

# Faculty Writeup

---

Felicitas Pojtinger (fp036)

2022-10-12



Hack The Box  
PEN-TESTING LABS

Figure 1: Hack the Box Logo

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

## Disclaimer

---

# 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](mailto:fp036@hdm-stuttgart.de)

## Preface and Personal Statement

---

## 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 difficulty.

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

## Skills Required

---

## 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.



## 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 arbitrary 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

## Information Gathering

---

## 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
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
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`:

## Exploitation

---

# 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.

The screenshot shows a web browser window with the URL `http://10.10.10.10/faculty/`. The page displays the 'School Faculty Scheduling System' interface. The 'Subject List' section is active, showing a table of subjects. The table has columns for '#', 'Subject', and 'Action'. The subjects listed are:

#	Subject	Action
1	Subject: DBMS Description: Database Management System	<a href="#">View</a> <a href="#">Delete</a>
2	Subject: Mathematics Description: Mathematics	<a href="#">View</a> <a href="#">Delete</a>
3	Subject: English Description: English	<a href="#">View</a> <a href="#">Delete</a>

The browser's developer tools are open, showing the 'Network' tab. The 'Request' for `http://10.10.10.10/faculty/` is selected, and the 'Response' pane shows the HTML content of the page. The response includes a meta-tag with the title 'School Faculty Scheduling System' and a link to the 'School Faculty Scheduling System'.

## User Flag

---

## 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%
```



## Root Flag

---

## 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		