**Penetration Test Report for Victim4**

**-Sairam Bokka**

**VX05251**

**Sbokka1@umbc.edu**

**Contents**

1. **Executive Summary------------------------------------------------------------------------------ 3**
2. **Host Discovery ----------------------------------------------------------------------------------- 4**
3. **Information Gathering ------------------------------------------------------------------------- 5**
4. **Exploitation --------------------------------------------------------------------------------------- 7**
5. **Post Exploitation -------------------------------------------------------------------------------- 11**
6. **Privilege Escalation ----------------------------------------------------------------------------- 14**
7. **Recommendations ------------------------------------------------------------------------------ 15**

**Executive Summary:**

The victim PC runs an older version of Ubuntu OS. We discover the victim PC using a simple Nmap scan over the local network and discover the vulnerable machine. We do a thorough scan of the victim PC using Nmap and discover all the open ports of the victim PC along with version information of all the applications running on different ports of the machine. We do not see any vulnerable ports using Nmap. We go to the web application running on the machine. We see an interactive terminal that directs us to different pages, which you can see on the URL. Once we try a URL that it does not recognize, we see an error page which has a login page. Since this is a WordPress site, we try to gather more information by enumerating this website using wpscan. We get a dictionary file using which we brute force into the website using hydra. We get access to the admin page, and we install a malicious php plugin to get remote shell on out machine. To get root access, we search for application running on the machine with root privileges. We open Nmap on victim machine and we open interactive terminal on Nmap which has root privileges.

**Host Discovery:**

A screenshot of a computer

Description automatically generated

Discovered the host by doing a simple nmap scan over the network. The IP address of the host is 192.168.17.135.

**Information Gathering:**

1. **Scanning the host using nmap:**

A black background with white text

Description automatically generated

We start by doing a detailed port scan using nmap with version detection. The results show only three ports to be open which are:

1. 22/tcp -ssh closed
2. 80/tcp -http open
3. 443/tcp -ssl/https open

Port 80 indicates that there is a website running on this machine. Open the website by typing the IP address of the machine in a browser.

A screen shot of a computer

Description automatically generated

The website seems to be hosting an interactive shell on a web browser. If we type one of the commands as shown on the screen, it gets appended to the IP address. We need to see how the website is hosted. For that lets append something unexpected to see what kind of error page the website generates.

A screenshot of a computer

Description automatically generated

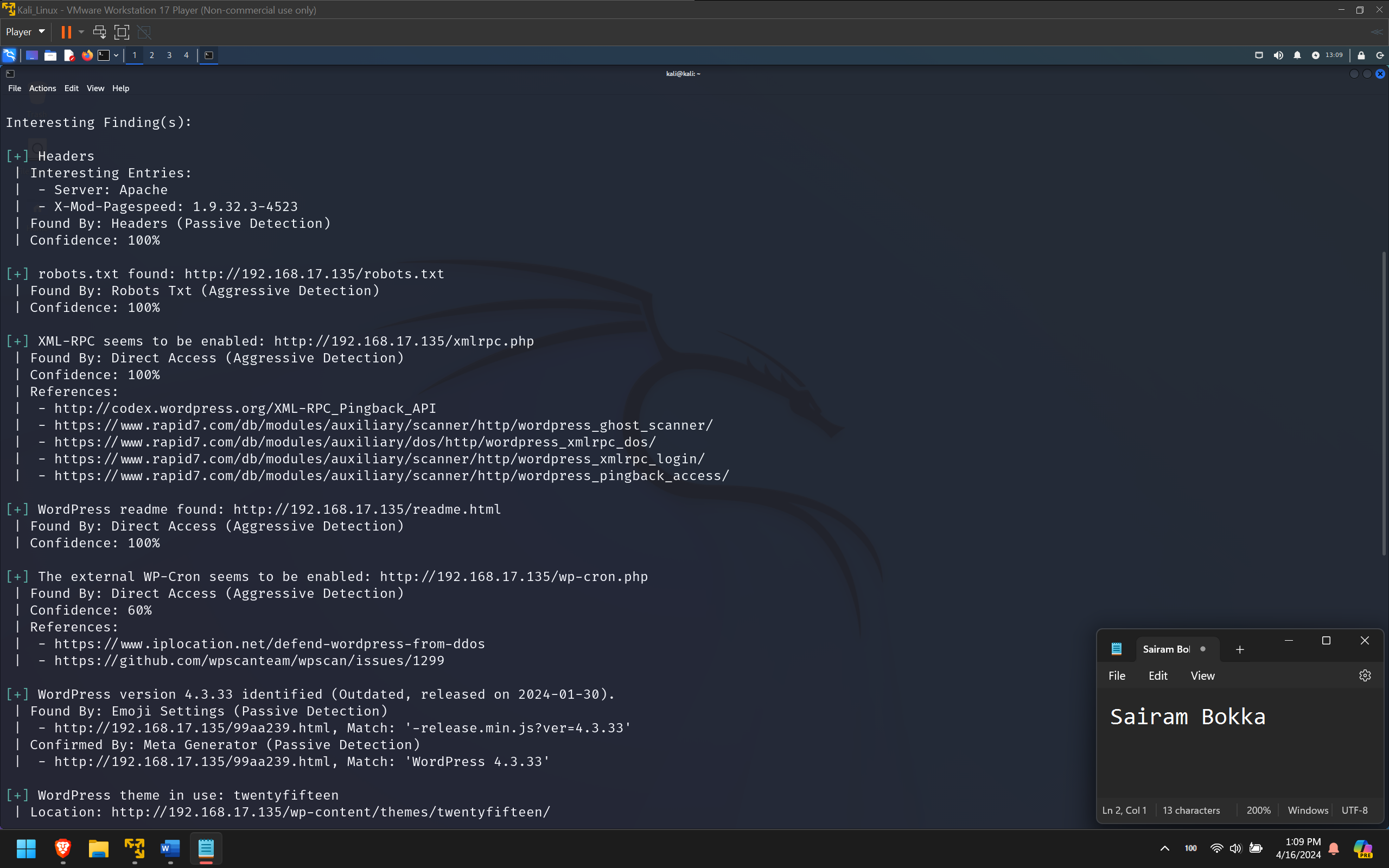
It looks like a wordpress website. It also has a login page.

A screenshot of a computer

Description automatically generated

**Exploitation:**

The login page generates an error if the username is incorrect. We need to find a username and password combination to login to the admin page. Since this is a wordpress site, let’s see what we can find by enumerating this website using wpscan.



We found three things:

1. There is a robots.txt file.
2. The wordpress version is outdated.
3. XML-RPC is enabled.

Let’s go into the robots.txt file.

A screenshot of a computer

Description automatically generated

We can see there is fsociety.dic file and key-1-of-3.txt file. By going into key-1-of-3.txt, we get one of the keys.

Let’s go into http;//192.168.17.135/fsociety.dic. we get the following dictionary file.

A screenshot of a computer

Description automatically generated

This might be useful to try and find a username and password combination for the login page we found earlier. The word list contains 858160 words and it seems to have repeated words. We can sort and create a shorter wordlist for our bruteforce attack by using sort -u fsociety.dic > fewer.txt

It sorts and creats a unique list of words. Our new file has 11451 words.

First step would be to find a valid username using our wordlist. For that we are going to use Hydra for bruteforcing our way through login page and find a page that says invalid password instead of username. We use the following command to do that.

**hydra -L fewer.txt -p randompassword 192.168.17.135 http-form-post "/wp-login.php:log=^USER^&pwd=^PASS:Invalid"**

This command uses the sorted list to find a response from the page that says invalid password.

A computer screen shot of a computer screen

Description automatically generated

We can see that we found three usernames from the bruteforce attack. They are Elliot, ELLIOT, and Elliot. Now, we can use these usernames and the sorted list again to find the password associated with atleast one of the usernames.

A computer screen with text

Description automatically generated

By using hydra again, we can see that, we have a successful login using username Elliot and password ER28-0652. Using this, we can successfully login to the wordpress login page we discovered.

A screenshot of a computer

Description automatically generated

By logging in, we have access to a lot of data. We can see the users table, which has users and roles. Here we can add another user and give him administrative access.

A screenshot of a computer

Description automatically generated

We also can see a lot of installed plugins which are not activated. After doing research on these plugins, there were no vulnerabilities to be found, but we have ability to install our own plugins, which means we can create a malicious plugin to gain shell access. We craft a php-reverseshell plugin to gain access to shell.

**Post Exploitation:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Now, we have the access to shell. Here we need to investigate for sensitive data.

A screenshot of a computer

Description automatically generated

We have two things here. An md5 hash and a key which we don’t have access to. The MD5 hash c3fcd3d76192e4007dfb496cca67e13b was successfully reversed into the string abcdefghijklmnopqrstuvwxyz.

Using this password, we can try to login as user robot, this password may be associated with the user robot. Now, to do that, we need a terminal which we can spawn using python.

A screenshot of a computer program

Description automatically generated

As we can see, python is installed and we can use the command: python -c “import pty;pty.spawn(‘/bin/bash’);” to spawn a terminal. After spawning the terminal, we can login as robot using command su robot and password which we discovered earlier.

Now we can access the key-2-of-3.txt which is 822c73956184f694993bede3eb39f959.

**Privilege Escalation:**

In order to access sensitive information on this machine, we need root access. After digging a little bit we found nmap is running with root privileges on this machine. Nmap has interactive mode, which we can use to spawn a shell using !sh command in nmap.  
A screenshot of a computer

Description automatically generated

We got access as root user using nmap as shown.

After going through the root directory, we got the last key-3-of-3.txt which is 04787ddef27c3dee1ee161b21670b4e4

**Recommendations:**

1. **Install Wordpress updates**

Some common brute force attacks actively target known vulnerabilities in older versions of WordPress, popular WordPress plugins, or themes. WordPress core and most popular WordPress plugins are open source, and vulnerabilities are often fixed very quickly with an update. However, if you fail to install updates, then you leave your website vulnerable to those old threats. Simply go to the Dashboard » Updates page in the WordPress admin area to check for available updates. This page will show all updates for your WordPress core, plugins, and themes.

1. **Disable XMLRPC:**

Disabling xmlrpc. php can help protect your WordPress site from DDoS attacks and brute force login attempts. It reduces the attack surface and minimizes the risk of unauthorized access.

1. **Install a WordPress Firewall Plugin:** Brute force attacks put a lot of load on your servers. Even the unsuccessful ones can slow down your website or completely crash the server. This is why it’s important to block them before they get to your server. To do that, you’ll need a website firewall solution. A firewall filters out bad traffic and blocks it from accessing your site.
2. **Disable PHP File Execution in Specific WordPress Folders:**

Hackers may want to install and execute a PHP script in your WordPress folders. WordPress is written mainly in PHP, which means you cannot disable that in all WordPress folders. However, there are some folders that don’t need any PHP scripts, such as your WordPress uploads folder located at /wp-content/uploads. You can safely disable PHP execution in the uploads folder, which is a common place that hackers use to hide backdoor files.

1. **Protect WordPress Admin Directory:**

Most brute force attacks on a WordPress site are trying to get access to the WordPress admin area. You can add password protection on your WordPress admin directory on a server level. This will block unauthorized access to your WordPress admin area. Simply log in to your WordPress hosting control panel (cPanel) and click on the ‘Directory Privacy’ icon under the Files section.

1. **Add Two-Factor Authentication in WordPress:**

Two-factor authentication adds an additional security layer to your WordPress login screen. Users will need their phones to generate a one-time passcode along with their login credentials to access the WordPress admin area. Adding two-factor authentication will make it harder for hackers to gain access even if they are able to crack your WordPress password.

1. **Disable root access to application that do not need it:**

We escalated access from robot to root using NMAP which was owned by root. We need to monitor all apps access and make sure that unnecessary applications which can trigger shell in interactive mode, should not have interactive mode.