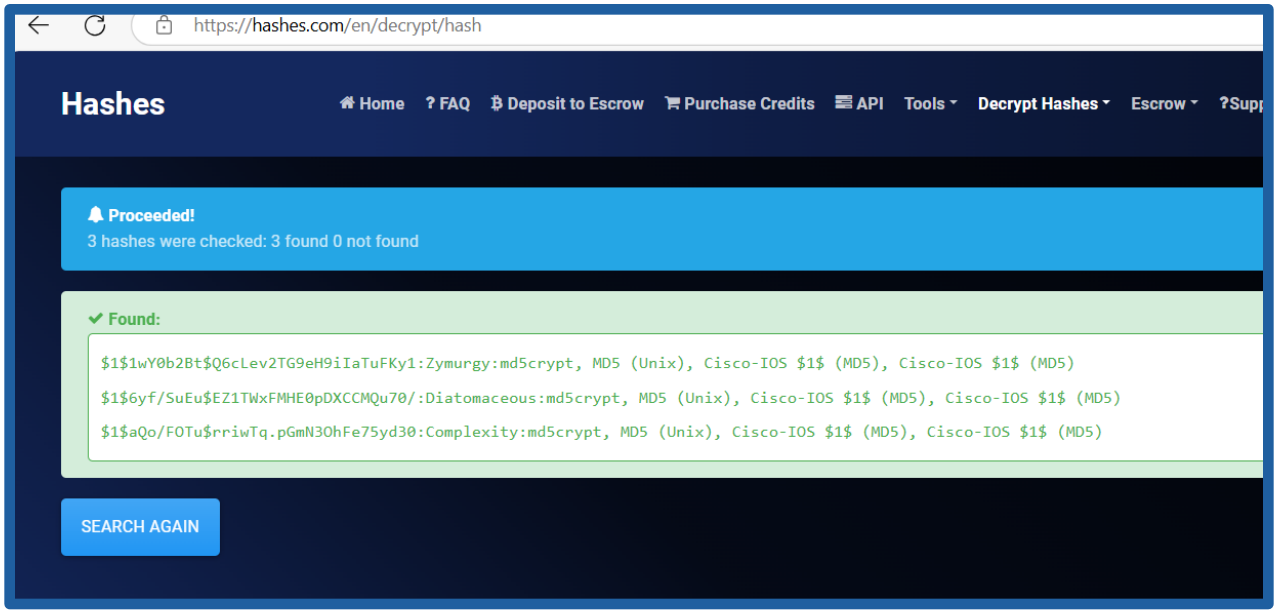


Figure 11. The Solved Password Hashes

https://hashcat.net/wiki/doku.php?id=example_hashes

Hash-Mode	Hash-Name	Example
0	MD5	8743b52063cd84097a65d1633f5c74f5
10	md5($pass.$salt)	01dfae6e5d4d90d9892622325959afbe:7050461
20	md5($salt.$pass)	f0fda58630310a6dd91a7d8f0a4ceda2:4225637426
30	md5(utf16le($pass).$salt)	b31d032cfdcf47a399990a71e43c5d2a:144816
40	md5($salt.utf16le($pass))	d63d0e21fdc05f618d55ef306c54af82:13288442151473
50	HMAC-MD5 (key = $pass)	fc741db0a2968c39d9c2a5cc75b05370:1234
60	HMAC-MD5 (key = $salt)	bfd280436f45fa38eaacac3b00518f29:1234
70	md5(utf16le($pass))	2303b15bfa48c74a74758135a0df1201
100	SHA1	b89eaac7e61417341b710b727768294d0e6a277b
110	sha1($pass.$salt)	2fc5a684737ce1bf7b3b239df432416e0dd07357:2014
120	sha1($salt.$pass)	cac35ec206d868b7d7cb0b55f31d9425b075082b:5363620024
130	sha1(utf16le($pass).$salt)	c57f6ac1b71f45a07dbd91a59fa47c23abcd87c2:631225
140	sha1($salt.utf16le($pass))	5db61e4cd8776c7969cfd62456da639a4c87683a:8763434884872
150	HMAC-SHA1 (key = $pass)	c898896f3f70f61bc3fb19bef222aa860e5ea717:1234
160	HMAC-SHA1 (key = $salt)	d89c92b4400b15c39e462a8caa939ab40c3aeeea:1234
170	sha1(utf16le($pass))	b9798556b741befdbddcbf640d1dd59d19b1e193
200	MySQL323	7196759210defdc0
300	MySQL4.1/MySQL5	fcf7c1b8749cf99d88e5f34271d636178fb5d130
400	phpass, WordPress (MD5), Joomla (MD5)	P984478476IagS59wHZvyQMArzfx58u.
400	phpass, phpBB3 (MD5)	H984478476IagS59wHZvyQMArzfx58u.
500	md5crypt, MD5 (Unix), Cisco-IOS 1 (MD5) [2]	$1$28772684$iEwNOgGugqO9.bIz5sk8k/
501	Juniper IVE	3u+UR6n8AgABAAAAHxxdXKmiOmUoqKnZlf8lTOhlPYy93EAkbPfs5+49Y
600	BLAKE2b-512	$BLAKE2$296c269e70ac5f0095e6fb47693480f0f7b97ccd0307f5c3bfa4
610	BLAKE2b-512($pass.$salt)	$BLAKE2$41fcd44c789c735c08b43a871b81c8f617ca43918d38aee6cf8

Figure 12. The Hashing Algorithm List Used

With the passwords for three of the accounts decrypted, the priority will be inputting the credentials into a ssh protocol session to gain access to the ccoffee account. The next step will include traversing the files and again exploring what directories can be found. Below list the files within this administrator directory. All accounts with decrypted passwords were also explored; however, the directories remained the same aside from user directories (which in this case, did not contain compelling information).

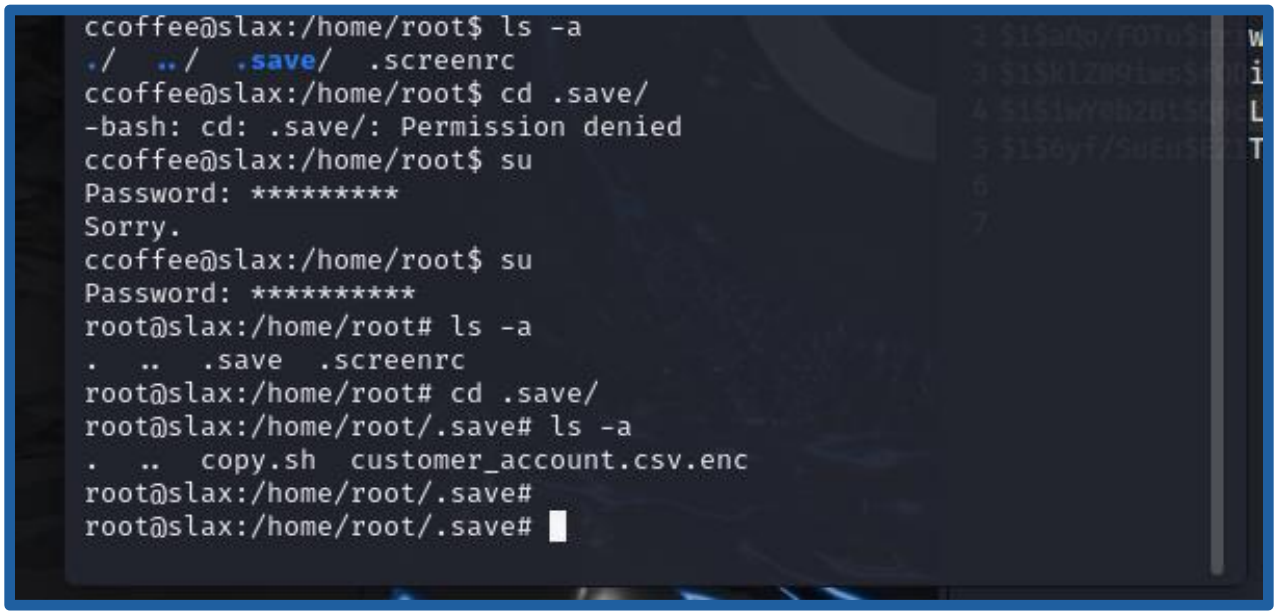


Figure 13. Ccoffee Administrator Account Directories

When traversing from the home directory to the save directory, the system denies the attempt as super user access is needed. This indicates that there may be some important data being protected.

Inputting the root password that was decrypted earlier will allow access to the save directory. Here, files names copy.sh and customer_account.csv are located. Upon displaying the contents of the latter file, it is apparent the content is encrypted. By displaying the contents of the prior file, the encryption method is revealed.

The following script will reverse the encryption process and print the new data to the highlighted file specified: OpenSSL enc -d -aes-256-cbc -salt -in customer_account.csv.enc -out customer_account-DECRYPTED.csv -pass file:/etc/ssl/certs/pw

Once decrypted, sensitive customer data is revealed as shown below.

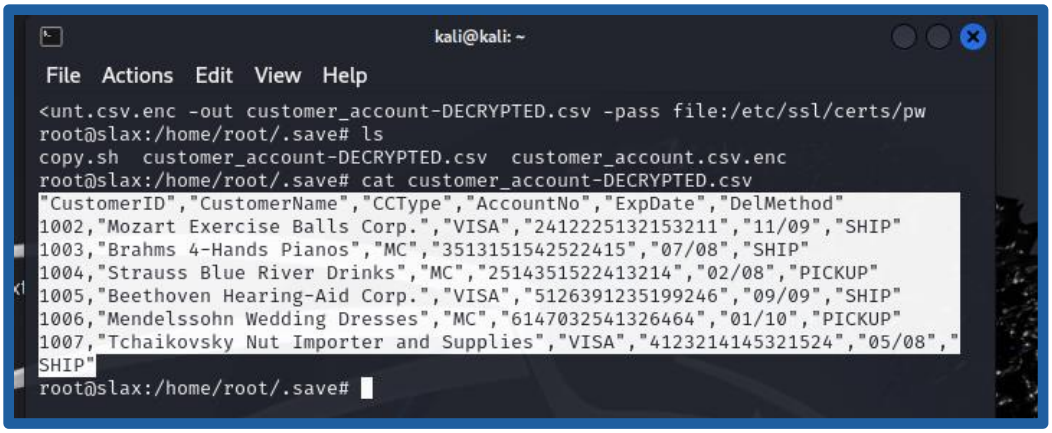


Figure 14. Customer Card Information

Results

This plan was not overly convoluted; however, there were a few “dead ends”. For example, the hosts file that is located via an FTP anonymous connection may seem like it may contain information but ultimately it acts as a red herring. To ensure no information was hidden within it, the file was analyzed with an abundance of caution.

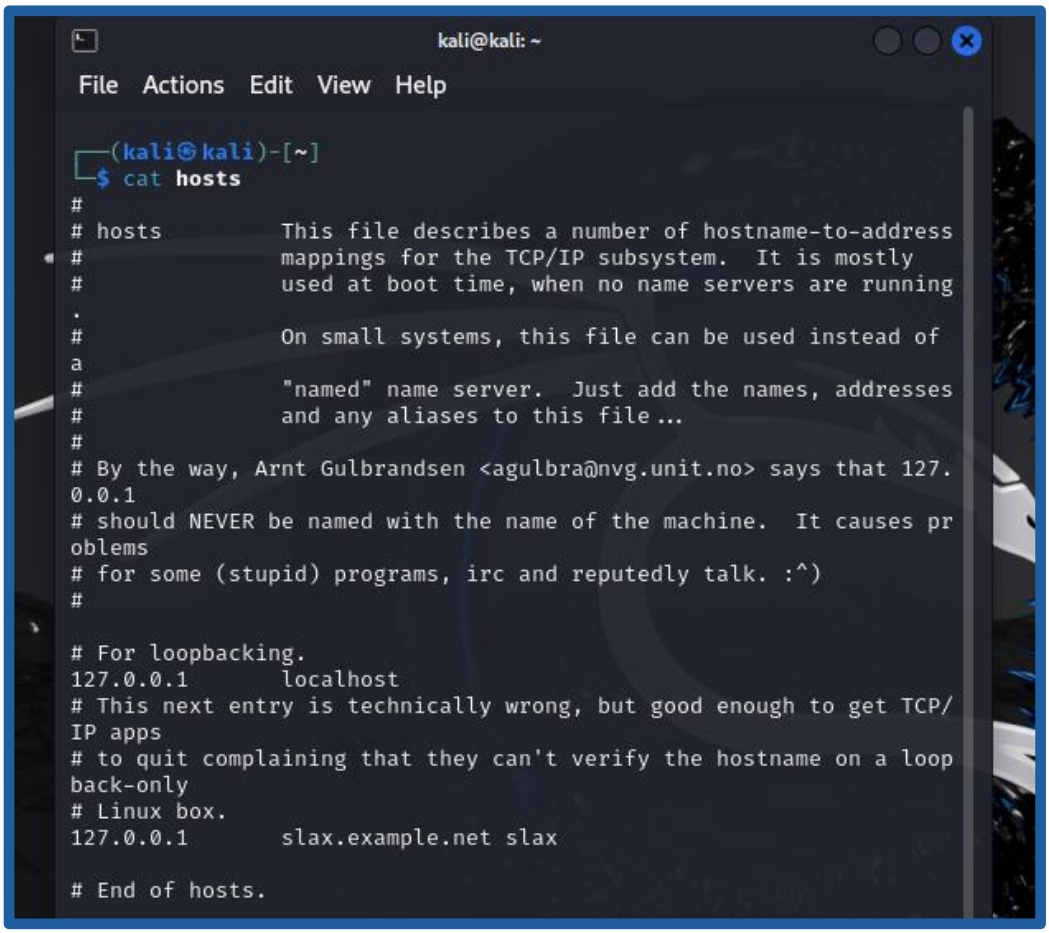


Figure 15. FTP Hosts File

Each step taken informed the next. There was emphasis on the FTP protocol, and this was reinforced through researching the server version after the completion of the Nmap version scan. Once the OpenVAS vulnerability scan was completed, the path forward becomes more obvious as the vulnerabilities and CVE’s provided further insight and confirmation.

Given experience with the prior penetration test of No Security Corp’s alternate server, the usernames and their format are similar.

With the above information established, some of the tools used prior were not so cooperative during this test: **Hydra, HashCat, and John the Ripper**. Unfortunately, as shown below, the tools were not responsive, there was not enough data within the virtual machine (regardless of adjustments), or they responded with incorrect data. While troubleshooting led to corrections and the tools functioning correctly, there are alternative tools showcased as well.

Any subsequent test plans would be adapted to include a wider array of tools, emphasizing those which were utilized successfully.

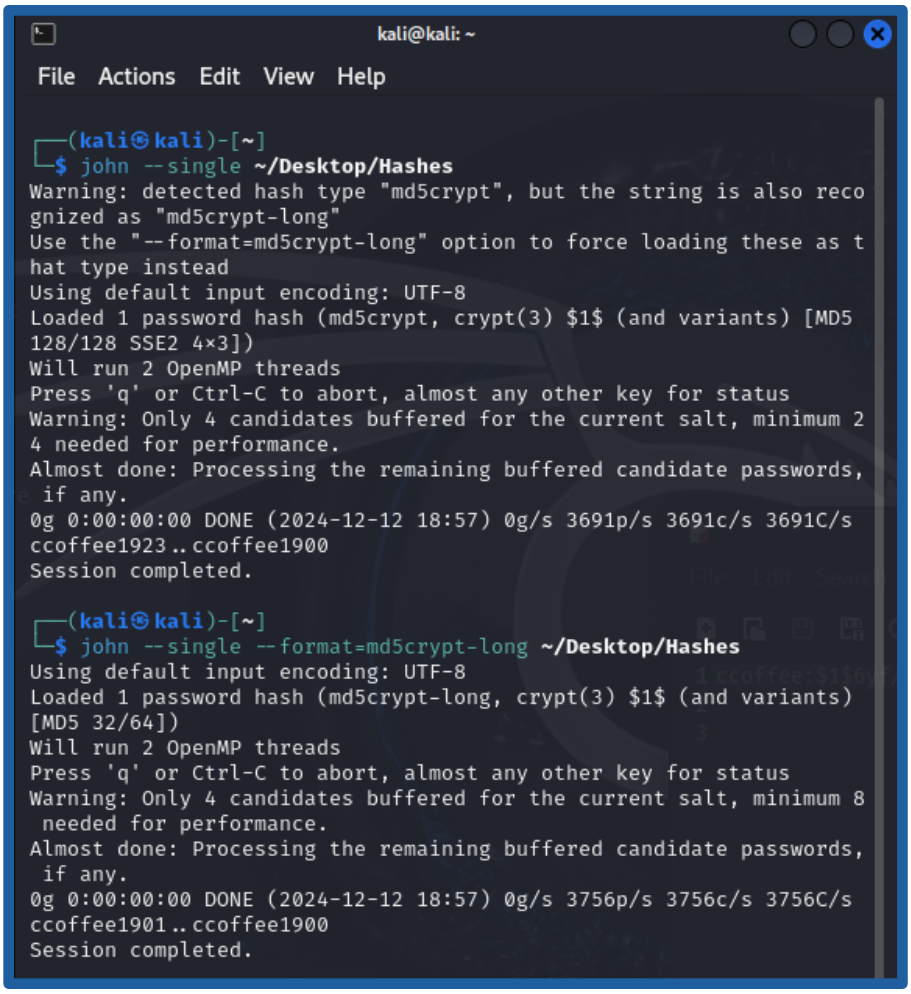


Figure 16. John the Ripper Incorrect Output

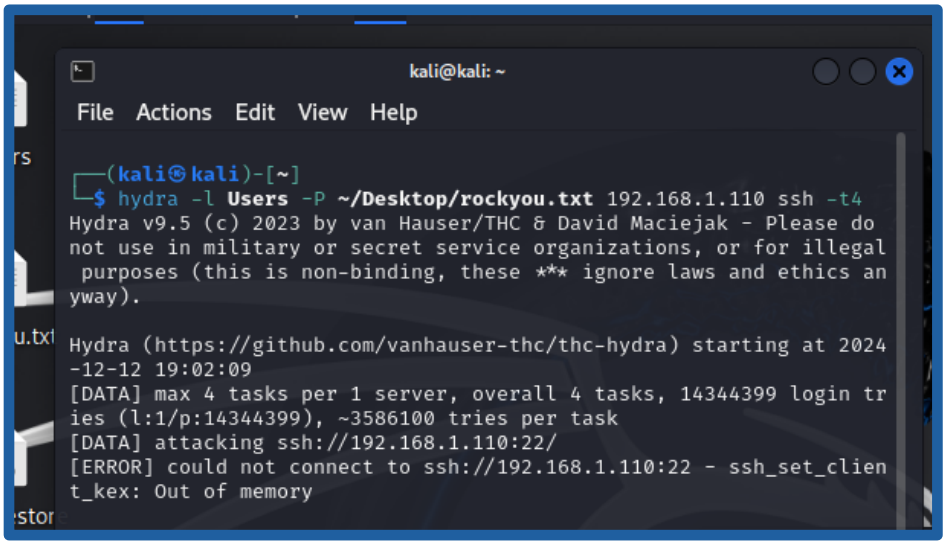


Figure 17. Hydra Memory Allocation Error

Overall, this penetration test was successful in locating and exploiting vulnerabilities to aid No Security Corp.'s security posture and locating confidential and sensitive customer information prior to a malicious threat actor.

With this report No Security Corp will be able to implement security measures they find appropriate against what has been provided here whilst also simulating and re-creating the attacks documented above.

Appendix A – Terms and Definitions

CVE – A CVE is a Common Vulnerability and Exposure. They are cyber security exploits, vulnerabilities, and exposures that are defines and scored on a scoring system to determine the threat level.
Nmap – A network scanning tool used for host and service discovery.
TCP – Transmission Control Protocol, it's utilized to reliably provide communication over an IP network.
UDP – User Datagram Protocol, it's also utilized to provide communication over and IP network; however, it is less reliable than TCP.
Shadow File – This Linux file typically stores hashed passwords.
Core File – This Linux file contains information that was utilized by a program before it was terminated or removed unexpectedly.
Hash – This is the result of a hashing algorithm which uses a mathematical and logical formula to turn a cleartext password into an encrypted string of characters.
Hydra, HashCat, and John the Ripper – These software's/tools are used to decrypt password hashes back into plaintext passwords by reversing their specified hashing algorithms.

Appendix B – Additional Attack Information

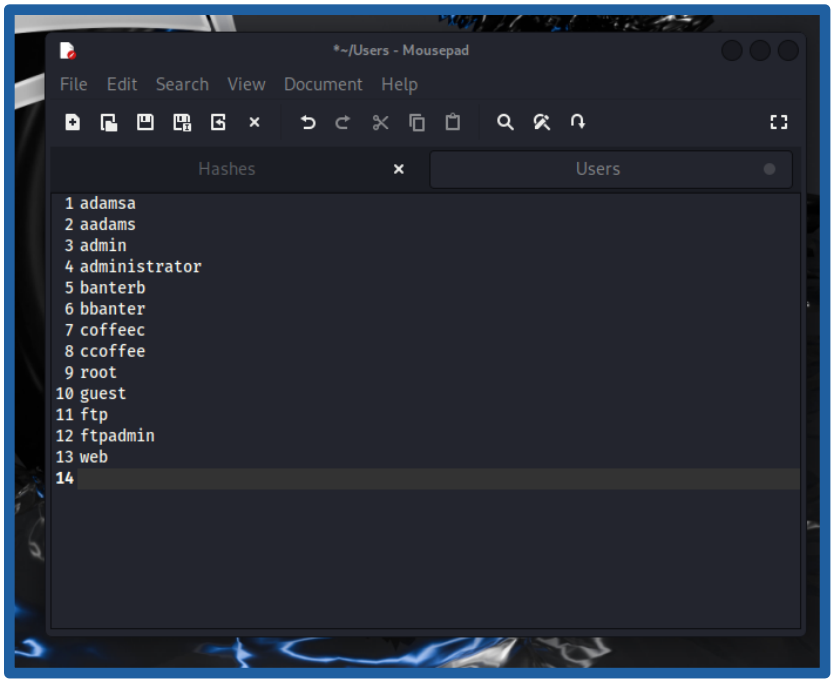


Figure 18. Potential Username List

This is a list of potential usernames to enumerate. Typically, a file like this could be created and utilized with a file containing hashes that had been located, along with a tool like HashCat or John the Ripper to discover passwords.

In this case, these attempts were unsuccessful due to host system limitation (via virtual box); however, this can be recreated in a more spacious environment.

This image depicts this file prior to decryption: customer_account.csv.enc. It is useful to note it's appearance and further enumerate the surrounding files for additional context.

Here, the script in the copy.sh folder was the key to reversing the encryption.

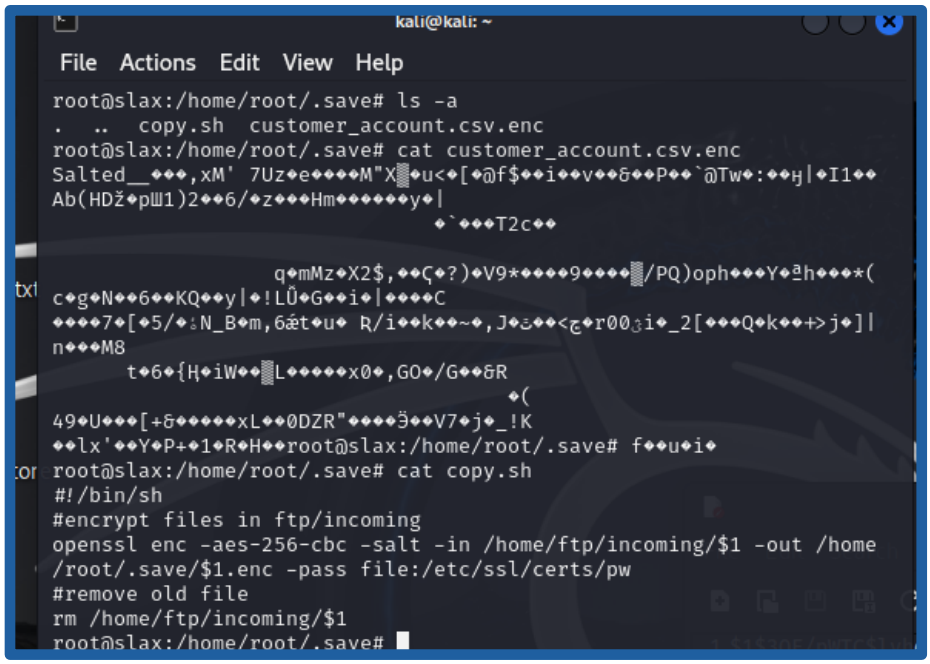


Figure 19. Customer Account and Copy File Output

Figure 20. Nmap -A Scan Output

When employing Nmap to gather information regarding the network, an Nmap -A scan was conducted.

This provided much of the same information that was gathered using the prior discussed scans; however, in one step.

Conducting the test this way will save time if re-creating results.

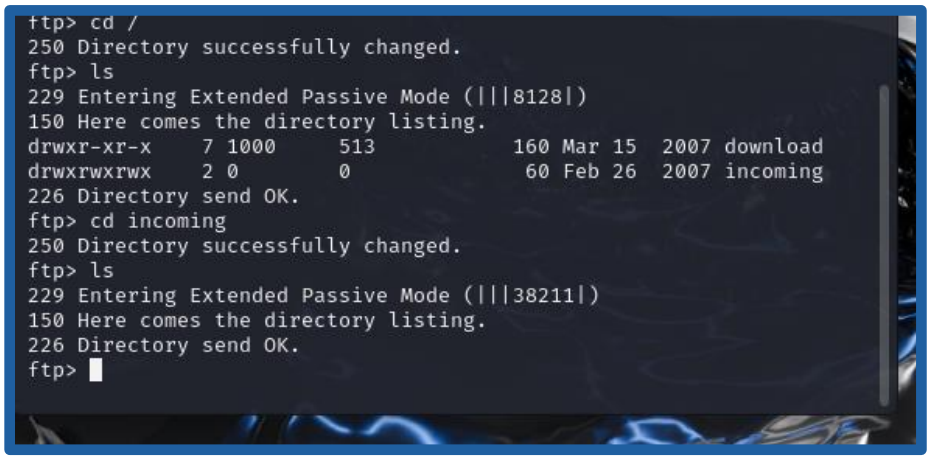


Figure 21. FTP Directory Listing

When the FTP server is breached initially with the anonymous account exploit, the files containing the hashes and user information can be found within the download directory as shown to the left.

After using the FTP get command to retrieve the files mentioned above (shadow, core, and user), the files will be transferred to the remote hosts directory. From there they can be moved, opened , and manipulated.

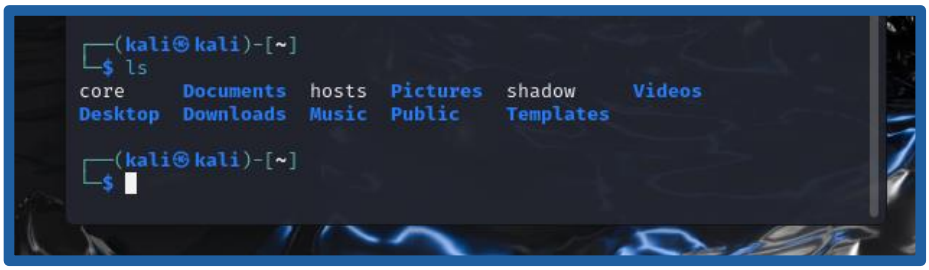


Figure 22. FTP Files Transferred to Remote Host

The hashes and password retrieved are as follows:

from shadow)
root:$1$3OF/pWTC$lvhdyl86pAEQcrvepWqpu.:12859:0:::::

(from core)
root:1aQo/FOTu$rriwTq.pGmN3OhFe75yd30:13574:0:::::: Complexity
aadams:1klZ09iws$fQDiqXfQXBErilgdRyogn.:13570:0:99999:7:::
bbanter:$1$1wY0b2Bt$Q6cLev2TG9eH9iIaTuFKy1:13571:0:99999:7::: Zymurgy
ccoffee:$1F$6yf/SuEu$EZ1TWxFMHE0pDXCCMQu70/:13574:0:99999:7::: Diatomaceous

References

Bocchino, S. (2022, August 17). *Cybersecurity Risk Levels: Where do you draw the line?* Retrieved from Webit: https://www.webitservices.com/blog/cybersecurity-risk-levels/

X-Force Vulnerability Report. (1993). *Anonymous FTP Users Engaging in Unauthorized Activities*. Retrieved from IBM X-Force Exchange: https://exchange.xforce.ibmcloud.com/vulnerabilities/543

X-Force Vulnerability Report. (1999). *FTP Server with World Writable Directories*. Retrieved from IBM X-Force Exchange: https://exchange.xforce.ibmcloud.com/vulnerabilities/6253