Detectando amenazas en vuestros equipos Windows

Técnicas de persistencia

Contenidos

- Introducción
- Contexto
- Técnicas
  - Registro de Windows
  - Scheduled tasks
  - Windows Management Instrumentation

$> Whoami

- **Marcos Rivera Martínez**
  - Ingeniero Informático
  - Investigador en Inteligencia Artificial y Ciberseguridad en Ideas Locas – Telefónica
  - **Twitter: @marcos_98_rm**

- **Alberto Rivera Martínez**
  - Ingeniero Informático
  - Investigador y desarrollador en Aura Prototypes – Telefónica
  - **Twitter: @ariveram2111**

c0r0n4con

Esto lo paramos hackeando

$> Introducción

No solo detectar intrusión, también existencia de atacantes.

Según Microsoft, los atacantes pueden permanecer meses en la organización hasta ser detectados.

$> Fases de un ataque

Recolección de información

Análisis de la información

Explotación

Post-explotación →

¡Técnicas de persistencia!

Objetivo: mantener conexión con el sistema vulnerado

Borrado de huellas

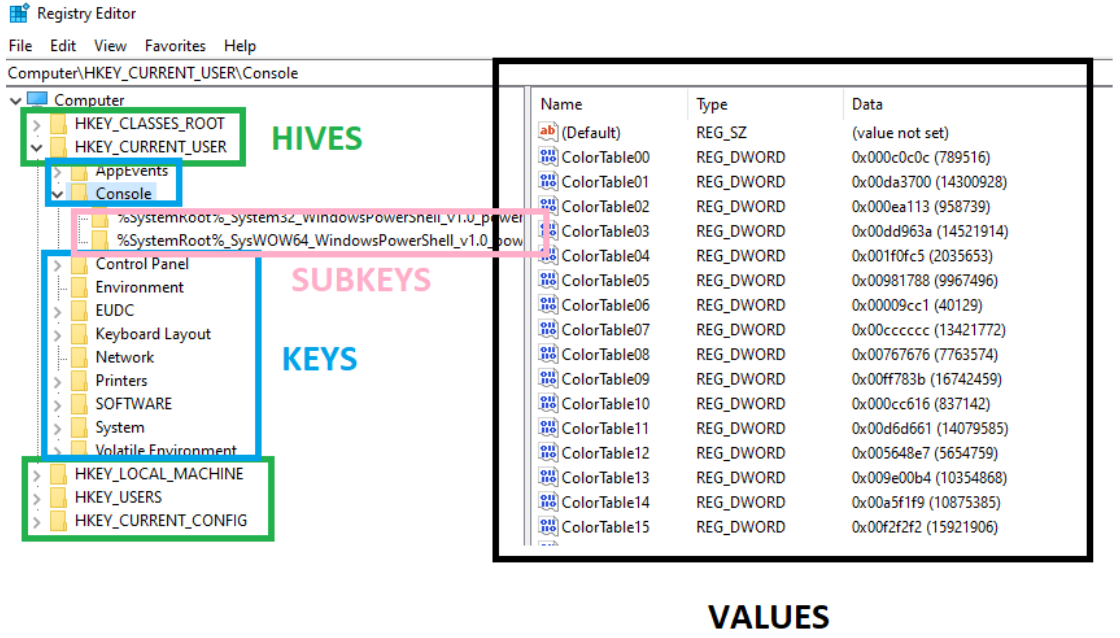
$> ¿Qué vamos a utilizar para persistir en el sistema? → PowerShell

- Línea de comandos orientada a objetos.
- Administración de sistemas
- Interacción con todos los productos Microsoft y sistema operativo
- Desde Windows Vista PowerShell viene instalado por defecto.
- Disponible para sistemas Linux.
- Acceso completo al Framework .NET

(New-Object Net.WebClient).DownloadString(“url”)

$> Qué es el registro de Windows

- Base de datos jerárquica que almacena información de configuración.
- Almacena información sobre:
  - Aplicaciones instaladas.
  - Tipos de documentos que se pueden crear.
  - Aplicaciones predeterminadas para cada tipo de documento.
  - Perfiles de usuario.
  - Etc
- El registro se compone de “keys / hives” (claves de registro).
- Cada clave se compone de subclaves. A su vez, estas subclaves se componen de otras subclaves.
- Las subclaves pueden contener valores o propiedades



$> Qué es el registro de Windows

Carpeta/clave predefinida	Descripción
HKEY_CURRENT_USER	Contiene la raíz de la información de configuración del usuario que ha iniciado sesión actualmente. Las carpetas del usuario, los colores de la pantalla y la configuración del panel de control se almacenan aquí. Esta información está asociada con el perfil del usuario. Esta clave a veces se abrevia como *HKCU*.
HKEY_USERS	Contiene todos los perfiles de usuario cargados activamente en el equipo. HKEY_CURRENT_USER es una subclave de HKEY_USERS. HKEY_USERS a veces se abrevia como *HKU*.
HKEY_LOCAL_MACHINE	Contiene información de configuración específica del equipo (para cualquier usuario). Esta clave a veces aparece abreviada como *HKLM*.
HKEY_CLASSES_ROOT	Es una subclave de `HKEY_LOCAL_MACHINE\Software` . La información que se almacena aquí garantiza que se abre el programa correcto al abrir un archivo con el explorador de Windows. Esta clave a veces se abrevia como *HKCR*. A partir de Windows 2000, esta información se almacena en las claves HKEY_LOCAL_MACHINE y HKEY_CURRENT_USER. La `HKEY_LOCAL_MACHINE\Software\Classes` clave contiene la configuración predeterminada que se puede aplicar a todos los usuarios del equipo local. La `HKEY_CURRENT_USER\Software\Classes` clave contiene la configuración que reemplaza la configuración predeterminada y se aplica solo al usuario interactivo. La clave HKEY_CLASSES_ROOT proporciona una vista del registro que combina la información de estos dos orígenes. HKEY_CLASSES_ROOT también proporciona esta vista combinada para los programas diseñados para versiones anteriores de Windows. Para cambiar la configuración del usuario interactivo, los cambios deben realizarse en en `HKEY_CURRENT_USER\Software\Classes` lugar de en HKEY_CLASSES_ROOT. Para cambiar la configuración predeterminada, los cambios deben realizarse en `HKEY_LOCAL_MACHINE\Software\Classes` . Si escribe claves en una clave en HKEY_CLASSES_ROOT, el sistema almacena la información en `HKEY_LOCAL_MACHINE\Software\Classes` . Si escribe valores en una clave en HKEY_CLASSES_ROOT y la clave ya existe en `HKEY_CURRENT_USER\Software\Classes` , el sistema almacenará la información allí, en lugar de en `HKEY_LOCAL_MACHINE\Software\Classes` .
HKEY_CURRENT_CONFIG	Contiene información acerca del perfil de hardware que usa el equipo local al iniciar el sistema.

Fuente: https://docs.microsoft.com/es-es/troubleshoot/windows-server/performance/windows-registry-advanced-users

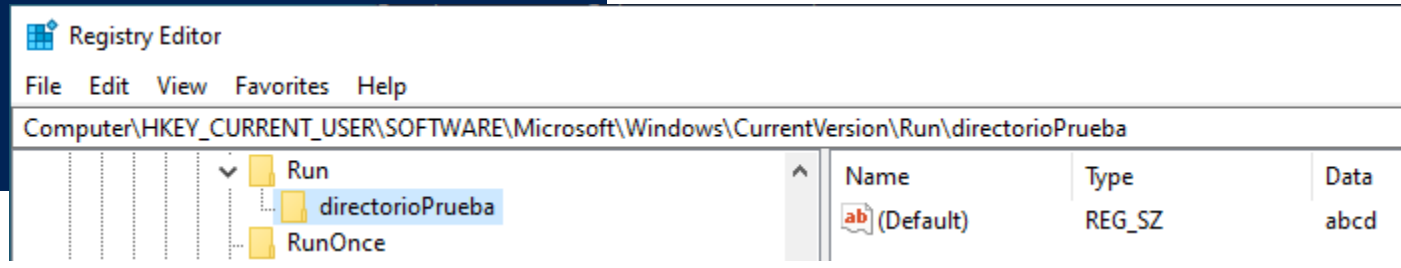
$> Comandos útiles de PowerShell para el registro de Windows

- Nos podemos mover por el registro de Windows como si fuese el sistema de ficheros gracias a los providers de PowerShell.

```
PS C:\Users\mrive> Set-Location HKCU:
PS HKCU:\> Get-ChildItem


    Hive: HKEY_CURRENT_USER


Name                           Property
----                           --------
AppEvents
Console                        ColorTable00            : 789516
                               ColorTable01            : 14300928
```

- Podemos crear claves de registro

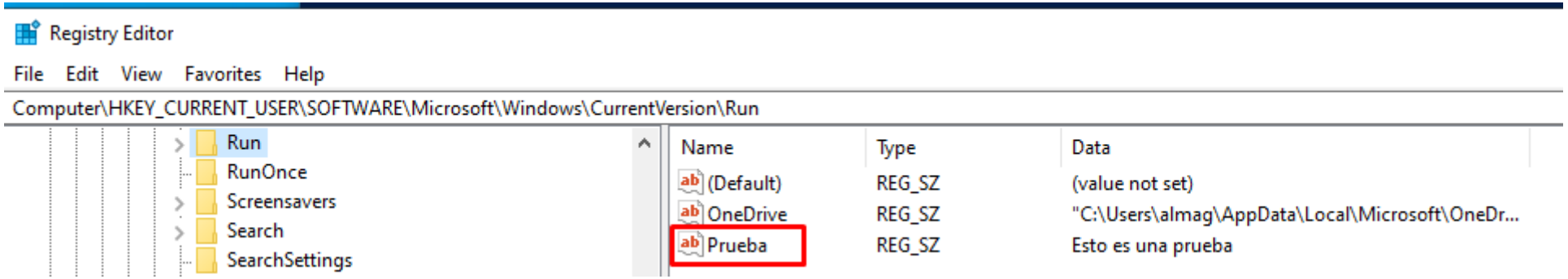
```
PS HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\run> New-Item -Path ./directorioPrueba -Value "abcd" -ItemType String


    Hive: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\run


Name                           Property
----                           --------
directorioPrueba               (default) : abcd
```



$> Comandos útiles de PowerShell para el registro de Windows

- También podemos crear valores.

```
PS HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\run> New-ItemProperty -Name "Prueba" -Value "Esto es una prueba" -Path ./ -PropertyType String


Prueba       : Esto es una prueba
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion
PSChildName  : run
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```

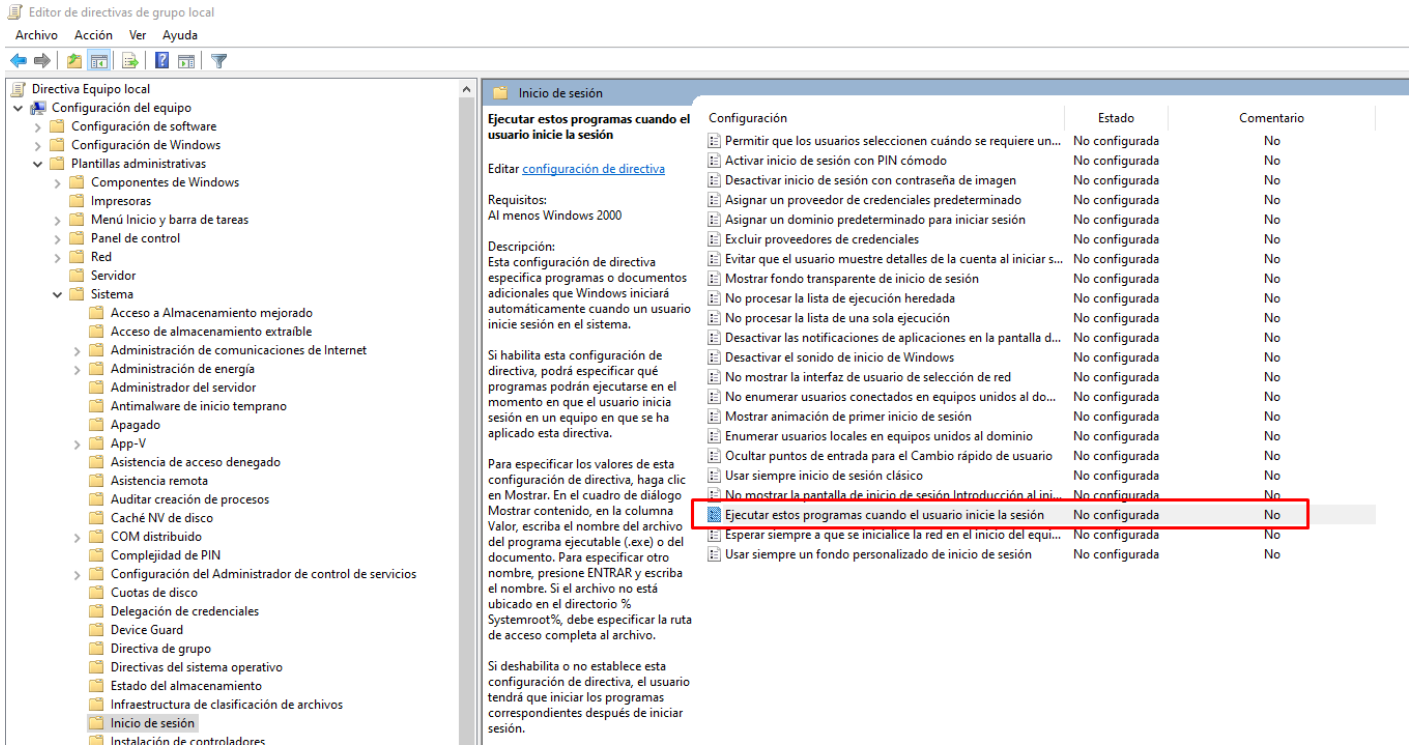


$> Persistencia usando el registro de Windows

- Run Keys por defecto en Windows:
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- Configuración de start up folders:
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- Claves de registro para ejecutar servicios de forma automática:
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices

$> Persistencia usando el registro de Windows

- Rutas de configuración de políticas para especificar programas de inicio:
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

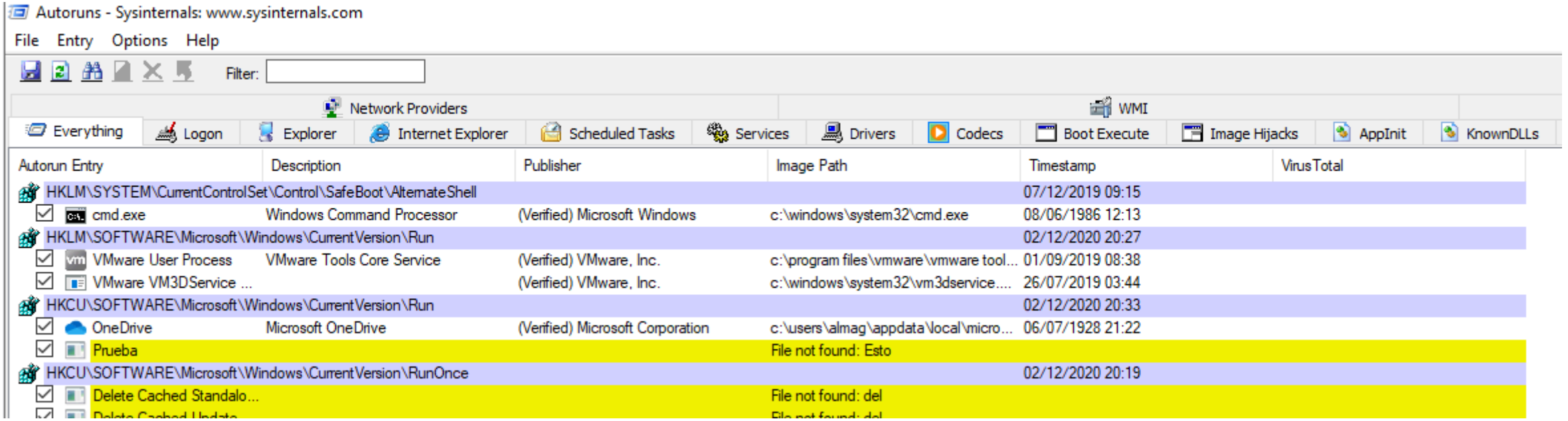


$> Persistencia usando el registro de Windows

- Otras ramas menos comunes:
  - HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
  - HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
- Se ejecutan todos los programas ubicados en la siguiente Startup folder:
  - C:\Users[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
  - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

$> Detección de persistencia usando el registro de Windows

- Uso de Sysinternals > Autoruns
- https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite



$> Detección de persistencia usando Sysmon

- Uso de Sysinternals > Sysmon
- Permite detectar numerosos eventos
- Configuración de Sysmon

```
PS C:\Users\almag\Downloads> .\Sysmon64.exe -accepteula -i .\sysmonconfig-export.xml


System Monitor v12.03 - System activity monitor
Copyright (C) 2014-2020 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.22
Sysmon schema version: 4.40
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.
PS C:\Users\almag\Downloads>
```

```
    <!--DATA: UtcTime, ProcessGuid, ProcessID, Image, FileVersion, Description, Product, Company, CommandLine, CurrentDirectory, User, LogonGu
<RuleGroup name="" groupRelation="or">
    <ProcessCreate onmatch="exclude">
        <!--SECTION: Microsoft Windows-->
        <CommandLine condition="begin with"> "C:\Windows\system32\wermgr.exe" "-queuereporting_svc" </CommandLine> <!--Windows:Windows error :
        <CommandLine condition="begin with">C:\Windows\system32\DllHost.exe /Processid</CommandLine> <!--Windows-->
        <CommandLine condition="begin with">C:\Windows\system32\wbem\wmiprvse.exe -Embedding</CommandLine> <!--Windows: WMI provider host-->
        <CommandLine condition="begin with">C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding</CommandLine> <!--Windows: WMI provider
        <CommandLine condition="is">C:\Windows\system32\wermgr.exe -upload</CommandLine> <!--Windows:Windows error reporting/telemetry-->
        <CommandLine condition="is">C:\Windows\system32\SearchIndexer.exe /Embedding</CommandLine> <!--Windows: Search Indexer-->
        <CommandLine condition="is">C:\Windows\system32\wermgr.exe -queuereporting</CommandLine> <!--Windows:Windows error reporting/telemetry
        <CommandLine condition="is">\??\C:\Windows\system32\autochk.exe *</CommandLine> <!--Microsoft:Bootup: Auto Check Utility-->
        <CommandLine condition="is">\SystemRoot\System32\smss.exe</CommandLine> <!--Microsoft:Bootup: Windows Session Manager-->
        <CommandLine condition="is">C:\Windows\System32\RuntimeBroker.exe -Embedding</CommandLine> <!--Windows:Apps permissions [ https://fos
        <Image condition="is">C:\Program Files (x86)\Common Files\microsoft shared\ink\TabTip32.exe</Image> <!--Windows: Touch Keyboard and Ha
        <Image condition="is">C:\Windows\System32\TokenBrokerCookies.exe</Image> <!--Windows: SSO sign-in assistant for MicrosoftOnline.com--
        <Image condition="is">C:\Windows\System32\plasrv.exe</Image> <!--Windows: Performance Logs and Alerts DCOM Server-->
        <Image condition="is">C:\Windows\System32\wifitask.exe</Image> <!--Windows: Wireless Background Task-->
        <Image condition="is">C:\Windows\system32\CompatTelRunner.exe</Image> <!--Windows: Customer Experience Improvement-->
        <Image condition="is">C:\Windows\system32\PrintIsolationHost.exe</Image> <!--Windows: Printing-->
        <Image condition="is">C:\Windows\system32\SppExtComObj.Exe</Image> <!--Windows: KMS activation-->
        <Image condition="is">C:\Windows\system32\audiodg.exe</Image> <!--Windows: Launched constantly-->
```

$> Eventos que permite detectar Sysmon

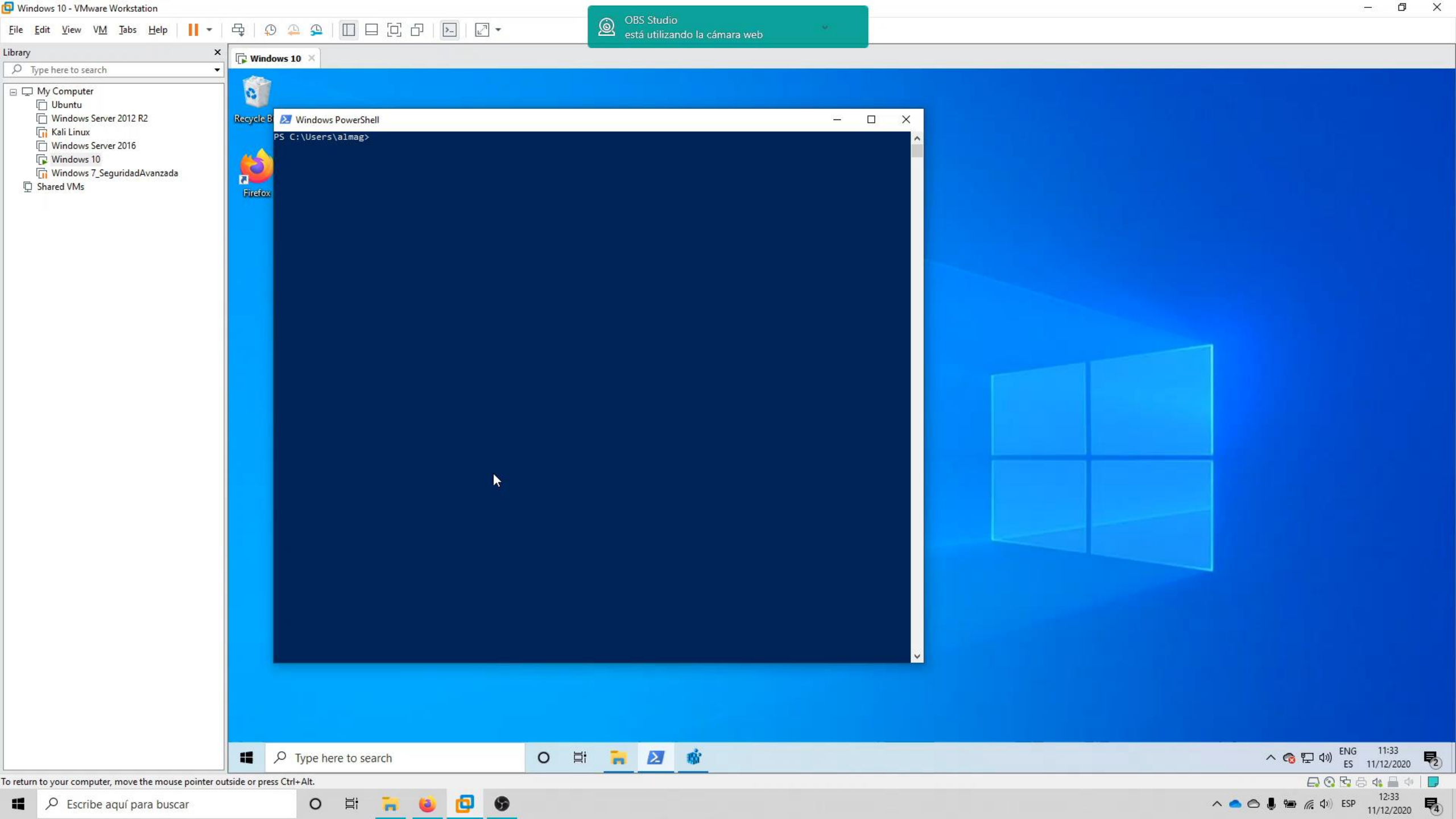
- Event ID 1: Process creation
- Event ID 3: Network connection
- Event ID 4: Sysmon service state changed
- Event ID 5: Process terminated
- Event ID 6: Driver loaded
- Event ID 7: Image loaded
- Event ID 12, 13 and 14: RegistryEvent
- Event ID 19, 20 and 21: WmiEvent
- Event ID 22: DNSEvent
- Event ID 23: FileDelete

$> Eventos de Sysmon para detectar persistencia usando el registro de Windows

- Event ID 12: RegistryEvent (Object create and delete)
- Event ID 13: RegistryEvent (Value Set)
- Event ID 14: RegistryEvent (Key and Value Rename)

```
Registry value set:
RuleName: T1060,RunKey
EventType: SetValue
UtcTime: 2020-05-06 11:59:09.466
ProcessGuid: {a4a2434a-6af3-5eb1-0000-0010d9f71700}
ProcessId: 4768
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
TargetObject: HKU\S-1-5-21-3519239064-3341317675-1809890911-1009\Software\Microsoft\Windows\CurrentVersion\Run\Prueba
Details: C:\Program Files (x86)\Notepad++\notepad++.exe
```

$> DEMO TIME


Windows 10 - VMware Workstation
File Edit View VM Tabs Help
OBS Studio
está utilizando la cámara web
Library
Type here to search
My Computer
Ubuntu
Windows Server 2012 R2
Kali Linux
Windows Server 2016
Windows 10
Windows 7_SeguridadAvanzada
Shared VMs
Windows 10
Recycle Bin
Firefox
Windows PowerShell
PS C:\Users\almag>
Type here to search
ENG
ES
11:33
11/12/2020
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
Escribe aquí para buscar
ESP
12:33
11/12/2020

$> Tareas programadas o scheduled tasks en Windows

- Automatizar la ejecución de programas y scripts
- Trigger o disparador
- Acción
- Usuario inicia sesión → Ejecutar script de PowerShell

```
PS C:\Users\almag\Downloads> Get-ScheduledTask

TaskPath                                    TaskName                          State
--------                                    --------                          -----
\                                           OneDrive Standalone Update Tas... Ready
\                                           PostponeDeviceSetupToast_S-1-5... Ready
\Microsoft\Windows\.NET Framework\          .NET Framework NGEN v4.0.30319    Ready
\Microsoft\Windows\.NET Framework\          .NET Framework NGEN v4.0.30319 64 Ready
\Microsoft\Windows\.NET Framework\          .NET Framework NGEN v4.0.30319... Disabled
\Microsoft\Windows\.NET Framework\          .NET Framework NGEN v4.0.30319... Disabled
\Microsoft\Windows\Active Directory Rights ... AD RMS Rights Policy Template ... Disabled
\Microsoft\Windows\Active Directory Rights ... AD RMS Rights Policy Template ... Ready
```

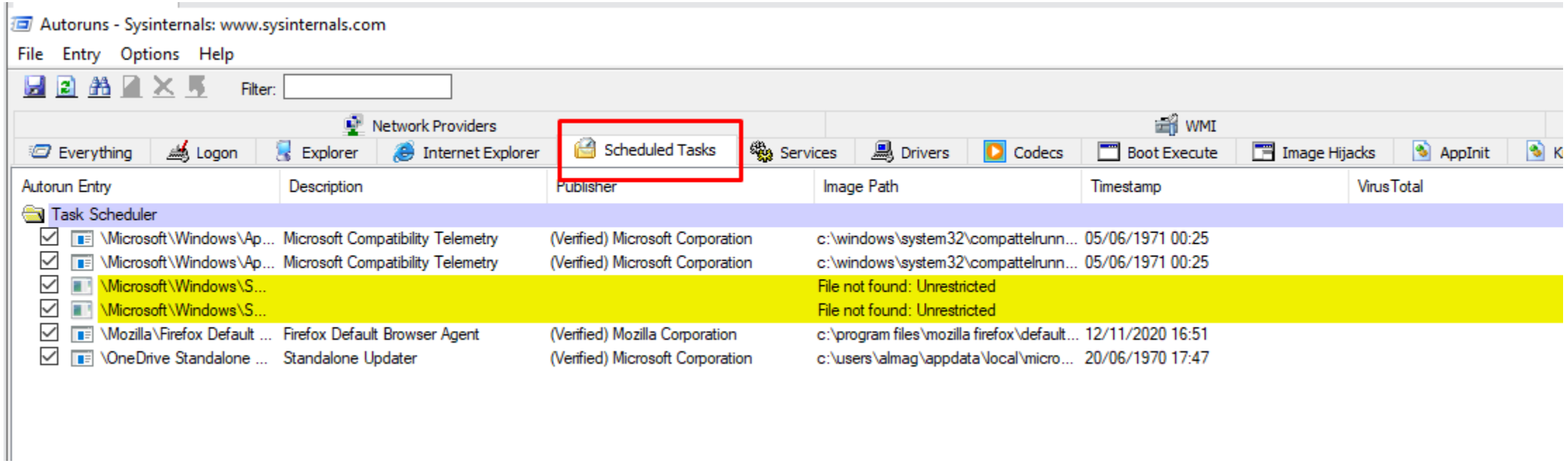
$> Cómo hacer persistencia usando Scheduled tasks

- Definir una tarea a realizar con el comando New-ScheduledTaskAction
- Definir el disparador con New-ScheduledTaskTrigger
- Registrar la tarea programada con Register-ScheduledTask

```
PS C:\Windows\system32> $TaskAction = New-ScheduledTaskAction -Execute 'C:\Windows\system32\WindowsPowerShell\v1.0\power
shell.exe' -Argument '-Command "Write-Host Esto es una prueba"'
PS C:\Windows\system32> $TaskTrigger = New-ScheduledTaskTrigger -AtLogOn
>>     $User = whoami
PS C:\Windows\system32> Register-ScheduledTask -Action $TaskAction -Trigger $TaskTrigger -TaskName "TFGMarcos" -Descript
ion "Prueba" -User $User -RunLevel Highest

TaskPath                                 TaskName                                 State
--------                                 --------                                 -----
\                                        TFGMarcos                                Ready
```

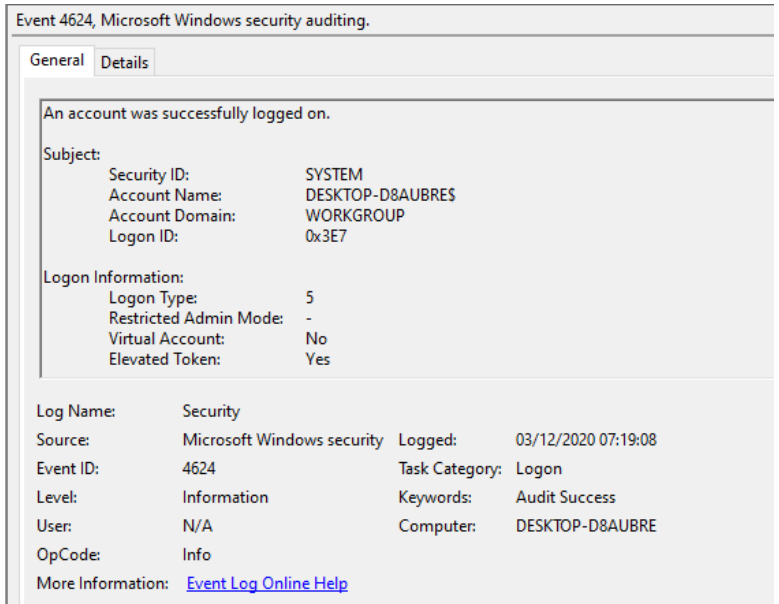
$> Cómo detectar la creación de Scheduled tasks

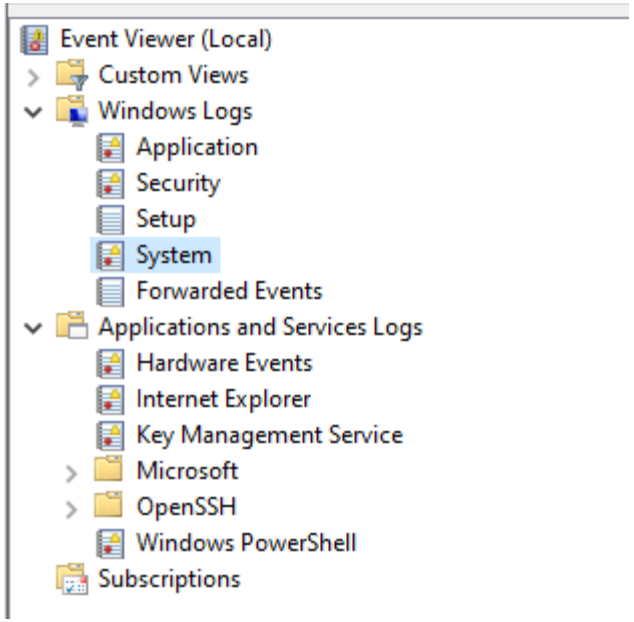
- Con Autoruns



$> Cómo detectar la creación de scheduled tasks

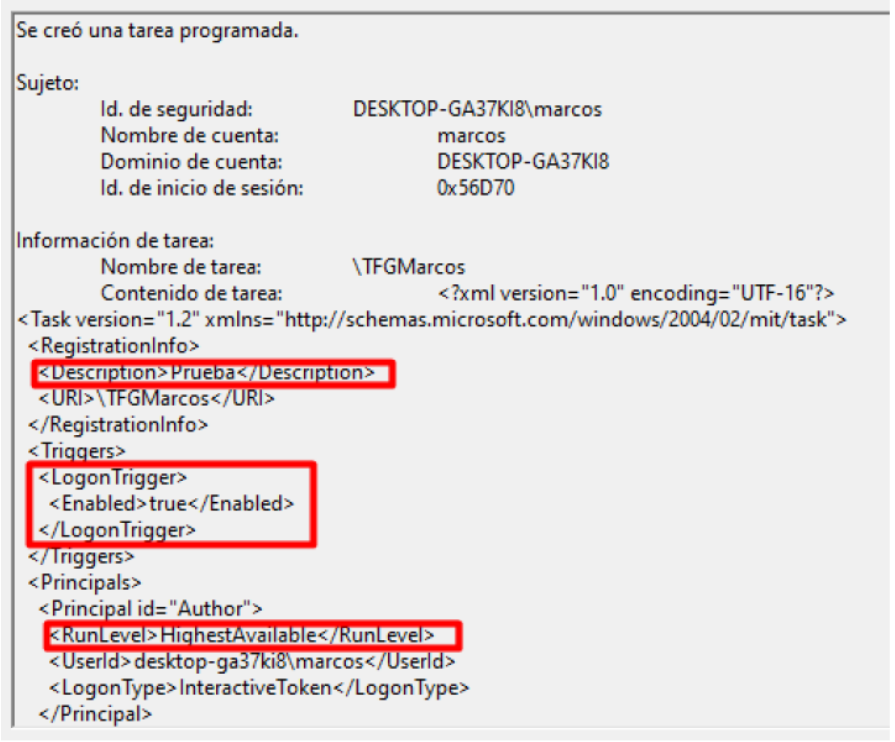
- Sysmon no permite detectar scheduled tasks.
- En su lugar podemos usar el sistema de eventos de Windows
  - Registros de Windows: System, Security, Application, Setup y Forwarded events
  - Registros de aplicaciones y servicios



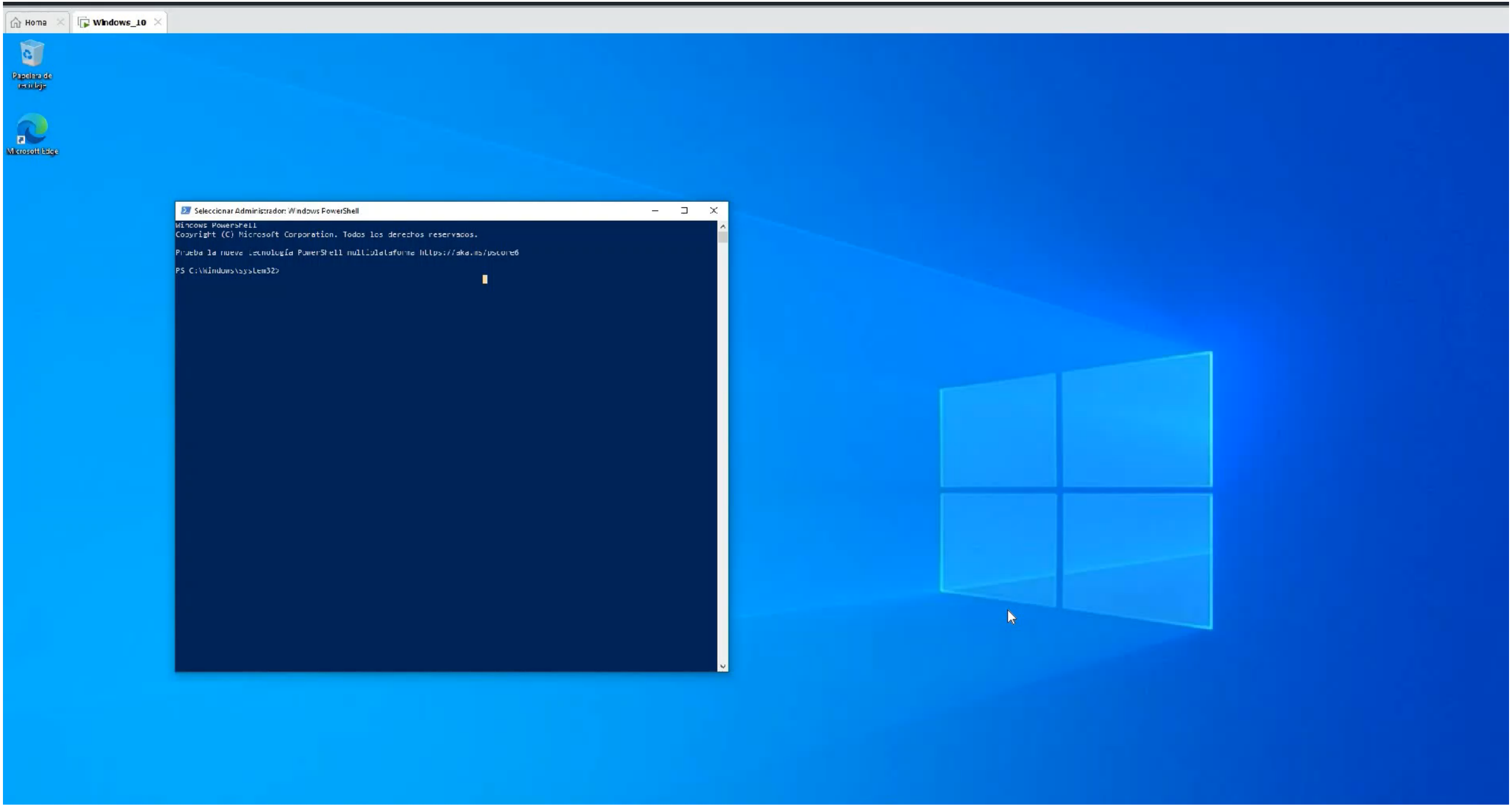


$> Cómo detectar la creación de scheduled tasks

- Activar política: *Auditpol /set /subcategory:"Otros eventos de acceso a objetos" /success:enable /failure:enable*
- Detección con el evento de seguridad de Windows con ID 4698



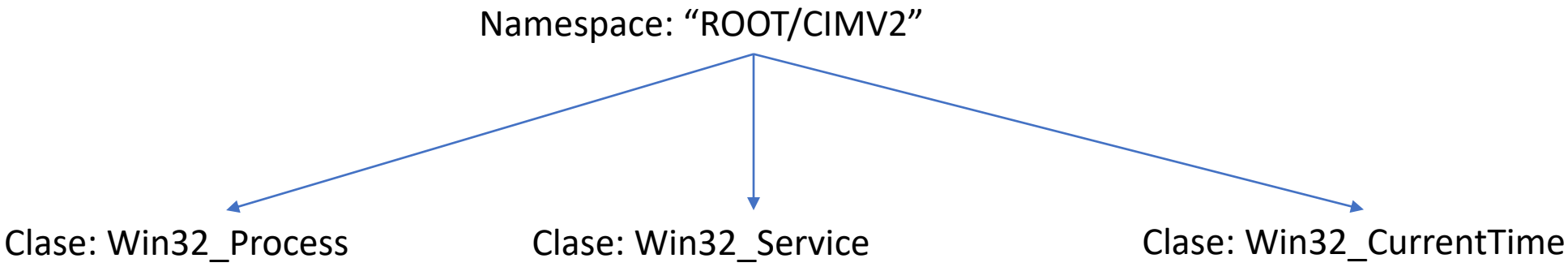
$> DEMO TIME


Home
Windows_10
Papelera de reciclaje
Microsoft Edge
Seleccionar Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6
PS C:\Windows\system32>

$> Windows Management Instrumentation

- Implementación de los estándares WBEM (Web-Based Enterprise Management) y CIM (Common Information Model), publicados por la DMTF (Distributed Management Task Force).
- Estos estándares permiten consultar y modificar diversos elementos del sistema operativo, como procesos que están corriendo en el sistema, claves de registro, servicios, etc.
- WMI permite realizar consultas utilizando un lenguaje similar a SQL, denominado WQL (WMI Query Language).
- Windows también cuenta con el estándar CIM que es una evolución de WMI. Cambian algunos aspectos como los protocolos utilizados, pero conceptualmente son muy similares.

$> Estructura de Windows Management Instrumentation




```
PS C:\Users\mrive> (Get-WmiObject -Class Win32_Process)[0] | Get-Member


   TypeName: System.Management.ManagementObject#root\cimv2\Win32_Process

Name                      MemberType     Definition
----                      ----------     ----------
Handles                   AliasProperty  Handles = Handlecount
ProcessName               AliasProperty  ProcessName = Name
PSComputerName            AliasProperty  PSComputerName = __SERVER
VM                        AliasProperty  VM = VirtualSize
WS                        AliasProperty  WS = WorkingSetSize
AttachDebugger            Method         System.Management.ManagementBaseObject AttachDebugger()
GetAvailableVirtualSize   Method         System.Management.ManagementBaseObject GetAvailableVirtualSize()
GetOwner                  Method         System.Management.ManagementBaseObject GetOwner()
GetOwnerSid               Method         System.Management.ManagementBaseObject GetOwnerSid()
SetPriority               Method         System.Management.ManagementBaseObject SetPriority(System.Int32 Priority)
Terminate                 Method         System.Management.ManagementBaseObject Terminate(System.UInt32 Reason)
```

$> Windows Management Instrumentation

- Obtención de productos antivirus con WMI:

```
PS C:\Users\mrive> Get-WmiObject -Namespace "root\SecurityCenter2" -Query "SELECT * FROM AntivirusProduct" | Select-Obje
ct -Property displayName, pathToSignedProductExe, pathToSignedReportingExe, timestamp, PSComputerName


displayName               : Windows Defender
pathToSignedProductExe    : windowsdefender://
pathToSignedReportingExe  : %ProgramFiles%\Windows Defender\MsMpeng.exe
timestamp                 : Sun, 31 May 2020 10:15:34 GMT
PSComputerName            : DESKTOP-1VVSIUF

displayName               : Kaspersky Total Security
pathToSignedProductExe    : C:\Program Files (x86)\Kaspersky Lab\Kaspersky Total Security 20.0\wmiav.exe
pathToSignedReportingExe  : C:\Program Files (x86)\Kaspersky Lab\Kaspersky Total Security 20.0\avp.exe
timestamp                 : Sat, 30 May 2020 18:00:41 GMT
PSComputerName            : DESKTOP-1VVSIUF
```

- WMI no solo permite recolectar información, sino que también permite realizar acciones sobre los programas y el sistema operativo

```
PS C:\Users\mrive\Desktop> (Get-WmiObject -Class Win32_Service) | Get-Member


   TypeName: System.Management.ManagementObject#root\cimv2\Win32_Service

Name                      MemberType     Definition
----                      ----------     ----------
PSComputerName            AliasProperty  PSComputerName = __SERVER
Change                    Method         System.Management.ManagementBaseObject Cha
ChangeStartMode           Method         System.Management.ManagementBaseObject Cha
Delete                    Method         System.Management.ManagementBaseObject Del
GetSecurityDescriptor     Method         System.Management.ManagementBaseObject Get
```

$> Persistencia usando Windows Management Instrumentation

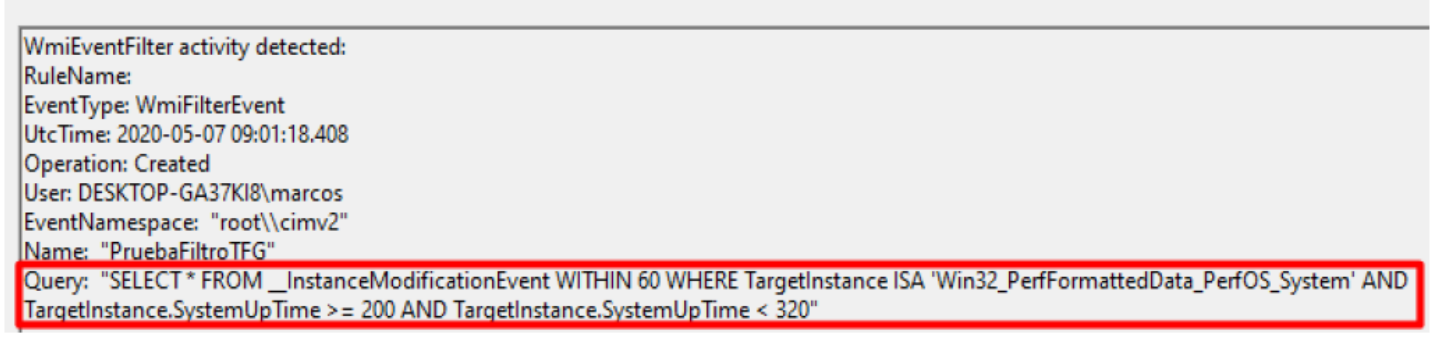
- WMI permite la suscripción a eventos:
  - Filtro de eventos
  - Consumidor de eventos
  - Ligar filtro al consumidor de eventos

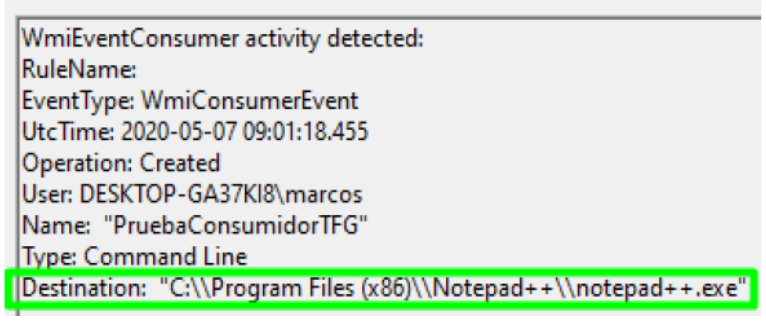
```
PS C:\Windows\system32> $filterName = 'FiltroPruebaTFG'
PS C:\Windows\system32> $consumerName = 'ConsumidorPruebaTFG'
PS C:\Windows\system32> $exePath = 'C:\Program Files (x86)\Notepad++\Notepad++.exe'
PS C:\Windows\system32> $Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_Pe
rfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 200 AND TargetInstance.SystemUpTime < 320"
PS C:\Windows\system32> $WMIEventFilter = Set-WmiInstance -Class __EventFilter -NameSpace "root\subscription" -Arguments
 @{Name=$filterName;EventNameSpace="root\cimv2";QueryLanguage="WQL";Query=$Query} -ErrorAction Ignore
PS C:\Windows\system32> $WMIEventConsumer = Set-WmiInstance -Class CommandLineEventConsumer -Namespace "root\subscriptio
n" -Arguments @{Name=$consumerName;ExecutablePath=$exePath;CommandLineTemplate=$exePath} -ErrorAction Ignore
PS C:\Windows\system32> Set-WmiInstance -Class __FilterToConsumerBinding -Namespace "root\subscription" -Arguments @{Fil
ter=$WMIEventFilter;Consumer=$WMIEventConsumer}
```

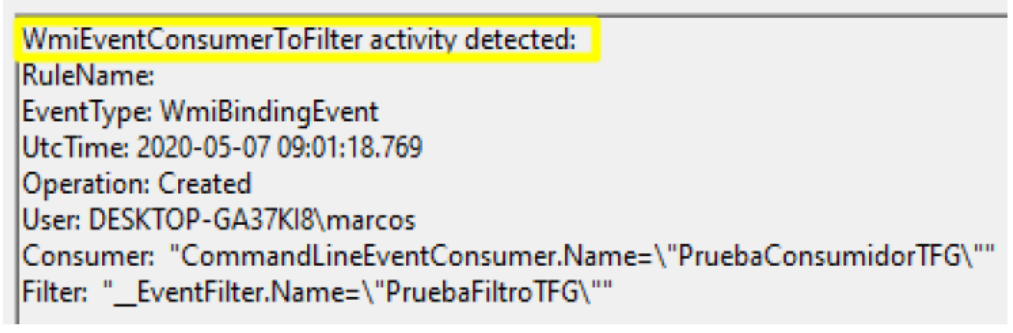
Inspirado en: [us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent Asynchronous-And-Fileless-Backdoor-wp.pdf (blackhat.com)]

$> Detección de persistencia realizada con WMI

- Con Sysmon:
  - Event ID 19: WmiEvent (WmiEventFilter activity detected)
  - Event ID 20: WmiEvent (WmiEventConsumer activity detected)
  - Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)







$> Sigma


Consultas generadas
Consultas
Sigma
Programa
Almacén de eventos
Reglas de detección
Eventos
(también es necesario especificar el sistema de almacenamiento de logs como Splunk, Powershell, etc)

$> DeepBlueCli

- Eric Conrad
- Programada en PowerShell
- Versión disponible en Python
- Descargar desde GitHub
- Analiza logs en busca de amenazas
  - Offline
  - Online

$> DEMO TIME

$> Referencias

- https://attack.mitre.org/techniques/T1547/001/
- https://attack.mitre.org/techniques/T1053/
- https://attack.mitre.org/techniques/T1047/
- https://www.elladodelmal.com/2020/10/deepbluecli-una-herramienta-para-hacer.html
- https://docs.microsoft.com/es-es/troubleshoot/windows-server/performance/windows-registry-advanced-users
- https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

$> exit 0;