**CYBER SECURITY – MINI PROJECT**

**BY: ADITYA SINGH**

**PENETRATION TESTING THROUGH COLDBOX**

1. **ABSTRACT**

The "Pentesting on ColdBox" project systematically evaluated the security of a ColdBox application through a comprehensive testing methodology, combining automated and manual techniques. Uncovering vulnerabilities like SQL injection and cross-site scripting, the project recommended specific mitigation strategies, such as implementing parameterized queries.

The findings underscored the ongoing need for robust security testing and risk management. The insights gained not only identified potential threats but also provided actionable recommendations for fortifying the ColdBox application. This proactive approach contributes to reducing the risk of successful cyber-attacks, safeguarding both the organization and its users, and offers a roadmap for future security testing endeavours.

1. **INTRODUCTION**

As organizations increasingly rely on web applications, securing digital assets becomes a paramount challenge. Web applications face diverse attacks, and attackers continually refine tactics. To address these risks, penetration testing, like the "Pentesting on ColdBox" project, simulates attacks to identify and address vulnerabilities, enhancing the overall security of web applications.

1. **BACKGROUND**

ColdBox, an open-source CFML framework, prioritizes simplicity and extensibility for web application development. Offering built-in security features like input validation, it remains popular among developers. However, no system is impervious to vulnerabilities, necessitating regular security testing to pinpoint and address potential weaknesses in ColdBox applications.

1. **OBJECTIVES**

The "Pentesting on ColdBox" project aimed to evaluate ColdBox application security by employing a comprehensive testing approach. Utilizing a blend of automated tools and manual techniques, including source code analysis, the project identified vulnerabilities like SQL injection and cross-site scripting. Recommendations for remediation were prioritized through risk analysis, targeting critical areas to reduce the risk of successful cyber-attacks on the ColdBox application. The testing methodology encompassed automated scanning, web application firewall configuration, and meticulous manual testing to ensure a thorough assessment.

1. **FINDINGS**

During the "Pentesting on ColdBox" project, vulnerabilities like SQL injection, cross-site scripting, and session fixation were identified. SQL injection allows attackers to manipulate the database, potentially accessing sensitive information. Cross-site scripting enables malicious code execution in the user's browser, and session fixation permits unauthorized session hijacking. Mitigation strategies included implementing parameterized queries to counter SQL injection and proper output encoding to prevent the reflection of untrusted input in the case of XSS.

1. **INSTRUCTIONS**

Penetration testing, or "pentesting," enhances security by identifying and exploiting vulnerabilities in computer systems. ColdBox, a popular open-source web application framework, presents a common target. The process involves reconnaissance, scanning for weaknesses, exploitation of identified vulnerabilities, post-exploitation assessment, and reporting. It's crucial to conduct pentesting with proper authorization to avoid legal consequences. Engaging professional and experienced teams is recommended for thorough testing and minimal risk to the system and its users.

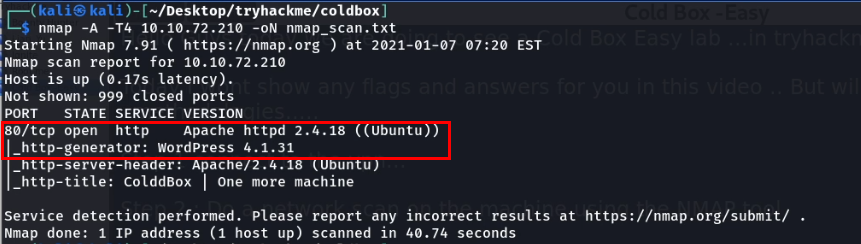
1. RECONNAISSANCE

Gather information about the target application, such as its architecture, technology stack, and any potential entry points or vulnerabilities.

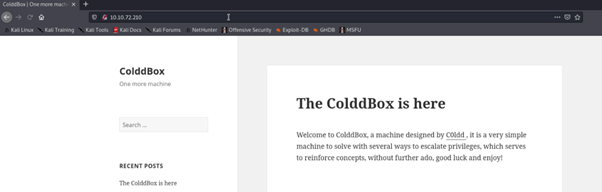
Let's start our Nmap scan using the following command.

*Nmap -A -T4 <IP Address>*

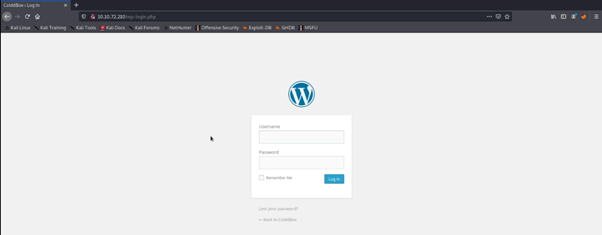
Using this command we got an output that there is only one port is opened in the TCP layer that is HTTP — 80.



After finding out the http port is opened, I further enumerated and found there is a word press blog is running in this machine.



So, I checked for the login page for the wp-admin panel and I got a login portal for the WordPress blog,

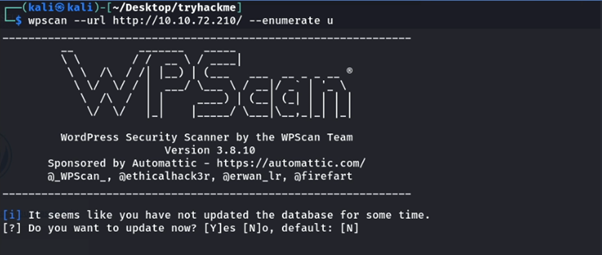


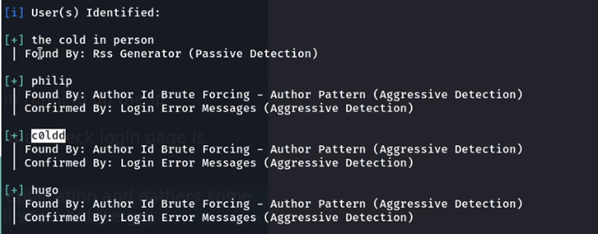
1. SCANNING

Use automated tools such as port scanners, vulnerability scanners, and web application scanners to identify any weaknesses in the system.

Next, I have an idea to enumerate the user credentials, for that we have an wonderful tool called wp-scan, the below command is used for to enumerate the users.

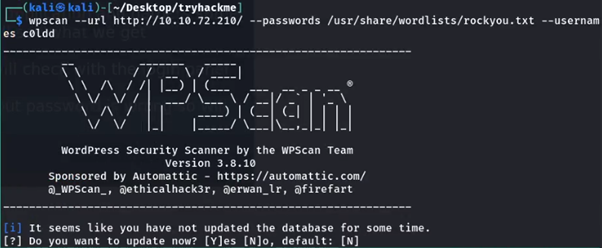
*wp-scan –url http://IP Address –enumerate u*

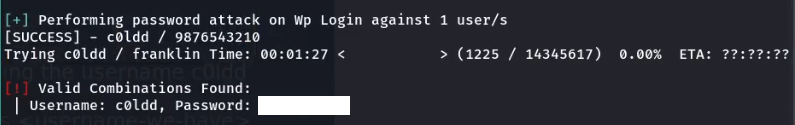
After viewing the result I confirmed that the user is c0ldd because the machine name also cold so having that as a hint I used this username in the login portal and again confirmed the user is valid with the error thrown by the portal.



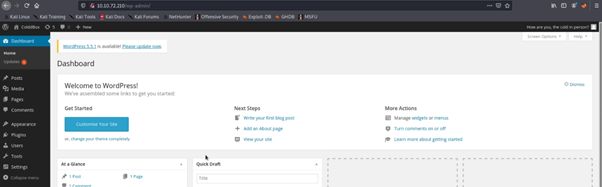
For brute forcing the password,

*wp-scan –url http://ipaddress/wp-login.php — passwords <password file > — usernames <username>*



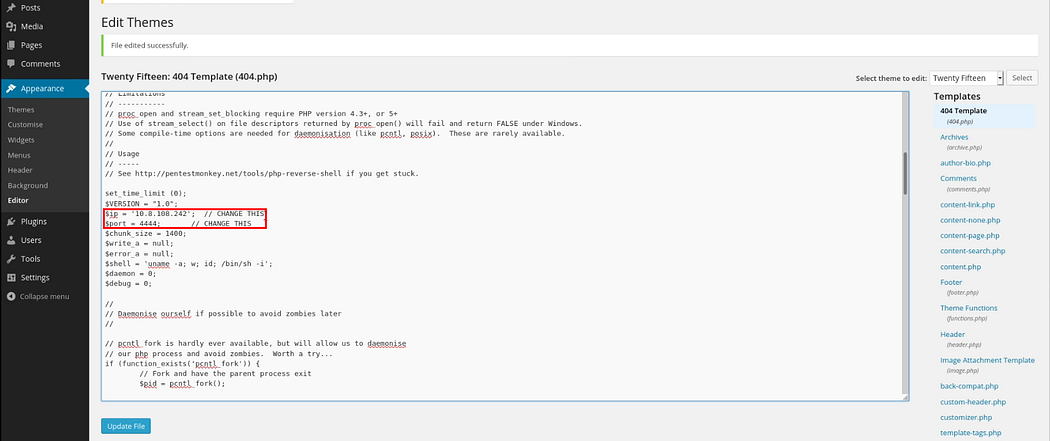


Then using this I logged into the WordPress dashboard.

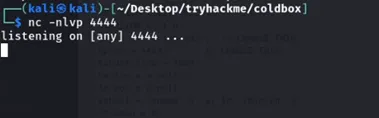


1. EXPLOITATION

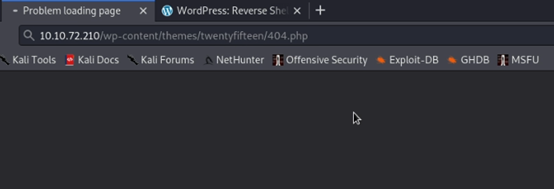
Once vulnerabilities have been identified, attempt to exploit them in order to gain access to the system and its data. This may involve using techniques such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

And in WordPress pentesting we knew that if we can able to change the theme editor code then we can able to get the reverse shell easily so I also did the same just edited the 404.php page with my php reverse shellcode and updated the 404.php page.

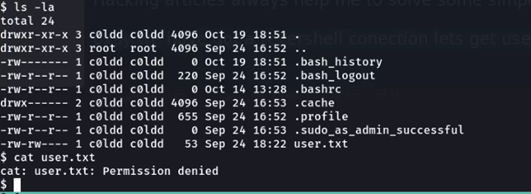
Then after the success update, I started my net cat listener.



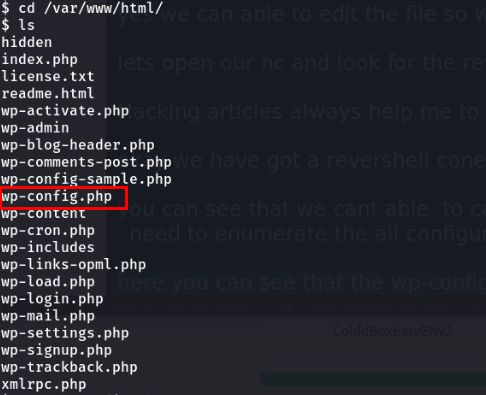
Then I accessed the page in the browser and I got a reverse connection from the webserver.

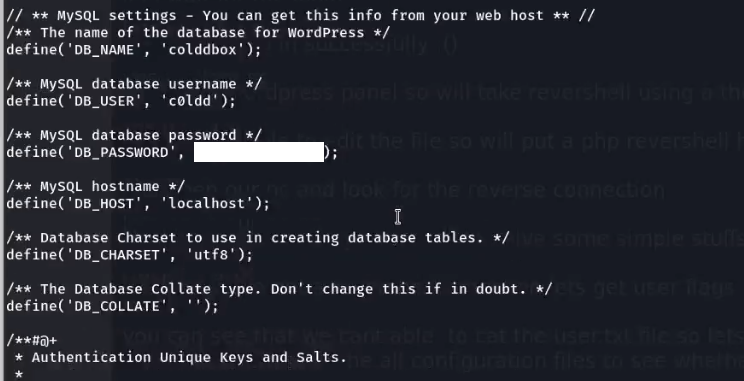


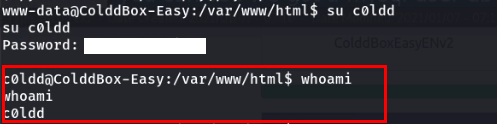
After this step I thought I easily get the user flag but unfortunately, I can able to get that because the machine saying that permission is denied for the www-data user.

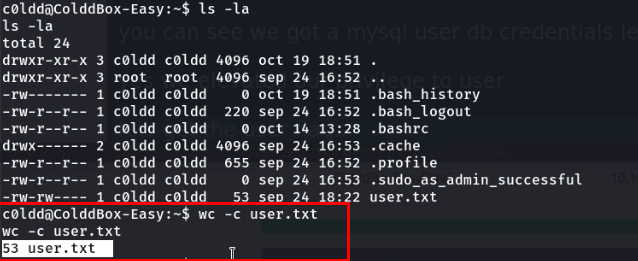


So our next step is to enumerate the internal configuration files and need to get the credentials for any user, so I finalized that this is the WordPress application so it has a wp-config .php file that contains some basic information and sometimes it has hardcoded the password with the file. So I enumerated the file and found a password for the user c0ldd.



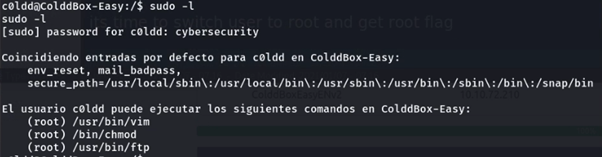
I found the DB credentials so I tried these credentials to switch to the user c0ldd using su command and I successfully logged in as a c0ldd user.

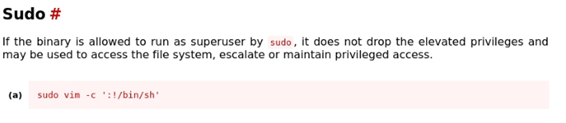


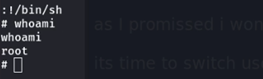
Now we need to get the user flag, but I will not show the user flag here. But you can view the user flag now without any issues.

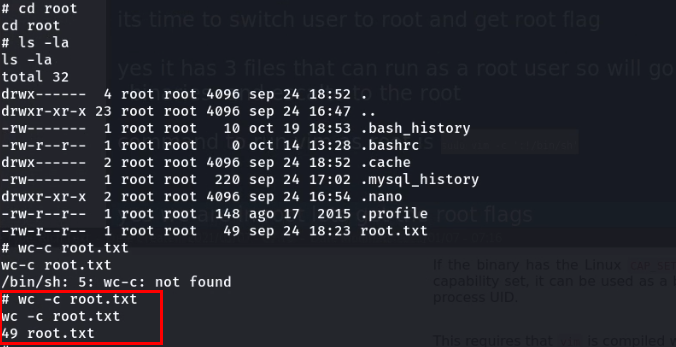
1. POST-EXPLOITATION

Now we need to escalate our privilege to root to get the root user flag.

Great we have found that vim is able to run with root permissions, so will go for our favourite website called GTFOBIN and form there I found a command to run vim as sudo root.

By this command, I escalated as root and got a root flag.





1. **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.