**Grandpa**

Difficulty: Easy

OS: Windows

**Nmap**

Performing an aggressive nmap scan shows only port 80 is open. We see under this some extra information the nmap scan picked up for us. The useful information here is Microsoft IIS Version 6.0. Doing a quick google search reveals this version of IIS released back with Windows Server 2003. Given the age of this IIS release, this could be a potential foothold.

|  |
| --- |

**Enumeration**

Going to the website, we are presented with a basic default page for Windows IIS. The first action I take when encountered with a web page is performing a FUZZ scan. Doing this enumerates a couple web directories that may be useful.

|  |
| --- |

Searching for vti\_bin admin.dll and author.dll exploits, we see there are potential vulnerabilities. However, I am going to focus on the fact that this web server is IIS 6.0, an extremely out of date release and thus the potential for more well known vulnerabilities and exploits.

Using searchsploit, we see there are a few vulnerabilities for IIS 6.0 that could be utilized.

| *Searchsploit iis 6.0* |
| --- |

Analyzing the results from searchsploit, a few vulnerabilities can be ruled out. Specifically, denial of service attacks, the internal IP address, and the PHP one at the bottom. The most attractive metasploit modules here are the WebDAV ones. This is due to the fact that our nmap scan from earlier reported some enumeration scripts on WebDAV, meaning that is where a vulnerability most likely resides. Based on this, I am going to use the “Remote Buffer Overflow” metasploit module for IIS 6.0.

**Metasploit**

Using the metasploit module “windows/iis/iis\_webdav\_scstoragepathfromurl”, we first set the LHOST and RHOST along with any other options that need to be set. In my case, only the aforementioned settings must be altered. Once those two are set, running the module sends us back a meterpreter shell.

|  |
| --- |

tun0 can be used here instead of our IP. Metasploit will automatically fetch our tun0 address and replace it in its script.

|  |
| --- |

The shell is very unstable and only lasts a few seconds. I attempted to use powershell, but I could not find its proper directory. Some googling would have done, but I found we can migrate the metasploit meterpreter shell to another more stable one with the following:

| *ps*  *migrate 3840* |
| --- |

Note that a different migration may be needed for everyone.

After obtaining a relatively stable meterpreter shell, I checked systeminfo and found the server was running on an outdated version of windows with no hotfixes.

|  |
| --- |

Given the information from systeminfo, we know the next best step is to run a vulnerability scanner. Since we are performing the box on metasploit, we are going to use metasploit’s vulnerability scanner. To access this, we need to place the meterpreter session into the background. This can be achieved with “Ctrl + Z” which will place the windows shell in the background. After this, the meterpreter itself can be backgrounded with the “background” keyword. Upon performing these steps, we find ourselves back in metasploit where we can use metasploit’s “local exploit suggester.” The following screenshot shows how to do all of this.

| *use post/multi/recon/local\_exploit\_suggester* |
| --- |

Now that we have the exploit suggester, we need to set the session to scan. We know what session our meterpreter shell is running either from the information given to us upon placing the meterpreter into the background, or with a simple “sessions -i” query. Once we know the session id, we can set that option in the exploit suggester and run the script.

|  |
| --- |

Running the exploit suggester, we see the following vulnerabilities.

|  |
| --- |

Looking at the results, we can take one of the suggested vulnerabilities and run it against the session metasploit created earlier. We will use “ms14-070\_tcpip\_ioctl”.

Looking at the “ms14-070” metasploit module, we see the only options that need to be set are LHOST and session number. Once again setting these to our local IP address and whatever session our meterpreter is running on will suffice. Following this, running the script will perform the exploit and return to us a shell as the administrator user.

| *use exploit/windows/local/ms14\_070\_tcpip* |
| --- |

Rooted