# **COLD - THM**

This was an Easy THM room that was intended to refine some textbook WordPress exploitation techniques. As always, a web penetration test cannot begin without careful enumeration of a box. Being classed as ‘Easy’ is relative; easy if you have the knowledge and experience and hard otherwise. It is helpful to add the IP address of the Cold box into the /etc/hosts file  
  
-------------------------------------------------  
  
#**ENUMERATION**  
  
Visiting the IP address in the browser, takes us to the Cold Home page. Having an extension such as Wapplyzer is beneficial as it does a passive scan on websites and shows us the technologies that are in use. Apache, WordPress, some miscellaneous utilities and the programming languages used. Wapplyzer can be used to confirm some results from other scans and I find it really helpful in passively understanding a site.   
  
>Run a Port scan on the machine IP address in order to understand what services are being utilized.  
  
***rustscan -a $IP\_ADDRESS -r 0-65535 -b 900 -t 3000 -- -sC -sV --version-light -oN ColdPorts***   
|\_\_ the flags assign the following respectively, the IP address to scan, the port range (all of them) , the batch size (to slow the scan down), timeout for failed requests and the pipe to Nmap that performs a default script scan, and a light version scan and saves the output to ColdPorts file  
  
  
>While the scan is running web tools can be used to paint a better picture. Ffuf and WPScan (no surprise) are beneficial in web penetration tests.   
  
***ffuf -u*** [***http://cold.thm***](http://cold.thm) ***-w ~/SecLists/directory-list-2.3-medium.txt -c -ac -of md -o ColdHomeFuzz -e .txt,.php,.html,.js,.css,***  
|\_\_ respectively the URL to scan, the directory-listing file to be used, a colored output, autocalibration of the scan, a output format of md and the file to save results to ColdHomeFuzz while looking for files that match the extensions  
  
***wpscan --url*** [***http://cold.thm***](http://cold.thm) ***-e --api-token***   
|\_\_ wpscan will scan the url provided and by default the -e flag meaning enumeration will look for installed themes, plugins, users and other information related to the WordPress site

|\_\_ the api-token is acquired by having an account in the WP site  
  
The port scan shows that Port 80 and Port 4512 are open running HTTP and SSH respectively. The WPScan shows that there are vulnerabilities that are possible in the site. At first glance they seem to require a lot of work and hacking is time sensitive so let’s target the weakest link in a WordPress site, the Users  
  
**#ATTACK VECTORS**  
  
With our enumerated information, we have found 4 potential users on the site. Create a wpuser file [ each user on their own line] for usage in further attacks. Our attacking options are   
  
*1. Brute Forcing xmlrpc*  
|\_\_ xmlrpc is active as browsing to <http://cold.thm/xmlrpc.php> brought up the default POST request acceptance message  
  
*2. Brute Forcing SSH*  
|\_\_ we have possible users who may be able to ssh into the box  
  
*3. Brute forcing wp-login*  
|\_\_ possible with WPScan or Hydra  
  
We used Hydra in order to brute force the WP-login paging hoping that the administrator haven’t installed a plugin (that wasn’t detected) that blocks brute force attacks. (Our WPScan should that there were no plugins found so this vector should work.

***hydra -L wpusers -P rockyou.txt cold.thm http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location' -I -v***

|\_\_ -L user file to read from

|\_\_ -P : the password file

|\_\_ the protocol attacking

|\_\_ -I : initiate straight away, don’t wait for the 10 seconds

|\_\_ -v : verbose

|\_\_ the URL request accompanying the login attempt (can be viewed in the developer tools or Burpe and copied)  
  
Hydra successfully brute-forced the page.  
  
  
**#IN THE TRENCHES**  
  
Logging into the dashboard, C0ldd is an administrative user as he has the ability to change the template and source code. A good practice in order not to corrupt the current theme (unless that is the intention) is to use a deactivated theme. Selecting an unused theme and editing that themes 404.php file to create a reverse shell to out tun0 IP. Before executing the 404.php file set up a listener. Just in case there is a WAF in place, set up the PORT to 80.  
  
 ***Listener: nc -lnvp 80***

|\_\_ on port 80 as I wont be blocked by any security measures if in place.   
  
Execute the reverse shell: <http://cold.thm/wp-content/themes/twentyfifteen/404.php>  
  
  
  
**#BEHIND ENEMY LINES**  
  
The reverse shell was successful and we need to enumerate. In order to cement enumeration concepts into one’s mind, it is good to perform manual enumeration instead of relying on automated scripts. The server is running on an old version of Ubuntu and may be vulnerable to exploits. Nothing special was found out but if we remember this has a WordPress site running and therefore will contain a wp-config.php file. Changing directory into the default installation of WordPress /var/www/html provides us with credentials to a compromised user.  
  
  
**#DEEPER INTO THE TRENCHES**  
  
Checking to see if this user is carrying out password reusing gives us a SSH login where we can carry out ***sudo -l*** to see our possible escalation points from where it will be straight forward to either gain a root shell or just cat /root/root.txt flag.