

WHITEPAPER_

Hidden Network: Detectando redes ocultas con los dispositivos USB

12.07.2017

Índice

1. Los riesgos de las conexiones.....	4
1.1. Aislamiento de redes y conexión con un USB	5
2. Conexiones de dispositivos USB en un equipo.....	6
2.1. Hidden Links: Detección de este tipo de redes.....	7
2.2. USB Hidden Networks for WinRM.....	8
2.2.1. Script: LaunchUSBHiddenNetworks.....	9
2.2.2. Script: RecollectUSBData	10
2.3. USB Hidden Networks for SMB con PSExec.....	14
2.3.1. Script: LaunchUSBHiddenNetworks.....	14
2.3.2. Script: RecollectUSBData	16
2.4. Información histórica.....	16
2.5. Hidden Links en OS X.....	18
2.6. Mitigación	19
3. Conclusiones.....	19
Más información.....	20

Resumen ejecutivo

Muchas de las empresas y organismos gubernamentales de hoy en día tienen redes aisladas de las comunicaciones o con el flujo de datos limitado a través de diferentes redes.

Estas redes de equipos se crean para situaciones especiales, ya que pueden ser redes muy especiales o con información crítica como, por ejemplo, sistemas de control en fábricas, entornos de alta seguridad en el procesado de cierta información o redes que cumplen con un estándar de seguridad. En la historia reciente de la Ciberseguridad se ha comprobado como un software malicioso denominado Stuxnet se infiltraba en una red aislada en una Central Nuclear.

Con este hecho se puede observar como no es suficiente tener una red de equipos no conectados con cable Ethernet o WiFi a otras redes. Cualquier tipo de conexión exterior a un equipo puede materializar una amenaza. Este trabajo refleja las posibilidades que ofrecen las denominadas Hidden Network y cómo se pueden localizar y enfocar a la protección de estos puntos en una red corporativa.

Chema Alonso – chema@11paths.com

Pablo González – pablo@11paths.com

Francisco Ramírez – franciscojose.ramirezvicente@telefonica.com

1. Los riesgos de las conexiones

En el ámbito de la seguridad de las redes de datos hay una tendencia a dibujar las redes por las conexiones a nivel de enlace, es decir, conexiones Ethernet, WiFi, etcétera. Las redes corporativas son mucho más complejas que esto y hay que realizar un estudio desde diferentes puntos de vista.

El análisis de tráfico en una red de datos corporativa es la mayor herramienta que se dispone para entender lo que sucede en ésta. Además, es la opción más coherente, realizando la búsqueda de los nodos de la red con los protocolos más utilizados, cuáles son los nodos con los servicios más críticos o cuáles se comportan como cuellos de botella. De estas afirmaciones se puede extrapolar que un análisis de red basado en los riesgos dispone de una gran cantidad de datos e información basada en distintas directrices.

Esquematizar una red de forma estética es correcto para entender los nodos y la configuración de la red, pero su análisis de telemetría es algo fundamental para poder contener y prepararse para los riesgos. Con estos datos se obtiene un mayor nivel de entendimiento sobre la red y las amenazas que ésta tiene.

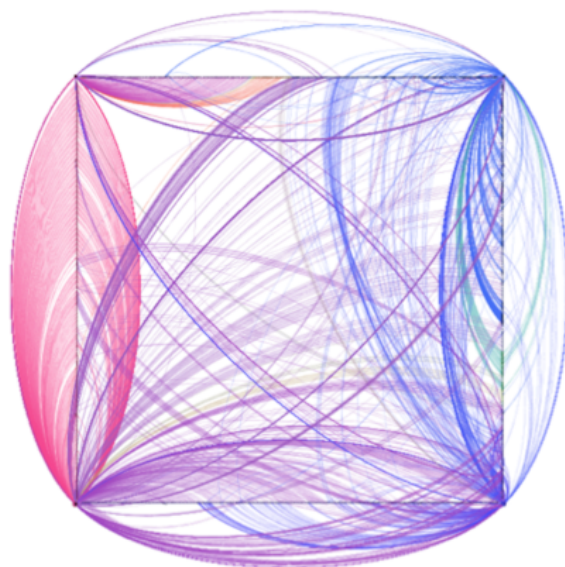


Figura 1: Simulación de un mapa de red con diferentes nodos y conexiones entre ellos

La representación de los nodos y las estructuras de conexión es fundamental para poder entender las diferentes fronteras que se tienen dentro de la red para poder mitigar intrusiones, detectar ataques o llevar a cabo la aplicación de medidas de seguridad de forma preventiva.

El problema viene dado ante la situación de entender qué es una red. En muchas ocasiones, se define una red como un grupo de máquinas unidas a través de la posibilidad de comunicar entre ellas a través de diferentes tecnologías y protocolos. La mayoría de las ocasiones, los usuarios o administradores de sistemas y redes dan por bueno que una red será conexión Ethernet o WiFi de las diferentes máquinas de la organización. Esto no sucede así, ya que una organización que no aplique medidas de prevención en el uso de dispositivos USB, puede disponer de lo que se denominan Hidden Networks o redes ocultas. Estas redes se crean a través del uso de dispositivos USB y permite comunicar máquinas que físicamente o lógicamente estén aisladas.

1.1. Aislamiento de redes y conexión con un USB

Para entender el riesgo de las Hidden Networks que se crean a través de los dispositivos USB se muestra un sencillo ejemplo. Suponiendo que una organización dispone de una red en la que existen 3 tipos de VLANs. En la primera VLAN se encuentra:

- Un equipo denominado A. Este equipo tiene conectividad con los equipos de su misma VLAN.
- Un equipo denominado B. Este equipo tiene conectividad con los equipos de su misma VLAN.

En la segunda VLAN se encuentra:

- Un equipo denominado C. Este equipo tiene conectividad con los equipos de su misma VLAN.
- Un equipo denominado D. Este equipo tiene conectividad con los equipos de su misma VLAN.
- Un equipo denominado E. Este equipo tiene conectividad con los equipos de su misma VLAN.

En la tercera VLAN se encuentra:

- Un equipo denominado F. Este equipo tiene conectividad con los equipos de su misma VLAN.

Si se observa el esquema de redes de la imagen, se puede visualizar como los equipos se encuentran aislados por las diferentes VLANs. Suponiendo que los empleados de dicha organización intercambian datos a través de dispositivos USB, existe un alto grado de posibilidad de que la información pase de un equipo de una VLAN a otra. Si se añade que el dispositivo USB es una fuente de amenazas, se está generando una red oculta dentro de la propia organización.

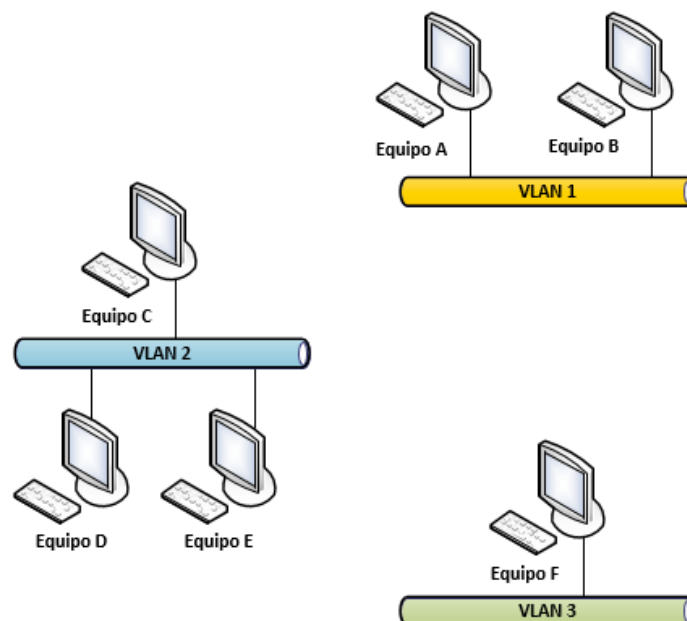


Figura 2: Esquema de red con equipos conectados a diferentes VLANs

Suponiendo que los usuarios de la máquina F y E se intercambian información a través de un dispositivo USB, se está creando una red oculta entre ambas máquinas. Se puede representar como dos nodos E y F con un arco dirigido entre el equipo que primero introdujo el dispositivo USB y el segundo equipo.

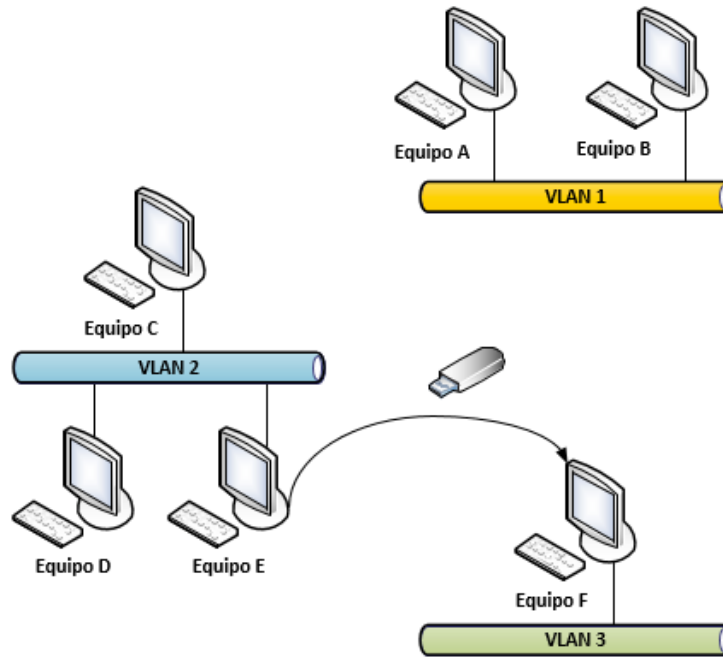


Figura 3: Superposición de red oculta en el esquema de red anterior

2. Conexiones de dispositivos USB en un equipo

Cuando un dispositivo USB se conecta de un equipo a otro surge el término “Polinización”. Este es un término similar al utilizado en otros ámbitos y consiste en llevar la amenaza o riesgo de un dispositivo USB entre diferentes equipos, aunque estén en diferentes redes.

Cuando un usuario conecta un dispositivo USB se crean en el sistema una serie de entradas en el registro de Windows. Este tipo de información es valiosa, por ejemplo, en un análisis forense, con el objetivo de conocer información de por dónde se produce la fuga de información o por dónde entró la amenaza en un Post-Mortem.

La clave USBStor que se ha creado en el registro del sistema Windows almacena los diferentes dispositivos que se han insertado en el equipo. Si se han insertado en una máquina N dispositivos USB, se encontrarán esos N dispositivos en la clave USBStor con toda la información para identificar al dispositivo.

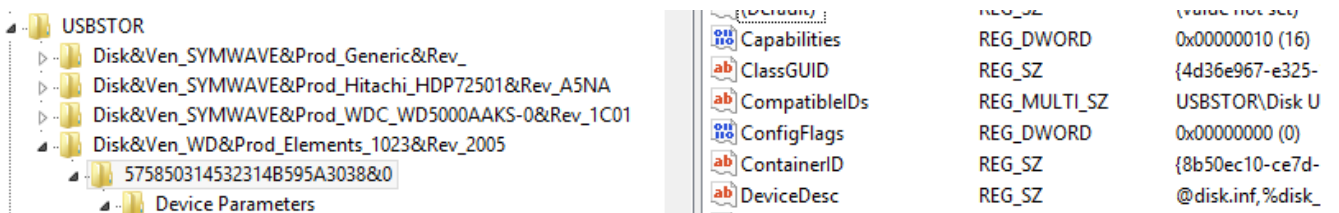


Figura 4: Visualización en el registro de los dispositivos USB introducidos

La información que se puede recopilar de los dispositivos USB conectados a un equipo es:

- Nombre del dispositivo.
- Class.

- ClassGUID.
- HardwareID.
- Servicio que proporciona el dispositivo, por ejemplo, si es un disco duro.
- Driver.
- Etcétera.

2.1. Hidden Links: Detección de este tipo de redes

Conociendo dónde y cómo se almacena la información de un dispositivo USB en un sistema operativo Microsoft se podría conocer quién está compartiendo el dispositivo USB con quién. De esta forma se puede generar dos nodos que representan dos equipos y un arco que identifica la conexión entre ambos equipos. Se está descubriendo una red oculta gracias al Hidden Link. Además, se puede detectar en qué equipo se conectó a priori, debido a los diferentes eventos que se pueden obtener de un sistema operativo. De esta forma el arco entre los nodos es dirigido.

Para llevar a cabo una automatización de detección de Hidden Links se ha propuesto el siguiente diseño:

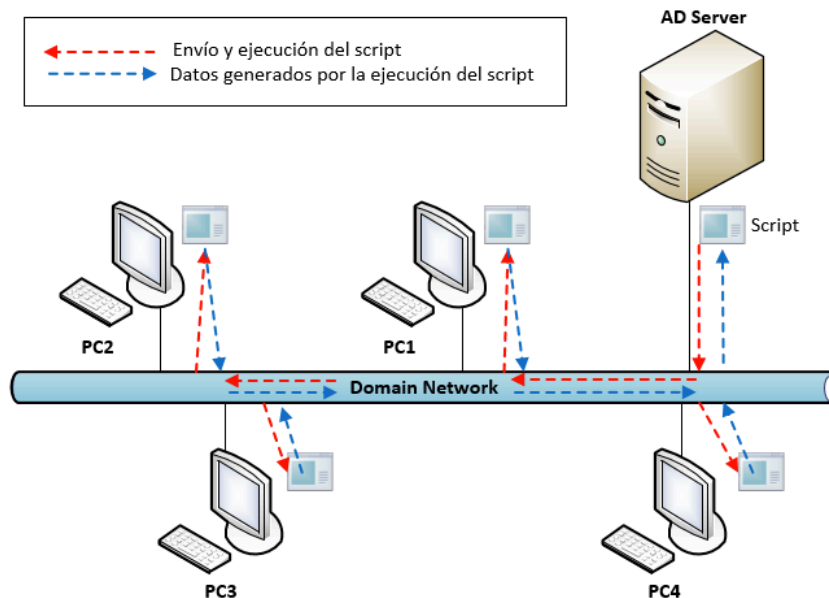


Figura 5: Esquema del lanzamiento del script en un AD (Active Directory)

En la imagen, se puede visualizar cómo la aplicación es ejecutada en un nodo central y puede utilizar diferentes tecnologías Microsoft para ejecutar las instrucciones en todas las máquinas del dominio. Las tecnologías estudiadas, las cuales encajan con el diseño de la solución son las siguientes:

- WinRM.
- SMB (Server Message Block).
- WMI.

Powershell es la línea de comandos de Microsoft orientada a objetos con la que se consigue una interacción sencilla y potente con cualquier estructura del sistema operativo de Microsoft.

```
PS C:\> Get-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Enum\USBSTOR\* |
Select-Object -ExpandProperty FriendlyName
-----
SanDisk U3 Cruzer Micro USB Device
WD Virtual CD 1110 USB Device
USB DISK 2.0 USB Device
USB DISK 2.0 USB Device
ADATA USB Flash Drive USB Device
Corsair Voyager USB Device
FLASH Drive AU_USB20 USB Device
hp USB Flash Drive USB Device
Kingston DT 101 G2 USB Device
Kingston DT 101 G2 USB Device
SanDisk U3 Cruzer Micro USB Device
USB Flash Disk USB Device
WD 3200BEV External USB Device
WD My Book 1110 USB Device
WDC WD25 00JB-00GVA0 USB Device
```

Figura 6: Obtención de dispositivos USB conectados con Powershell

2.2.USB Hidden Networks for WinRM

La versión WinRM del script en PowerShell requiere tener activado el servicio de Administración Remota de Windows (WinRM) en todos los equipos de la red que se va a auditar:

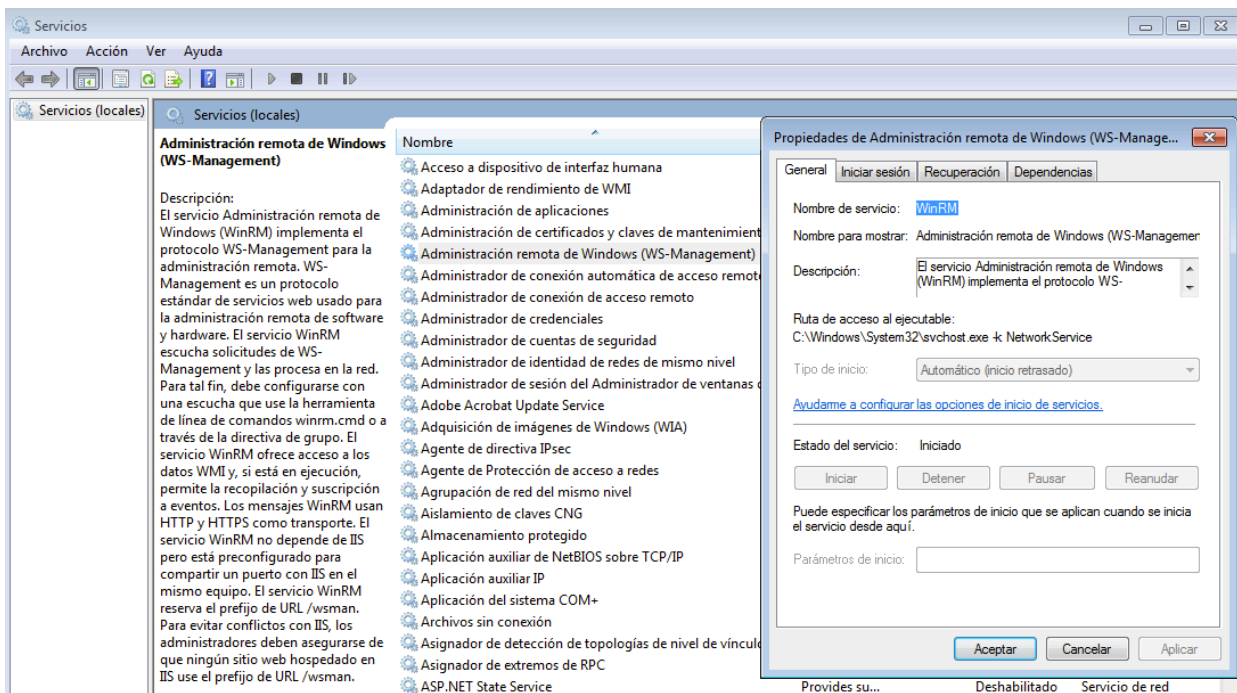


Figura 7: Servicio WinRM

Por otro lado, el script se ha probado en una red con un Dominio y un Directorio Activo o AD para automatizar lo máximo posible la recolección de información. Se utilizan las credenciales de administrador de Dominio para autorizar la ejecución en los equipos remotos de la red local. Al ejecutar el script se solicitarán las credenciales.

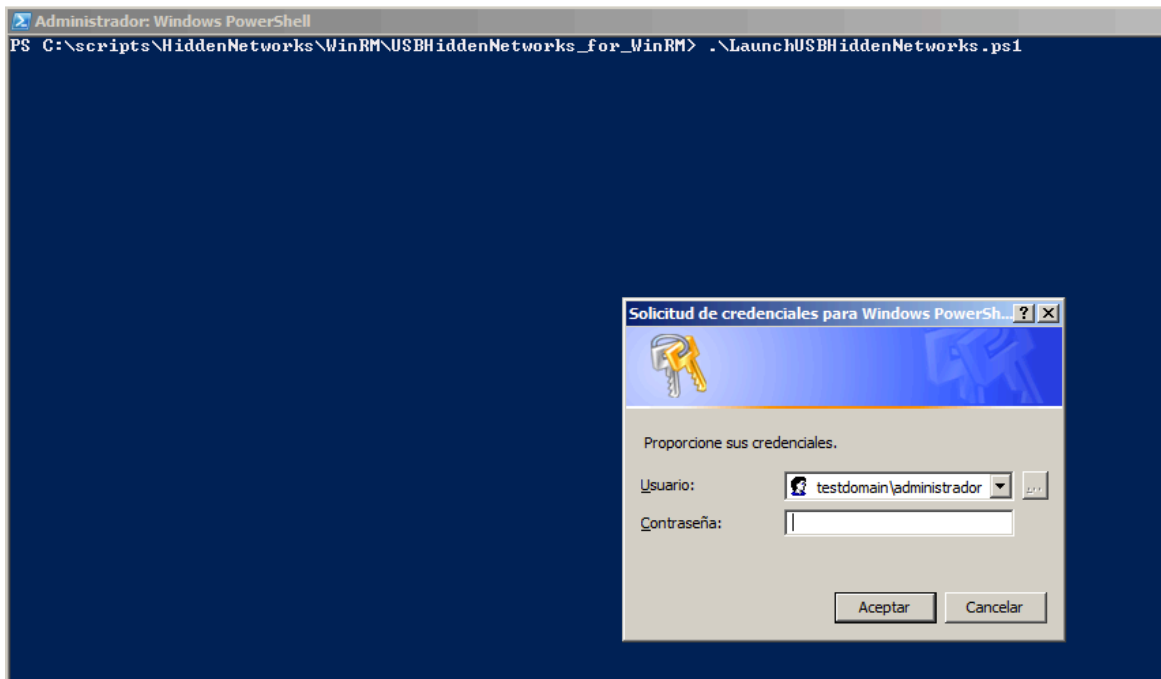


Figura 8: Solicitud de credenciales en el uso del script

La ejecución principal del script se realiza a través del programa “LaunchUSBHiddenNetworks.ps1” el cual conectará con los equipos remotos pasando como parámetro el script a ejecutar llamado “RecollectUSB.ps1” el cual se encarga de recopilar la información de los dispositivos USB. Por lo tanto, el script se ejecutará de forma individual en cada uno de los equipos asignados.

2.2.1.Script: LaunchUSBHiddenNetworks

La ejecución de este comando se basa en el comando “Invoke-Command” de PowerShell. Dicho comando permite conectar con un equipo de la red pasando como parámetro el FQDN, nombre de equipo o la dirección IP y por otro lado el script de PowerShell que se quiere ejecutar:

```
$salida=invoke-command -ComputerName (Get-Content servers.txt) -FilePath 'PathToScript\RecollectUSBData.ps1'-Credential testdomain\administrador
```

Con el parámetro –ComputerName se asigna el nombre de equipo o equipos que se quieren auditar dentro de nuestro AD. Es posible introducir directamente el nombre de los equipos seguidos por comas, pero en este caso se ha utilizado un listado de equipos en un fichero TXT, servers.txt, y pasarlo como parámetro.

El parámetro -FilePath asigna la ruta donde se encuentra el script en PowerShell que ejecutará la recolección de datos. Finalmente, el parámetro –Credential permite utilizar las credenciales del administrador de dominio para autorizar la ejecución en el equipo remoto, en este caso dominio “testdomain” y usuario “administrador”.

El resultado de la ejecución se almacena en el objeto \$salida. La información recuperada se almacenará a su vez en un fichero CSV denominado “USBDATA.csv” de la siguiente manera:

```
$salida | Out-File USBDATA.csv
```

El formato del fichero CSV tiene la siguiente estructura después de una ejecución del script:

Nombre de Equipo, IP (en formato IPv4), Nombre del USB, ID (identificador único)

```

USBDATA.csv: Bloc de notas
Archivo Edición Formato Ver Ayuda
PC001,192.168.1.16,Kingston DataTraveler G3 USB Device,{2057d6e6-7725-52d5-8d5e-3fdab3357470}
PC001,192.168.1.16,SanDisk Cruzer Blade USB Device,{1df90487-d45c-5a58-8509-dff4fae7bca6}
PC002,192.168.1.15,Kingston DataTraveler G3 USB Device,{2057d6e6-7725-52d5-8d5e-3fdab3357470}
PC002,192.168.1.15,SanDisk Cruzer Blade USB Device,{1df90487-d45c-5a58-8509-dff4fae7bca6}
    
```

Figura 9: Resultados obtenidos en formato CSV

Con esta información se podría generar un grafo como el mostrado a continuación creado con la aplicación Gephi:

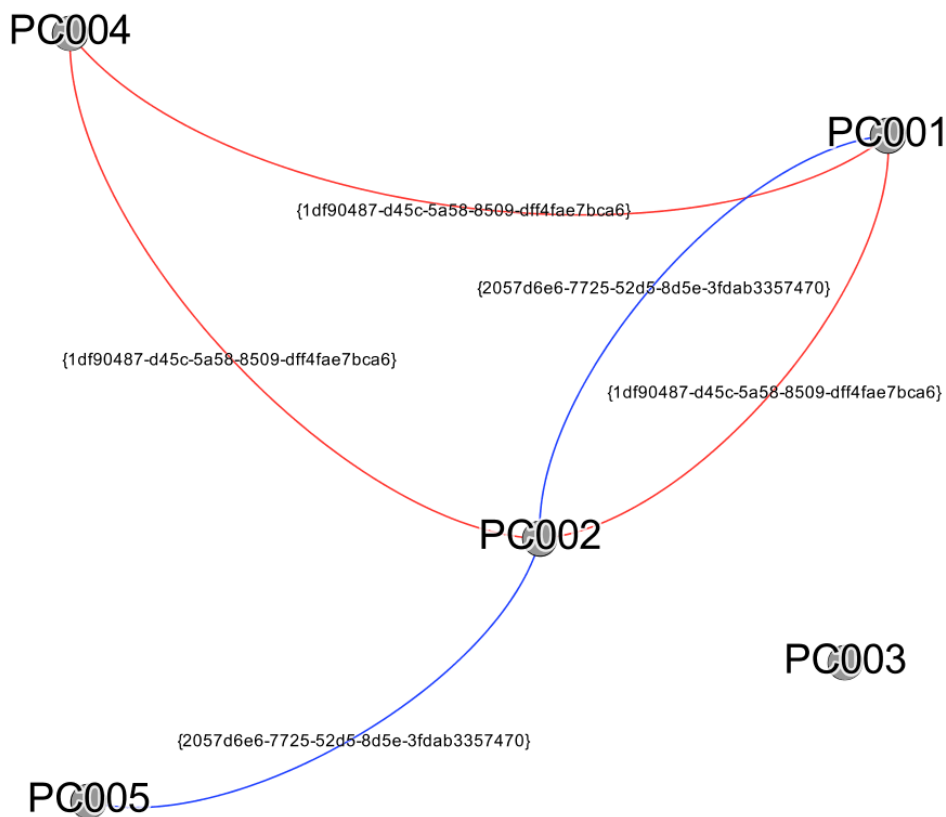


Figura 10: Grafo que representa las conexiones ocultas de los dispositivos USB en una red

2.2.2.Script: RecollectUSBData

Este script es el encargado de recopilar toda la información referente a los dispositivos USB conectados al equipo y se ejecuta localmente en los ordenadores a auditar. Los datos se recuperan desde una rama específica del registro de Windows.

```

$USBDevices = @()
$USBContainerID = @()
$USBComputerName = @()
$USBComputerIP = @()
$SubKeys2 = @()
    
```

```
$USBSTORSubKeys1 = @()
```

Se inicializan las matrices donde se van a almacenar la información relativa al ordenador auditado y los datos referentes a los dispositivos USB que tenga almacenado en el registro o se conectaron en algún instante de tiempo al equipo.

```
$Hive = "LocalMachine"
```

```
$Key = "SYSTEM\CurrentControlSet\Enum\USBSTOR"
```

En las variables \$Hive y \$Key se almacena la ruta completa de la rama del registro donde se va a realizar la búsqueda de los datos referentes al dispositivo USB. La variable \$Hive con el valor "LocalMachine" equivale a HKLM o HKEY_LOCAL_MACHINE.

```
$ComputerName = $Env:COMPUTERNAME
```

```
$ComputerIP = $localIpAddress=((ipconfig | findstr [0-9].\.)[0]).Split()[-1]
```

Se almacena el nombre del equipo local, así como su dirección IP en las variables \$ComputerName y \$ComputerIP.

```
$Reg = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey($Hive,$Computer)
```

```
$USBSTORKey = $Reg.OpenSubKey($Key)
```

```
$nop=$false
```

En el objeto \$Reg se ejecuta la consulta al registro utilizando el comando OpenRemoteBaseKey pasando como parámetro las variables \$Hive y \$Computer, las cuales definen la rama que se quiere consultar. La variable \$nop se utilizará más adelante para controlar el flujo de ejecución.

```
Try {
```

```
    $USBSTORSubKeys1 = $USBSTORKey.GetSubKeyNames()
```

```
}
```

```
Catch
```

```
{
```

```
    Write-Host "Computer: ",$ComputerName -foregroundcolor "white" -backgroundcolor "red"
```

```
    Write-Host "No USB data found"
```

```
    $nop=$true
```

```
}
```

El bloque Try – Cath se encarga de gestionar el error en caso de no encontrar ninguna información sobre algún dispositivo USB. En caso de no encontrar ninguna información, se asigna el valor \$true a la variable \$nop para evitar que se ejecute todo el proceso de identificación y recuperación de datos del dispositivo USB.

```
if(-Not $nop)
```

En caso de existir alguna entrada relacionada con la conexión de un dispositivo USB, variable \$nop a \$true, se ejecutarán los siguientes bloques:

Bloque 1:

```
ForEach($SubKey1 in $USBSTORSubKeys1)
{
    $Key2 = "SYSTEM\CurrentControlSet\Enum\USBSTOR\$SubKey1"
    $RegSubKey2 = $Reg.OpenSubKey($Key2)
    $SubkeyName2 = $RegSubKey2.GetSubKeyNames()
    $Subkeys2 += "$Key2\$SubKeyName2"
    $RegSubKey2.Close()
}
```

Cada elemento existente en la rama del registro donde se realiza la búsqueda es un dispositivo USB distinto. Cada elemento se almacena en la matriz @Subkeys2.

Bloque 2:

```
ForEach($Subkey2 in $Subkeys2)
{
    $USBKey = $Reg.OpenSubKey($Subkey2)
    $USBDevice = $USBKey.GetValue('FriendlyName')
    $USBContainerID = $USBKey.GetValue('ContainerID')
    If($USBDevice)
    {
        $USBDevices += New-Object -TypeName PSObject -Property @{
            USBDevice = $USBDevice
            USBContainerID = $USBContainerID
            USBComputerName= $ComputerName
            ComputerIP = $ComputerIP
        }
    }
}
$USBKey.Close()
}
```

Este bloque recorre todos los dispositivos USB que se han identificado previamente en el Bloque 1 y que están almacenados en la matriz @Subkeys2. Para cada elemento que tenga algún valor en el campo \$USBDevice, se procede

a recuperar el ID del dispositivo USB, USBContainerID. También se asigna el nombre del equipo y la dirección IP del mismo para añadirlo posteriormente al fichero CSV de salida.

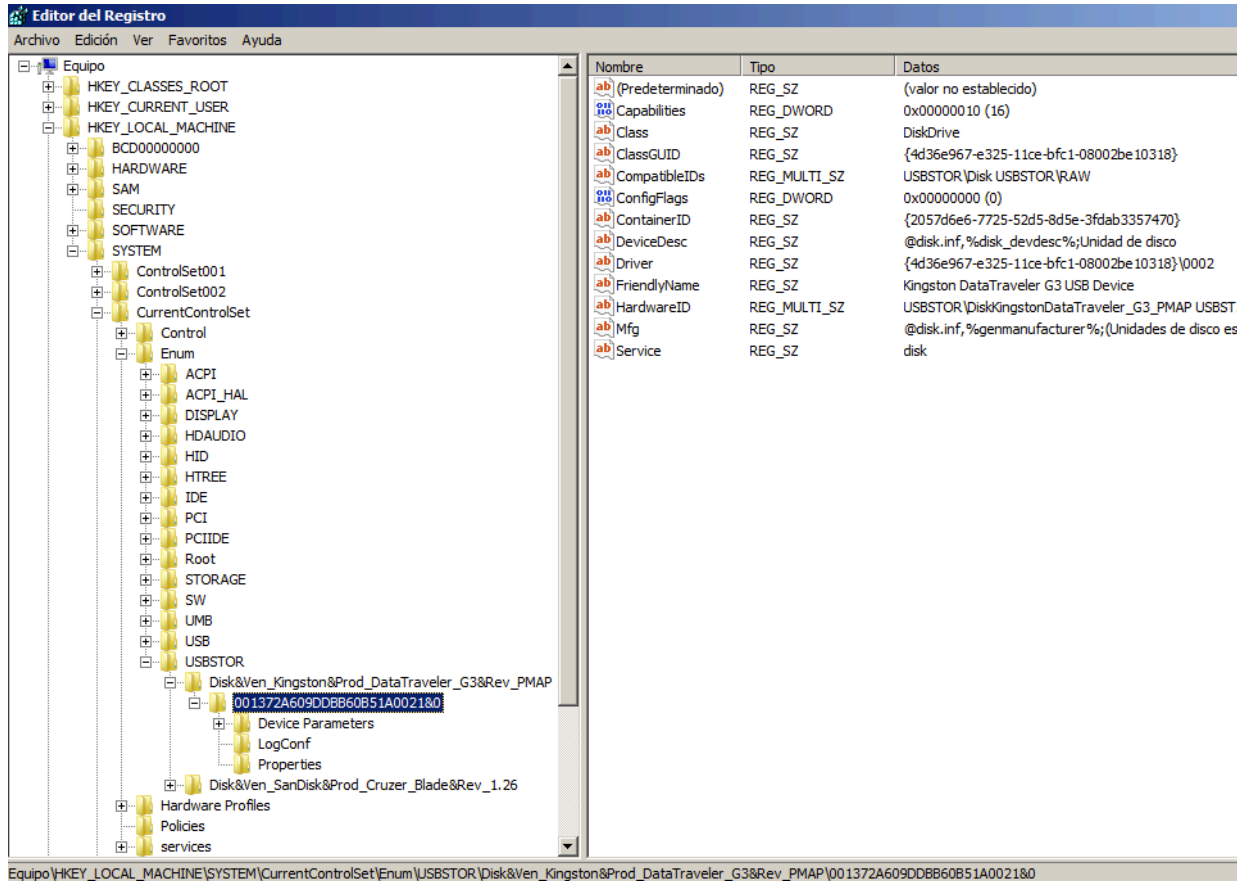
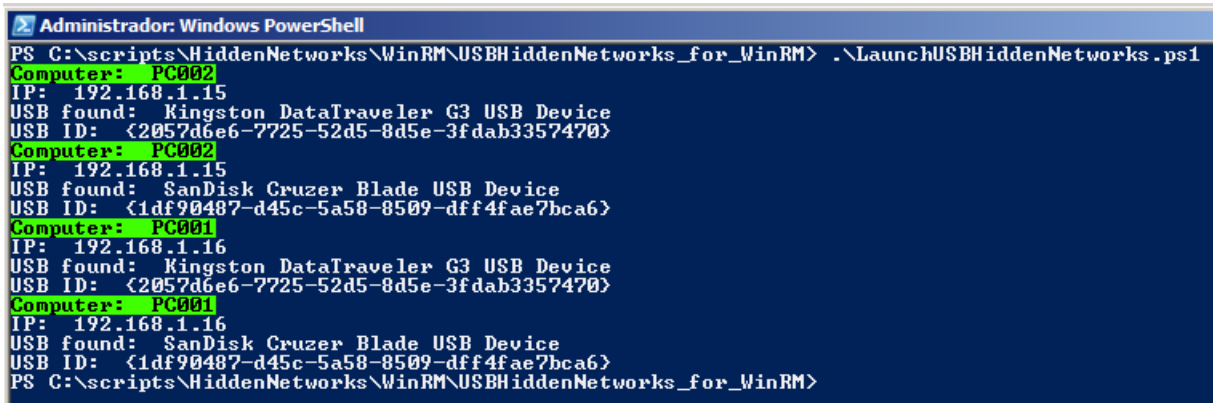


Figura 11: Rama del registro dónde se encuentran los dispositivos USB

Bloque 3:

```
for ($i=0; $i -lt $USBDevices.length; $i++) {
    $IDUnico=$USBDevices[$i] | Select -ExpandProperty "USBContainerID"
    $USBNombre=$USBDevices[$i] | Select -ExpandProperty "USBDevice"
    Write-Host "Computer: ",$ComputerName -foregroundcolor "black" -backgroundcolor
"green"
    Write-Host "IP: ",$ComputerIP
    Write-Host "USB found: ",$USBNombre
    Write-Host "USB ID: ",$IDUnico
    Echo "$ComputerName,$ComputerIP,$USBNombre,$IDUnico"
}
```

Finalmente, este bloque muestra por pantalla la información relevante obtenida del equipo remoto. Se utiliza el comando Write-Host imprimir en la pantalla del servidor donde se ha ejecutado el script. El comando Echo se utiliza como salida de los datos para escribirlos, posteriormente, al fichero CSV.



```

Administrador: Windows PowerShell
PS C:\scripts\HiddenNetworks\WinRM\USBHiddenNetworks_for_WinRM> .\LaunchUSBHiddenNetworks.ps1
Computer: PC002
IP: 192.168.1.15
USB found: Kingston DataTraveler G3 USB Device
USB ID: {2057d6e6-7725-52d5-8d5e-3fdab3357470}
Computer: PC002
IP: 192.168.1.15
USB found: SanDisk Cruzer Blade USB Device
USB ID: {1df90487-d45c-5a58-8509-dff4fae7bca6}
Computer: PC001
IP: 192.168.1.16
USB found: Kingston DataTraveler G3 USB Device
USB ID: {2057d6e6-7725-52d5-8d5e-3fdab3357470}
Computer: PC001
IP: 192.168.1.16
USB found: SanDisk Cruzer Blade USB Device
USB ID: {1df90487-d45c-5a58-8509-dff4fae7bca6}
PS C:\scripts\HiddenNetworks\WinRM\USBHiddenNetworks_for_WinRM>
  
```

Figura 12: Salida tras la ejecución del script

2.3.USB Hidden Networks for SMB con PSExec

Para la ejecución del script utilizando SMB será necesario tener instaladas previamente las utilidades PSTools, en concreto para poder ejecutar el comando PSExec, en los equipos que se van a verificar. La filosofía de funcionamiento será prácticamente idéntica a la versión WinRM. Desde el servidor se conectará con el equipo remoto y se lanzará el script desde el servidor con la cuenta de administrador de dominio, y, posteriormente, ejecutará el script de recolección de datos USB.

El script principal LaunchUSBHiddenNetworks.ps1 tendrá algunas modificaciones para adaptarse a este nuevo tipo de conexión. La principal será que esta vez no se utiliza el comando Invoke-Command para ejecutar remotamente el script. Esta vez se abrirá una shell de Powershell y se ejecutará el script desde ella. El script se descargará desde una ubicación de red, preferiblemente un servidor web que permita descargarlo mediante el protocolo HTTP. De esa forma se evitan problemas con la política de ejecución y permisos que se puedan encontrar al acceder a un recurso local compartido.

De forma similar a la anterior versión con WinRM, los resultados se irán almacenando en un fichero CSV. Para evitar problemas de sincronización y dar tiempo a que el programa se ejecute en el equipo destino, se han incluido algunos retardos como se verá a continuación en el análisis del código.

2.3.1.Script: LaunchUSBHiddenNetworks

```

$computers = gc
"C:\scripts\HiddenNetworks\PSExec\USBHiddenNetworks_for_SMB\servers.txt"

$url = "http://192.168.1.14/test/RecollectUSBData.ps1"

$sincro = 40
  
```

Se asignan varias variables. La matriz donde se irán almacenando los nombres o direcciones IP de los servidores, \$computers, que están en el fichero servers.txt, la variable \$url que indica dónde se encuentra el script RecollectUSBData.ps1 y, finalmente, el tiempo de espera para sincronizar la operación. Hay que tener en cuenta que este número puede variar en función del entorno donde se ejecute el script. Este sería un ejemplo de ejecución:

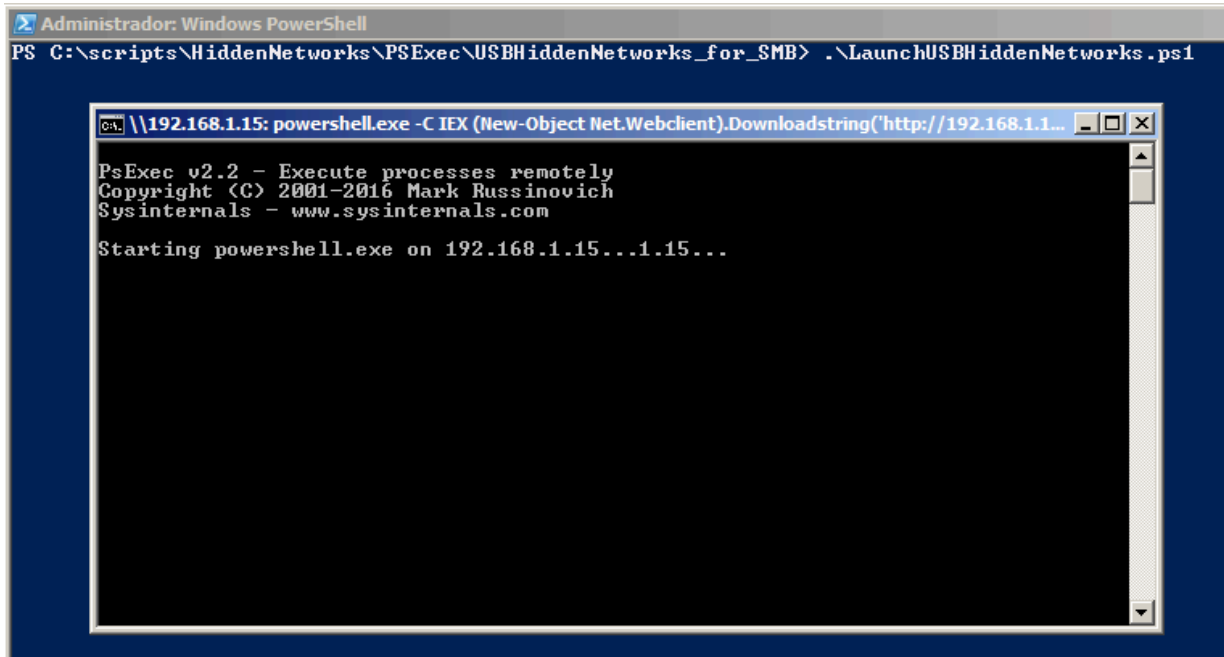


Figura 13: Ejecución de una Powershell y el script a través de la herramienta PSEXEC

El fichero servers.txt tendrá almacenado los nombres de los equipos o directamente la dirección IP como se puede apreciar en la siguiente imagen:

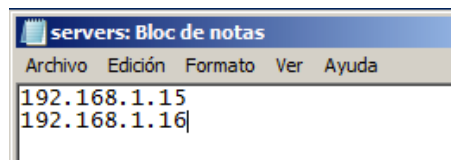
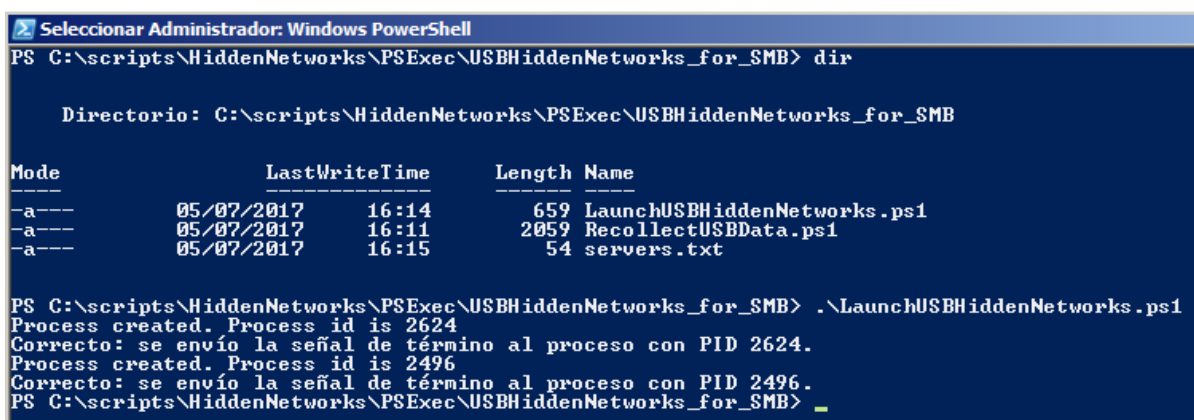


Figura 14: Listado de máquinas que se estudiarán

```
foreach ($computer in $computers) {
    $Process = [Diagnostics.Process]::Start("cmd.exe", "/c psexec.exe
        \\$computer powershell.exe -C IEX (New-Object
        Net.Webclient).Downloadstring('$url') >>
        C:\scripts\HiddenNetworks\PsExec\USBHiddenNetworks_for_SMB\
        usldata.csv")
    $id = $Process.Id
    sleep $sincro
    Write-Host "Process created. Process id is $id"
    taskkill.exe /PID $id
}
```

En este bucle se comprueba cada ordenador que vamos a analizar los cuales se han cargado en la variable \$computers desde el fichero servers.txt. La parte principal de la ejecución se centra en el objeto \$Process. En él, se abre una consola en el equipo remoto que a su vez lanzará otra consola de Powershell pasando como parámetro el fichero RecollectUSBData.ps1 localizado en la ubicación designada por la variable \$url. Es importante tener configuradas correctamente las rutas de ubicación de todos los ficheros antes de ejecutar el script.

Antes de pasar al siguiente ordenador de la lista será necesario estar seguros de que se ha finalizado el proceso de recopilar información. Hay muchas formas de optimizar esta operación, pero en este ejemplo se ha optado, simplemente, por agregar un retardo de X segundos entre cada ejecución mediante el uso del comando sleep. Una vez ha terminado la recopilación de información del equipo a auditar, eliminamos el proceso en ejecución antes de pasar al siguiente con el comando taskkill. A modo de información, se imprime por pantalla el ID y el resultado de dicha operación como se puede apreciar en la siguiente captura:



```

Selecionar Administrador: Windows PowerShell
PS C:\scripts\HiddenNetworks\PSExec\USBHiddenNetworks_for_SMB> dir

Directorio: C:\scripts\HiddenNetworks\PSExec\USBHiddenNetworks_for_SMB

Mode                LastWriteTime         Length Name
----                -
-a----             05/07/2017   16:14           659 LaunchUSBHiddenNetworks.ps1
-a----             05/07/2017   16:11          2059 RecollectUSBData.ps1
-a----             05/07/2017   16:15            54 servers.txt

PS C:\scripts\HiddenNetworks\PSExec\USBHiddenNetworks_for_SMB> .\LaunchUSBHiddenNetworks.ps1
Process created. Process id is 2624
Correcto: se envió la señal de término al proceso con PID 2624.
Process created. Process id is 2496
Correcto: se envió la señal de término al proceso con PID 2496.
PS C:\scripts\HiddenNetworks\PSExec\USBHiddenNetworks_for_SMB>
  
```

Figura 15: Ejecución del script en Powershell

2.3.2.Script: RecollectUSBData

Este script sólo se ha modificado en el último bloque, el Bloque 3, para adaptar la salida al nuevo tipo de ejecución. Como se puede apreciar en el código mostrado a continuación, se ha sustituido el comando Echo por un Write-Host con las variables, eliminado la salida por pantalla:

```

for ($i=0; $i -lt $USBDevices.length; $i++) {
    $IDUnico=$USBDevices[$i] | Select -ExpandProperty "USBContainerID"
    $USBNombre=$USBDevices[$i] | Select -ExpandProperty "USBDevice"
    Write-Host "$ComputerName,$ComputerIP,$USBNombre,$IDUnico"
}
  
```

El fichero generado USBData.CSV será exactamente igual al mostrado en el apartado anterior.

2.4.Información histórica

También es posible obtener un registro de fechas de la primera conexión al ordenador, en caso de necesitar más información sobre el recorrido del dispositivo USB dentro de la HiddenNetwork. En el registro de eventos, la rama que más información puede ofrecer viene deshabilitada por defecto en todas las versiones de Windows. Dicha rama es la siguiente:

Windows Logs -> Applications and Services Logs -> Windows -> DriverFrameworks-UserMode -> Operational

Por lo tanto, la forma de obtener la fecha de la primera conexión del dispositivo USB al equipo, sin tener que activar la auditoría en los equipos, es analizar el siguiente fichero de sistema:

`C:\Windows\inf\setuoapi.dev.log`

En dicho fichero se tienen registrados, entre otros datos, la fecha de la primera vez que se conectó. Para poder localizar correctamente el dispositivo USB insertado será necesario almacenar un nuevo valor durante la ejecución del script "RecollectUSBData.ps1", dicho valor es el campo DiskID:

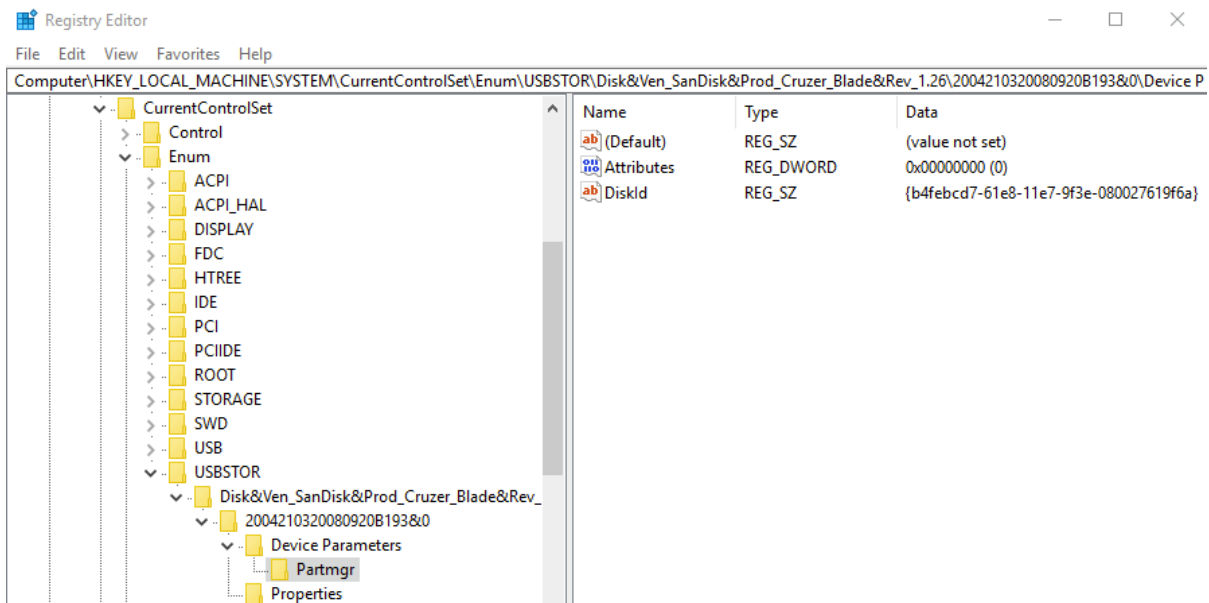


Figura 16: Búsqueda de la clave DiskID

Este valor es único dentro del sistema Windows actual, pero varía al conectarlo a otro ordenador, a diferencia del campo ContainerId, el cual es el mismo en cada ordenador Windows. Con este valor se puede identificar el dispositivo USB dentro del fichero setuoapi.dev.log. En la siguiente imagen se puede mostrar su ubicación utilizando su DiskID y la fecha de su primera inserción en el sistema auditado:

```

264 dvi: Device post-install completed. 18:16:50.685
265 <<< Section end 2017/07/05 18:16:50.794
266 <<< [Exit status: SUCCESS]
267
268
269 >>> [Device Install (Hardware initiated) - SWD\WPDBUSENUM\{b4febcd7-61e8-11e7-9f3e-080027619f6a}#000000000004000]
270 >>> Section start 2017/07/05 18:41:28.029
271 dvi: {Build Driver List} 18:41:28.076
272 dvi: Searching for compatible ID(s):
273 dvi: wpdbusenum\fs
274 dvi: swd\generic
275 dvi: Created Driver Node:
276 dvi: HardwareID - wpdbusenum\fs
277 dvi: InfName - C:\Windows\System32\DriverStore\FileRepository\wpdfs.inf_amd64_e898714e5623f0fe\
278 dvi: DevDesc - WPD FileSystem Volume Driver
279 dvi: Section - Basic_Install
280 dvi: Rank - 0x00ff2000
281 dvi: Signer Score - INBOX
282 dvi: DrvDate - 06/21/2006
283 dvi: Version - 10.0.15063.0
284 dvi: {Build Driver List - exit(0x00000000)} 18:41:28.122
285 dvi: {DIF_SELECTBESTCOMPATDRV} 18:41:28.122
286 dvi: Using exported function 'WpdClassInstaller' in module 'C:\Windows\system32\wpd_ci.dll'.
287 dvi: Class installer == wpd_ci.dll,WpdClassInstaller
288 dvi: Class installer: Enter 18:41:28.170
289 dvi: Class installer: Exit
290 dvi: Default installer: Enter 18:41:28.170
291 dvi: {Select Best Driver}
292 dvi: Class GUID of device changed to: {eec5ad98-8080-425f-922a-dabf3de3f69a}.
293 dvi: {DIF_DESTROYPRIVATEDATA} 18:41:28.170
294 dvi: Class installer: Enter 18:41:28.170
295 dvi: Class installer: Exit
296 dvi: Default installer: Enter 18:41:28.170
    
```

Figura 17: Obtención de la fecha de conexión del dispositivo USB

2.5.Hidden Links en OS X

En los equipos que ejecutan Mac OS X o macOS disponen de un fichero con extensión PLIST, el cual almacena esta información sobre dispositivos USB conectados al equipo. El fichero se denomina com.apple.finder.plist. En la imagen se puede visualizar un ejemplo de captura de información en entonos OS X o macOS.

The image shows a file explorer window with a list of files on the left and a preview of the selected file, `com.apple.finder.plist`, on the right. The preview displays XML-style key-value pairs. Three keys are highlighted with red boxes:

- `<key>Adium_1.4.3_0x1.413c891p+28</key>`
- `<key>BOOTCAMP_0x1.d27e44p+29</key>`
- `<key>Citrix_Receiver_0x1.33a352bp+28</key>`

Figura 18: Recopilación de información sobre dispositivos USB en sistemas OS X

2.6. Mitigación

La forma de evitar este tipo de “Polinización” entre equipos dentro de la red de una organización consiste en limitar el uso de dispositivos USB en los equipos. La mitigación o prevención a través del uso forzado por políticas de Active Directory que solo se pueda conectar en el equipo de un usuario los dispositivos autorizados para él. La aplicación de una política de seguridad con una lista blanca de dispositivos autorizados para cada usuario permite evitar este tipo de Hidden Links, pero es algo complejo y costoso de mantener.

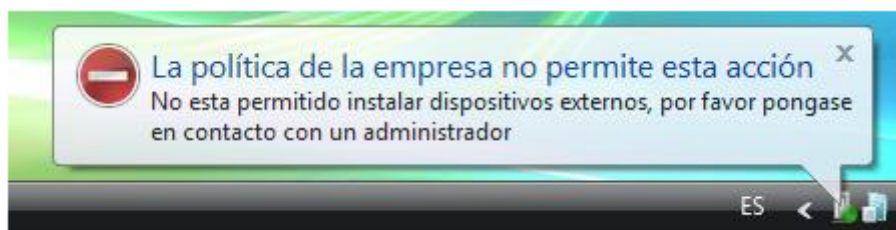


Figura 19: Por política se prohíbe el uso del dispositivo USB en dicha máquina

3. Conclusiones

Más allá de la potencial fuga de información de la empresa, una red oculta o HiddenNetwork es también un problema para la integridad de nuestro sistema. Estos dispositivos USB podrían propagar un malware hasta diferentes secciones de la infraestructura, donde en teoría, la seguridad es mayor. Tener redes desconectadas de Internet transmite la falsa seguridad de estar más protegidos ante cualquier incidente, lo que hace aún más vulnerable el sistema.

La infección a través de un dispositivo USB de un malware es un problema latente y actual, no sólo en el ya mencionado histórico caso de Stuxnet, sino en otros de mayor actualidad como Brutal Kangaroo utilizado por la CIA.

Debido al gran impacto en nuestra infraestructura que puede tener este tipo de infecciones y fuga de información, hemos creado este documento para ayudar a localizar estas redes ocultas y ofrecer una herramienta para su control. De ese modo será más fácil evitar incidentes y ofrecer también una utilidad que ofrezca información útil para casos de análisis forense.

Referencias

<http://www.elladodelmal.com/2014/02/como-localizar-los-hidden-links-de-las.html>

<http://www.elladodelmal.com/2017/06/script-powershell-winrm-para-descubrir.html>

<https://blogs.technet.microsoft.com/heyscriptingguy/2012/05/18/use-powershell-to-find-the-history-of-usb-flash-drive-usage/>

<http://www.elladodelmal.com/2017/06/brutal-kangaroo-y-la-infeccion-por-usb.html>

<https://github.com/ElevenPaths/USBHiddenNetworks>

Más información

www.elevenpaths.com

[@ElevenPaths](https://twitter.com/ElevenPaths)

blog.elevenpaths.com

2017 © Telefónica Digital España, S.L.U. Todos los derechos reservados.

La información contenida en el presente documento es propiedad de Telefónica Digital España, S.L.U. ("TDE") y/o de cualquier otra entidad dentro del Grupo Telefónica o sus licenciantes. TDE y/o cualquier compañía del Grupo Telefónica o los licenciantes de TDE se reservan todos los derechos de propiedad industrial e intelectual (incluida cualquier patente o copyright) que se deriven o recaigan sobre este documento, incluidos los derechos de diseño, producción, reproducción, uso y venta del mismo, salvo en el supuesto de que dichos derechos sean expresamente conferidos a terceros por escrito. La información contenida en el presente documento podrá ser objeto de modificación en cualquier momento sin necesidad de previo aviso.

La información contenida en el presente documento no podrá ser ni parcial ni totalmente copiada, distribuida, adaptada o reproducida en ningún soporte sin que medie el previo consentimiento por escrito por parte de TDE.

El presente documento tiene como único objetivo servir de soporte a su lector en el uso del producto o servicio descrito en el mismo. El lector se compromete y queda obligado a usar la información contenida en el mismo para su propio uso y no para ningún otro.

TDE no será responsable de ninguna pérdida o daño que se derive del uso de la información contenida en el presente documento o de cualquier error u omisión del documento o por el uso incorrecto del servicio o producto. El uso del producto o servicio descrito en el presente documento se regulará de acuerdo con lo establecido en los términos y condiciones aceptados por el usuario del mismo para su uso.

TDE y sus marcas (así como cualquier marca perteneciente al Grupo Telefónica) son marcas registradas. TDE y sus filiales se reservan todos los derechos sobre las mismas.