



# D.E.A.T.H.

Ingeniería de detección en procesos de Threat Hunting

# \$ whoami\_

```
~/D/D.E.A.T.H. 1

$ id

uid=1001(Abraham Vargas) gid=1001(@0ldb14ck)
groups=1001(threat-hunter),1003(hackers),1004(open-source-contributors)

$ id

uid=1002(Yael Basurto) gid=1002(@_zkvL)
groups=1002(offensive-sec),1003(hackers),1005(Bishop-Fox),1006(Bsides-CDMX)
```





# Agenda

- ▶ Intro - Repaso del pre-work
- ▶ Laboratorios
- ▶ Cyber Kill Chain Hunting
- ▶ Cierre, preguntas y conclusiones

**BUGCON**  
2024



**SAFETY IS JUST A MYTH**

21 - 22 NOVIEMBRE 2024

CDMX



# BUGCON



## ¿Dudas del prework?

**OMITIR INTRO**





# Laboratorios (1.1)

¡Construye tus propias detecciones!

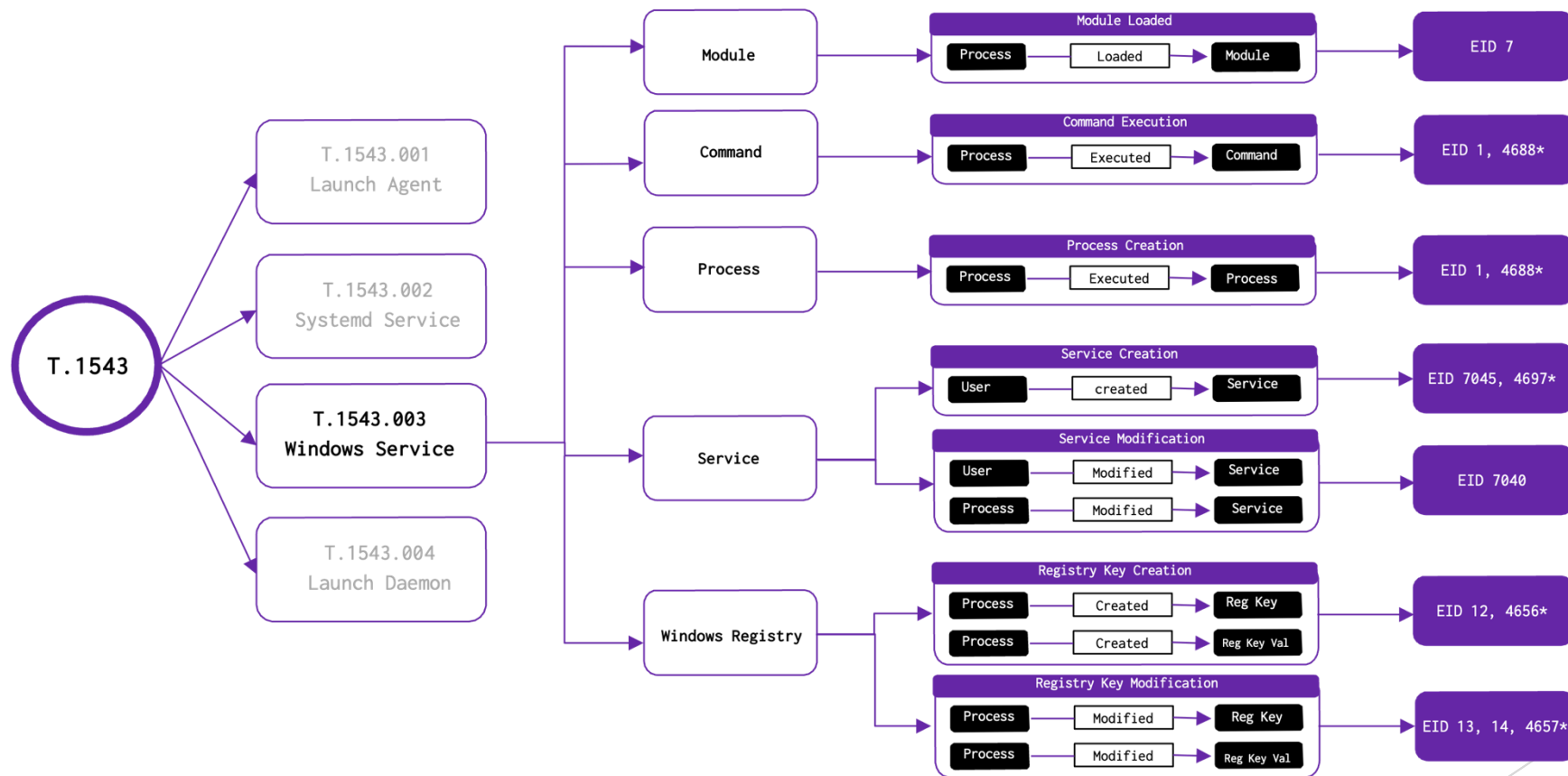
# Laboratorios (1.1)

Ingeniería de detección

## PREVIOUSLY ON...

# Laboratorios (1.1)

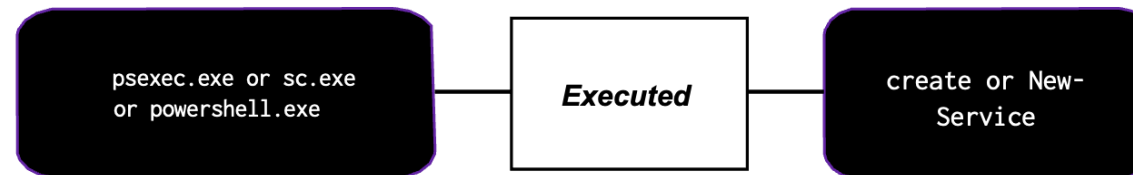
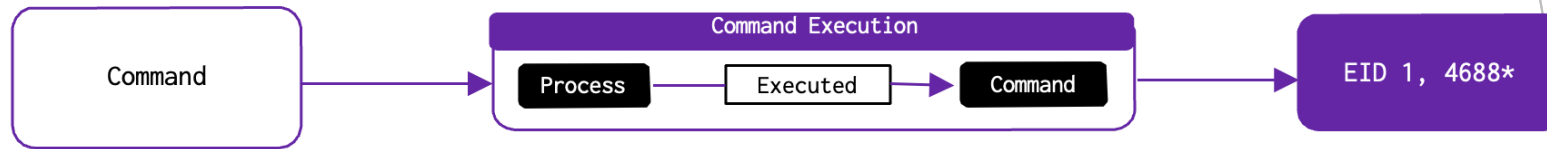
## Ingeniería de detección





# Laboratorios (1.1)

## Ingeniería de detección





# Laboratorios (1.1)

## Ingeniería de detección

Buscar cualquier proceso llamado psexec.exe o sc.exe o powershell.exe que tenga como parámetro de línea comandos "create" o "new-service".

event\_id: 1

process\_name:  
(psexec.exe or sc.exe  
or powershell)

process\_command\_line:  
(create or new-service)

event\_id: 1 AND process\_name: (psexec.exe OR sc.exe OR powershell.exe) AND process\_command\_line: ("create" OR "new-service")

# Laboratorios (1.1)

## Ingeniería de Detección

### Objetivo:

- Convertir las fuentes de datos presentes en la base de conocimiento del Mitre ATT&CK en posibles consultas que pueden ser usadas para identificar el uso de la técnica de Windows Services



# Laboratorios (1.2)

¡Crea tu propio ADS!

# Laboratorios (1.2)

## Alert and Detection Strategy - Retos

Algunos problemas potenciales experimentados por el desarrollo de alertas incluyen:

- La alerta no tiene suficiente documentación
- La alerta no está validada para la durabilidad
- La alerta no se revisa antes de su puesta en producción.



# Laboratorios (1.2)

## Alert and Detection Strategy - Una alternativa

- Para combatir los problemas y deficiencias notadas anteriormente, se creó el Framework ADS, el cual es usado para todos los desarrollos de alertamiento.
- Esta plantilla en lenguaje natural ayuda a la generación de hipótesis, prueba y administración de ADS
- Es un conjunto de plantillas de documentación, procesos y convenciones relacionadas con el diseño, la implementación y el despliegue de ADS.

# Laboratorios (1.2)

## Alert and Detection Strategy - Una alternativa





# Laboratorios (1.2)

## Alert and Detection Strategy - Una alternativa



# Laboratorios (1.2)

## Alert and Detection Strategy - Ingeniería de Detección

- La metodología de ingeniería de detección permite crear detecciones utilizando un proceso robusto y repetible
- Estos pasos metódicos permiten obtener información que puede ser usada no solo en el desarrollo de detecciones sino también para documentar su alertamiento y respuesta.
- Tomando toda esta información generada en el desarrollo de una detección es posible trasladarla a nuestro ADS.



# Laboratorios (1.2)

## Alert and Detection Strategy - Ingeniería de Detección

1  
Seleccionar una técnica objetivo.

2  
Investigar la tecnología asociada.

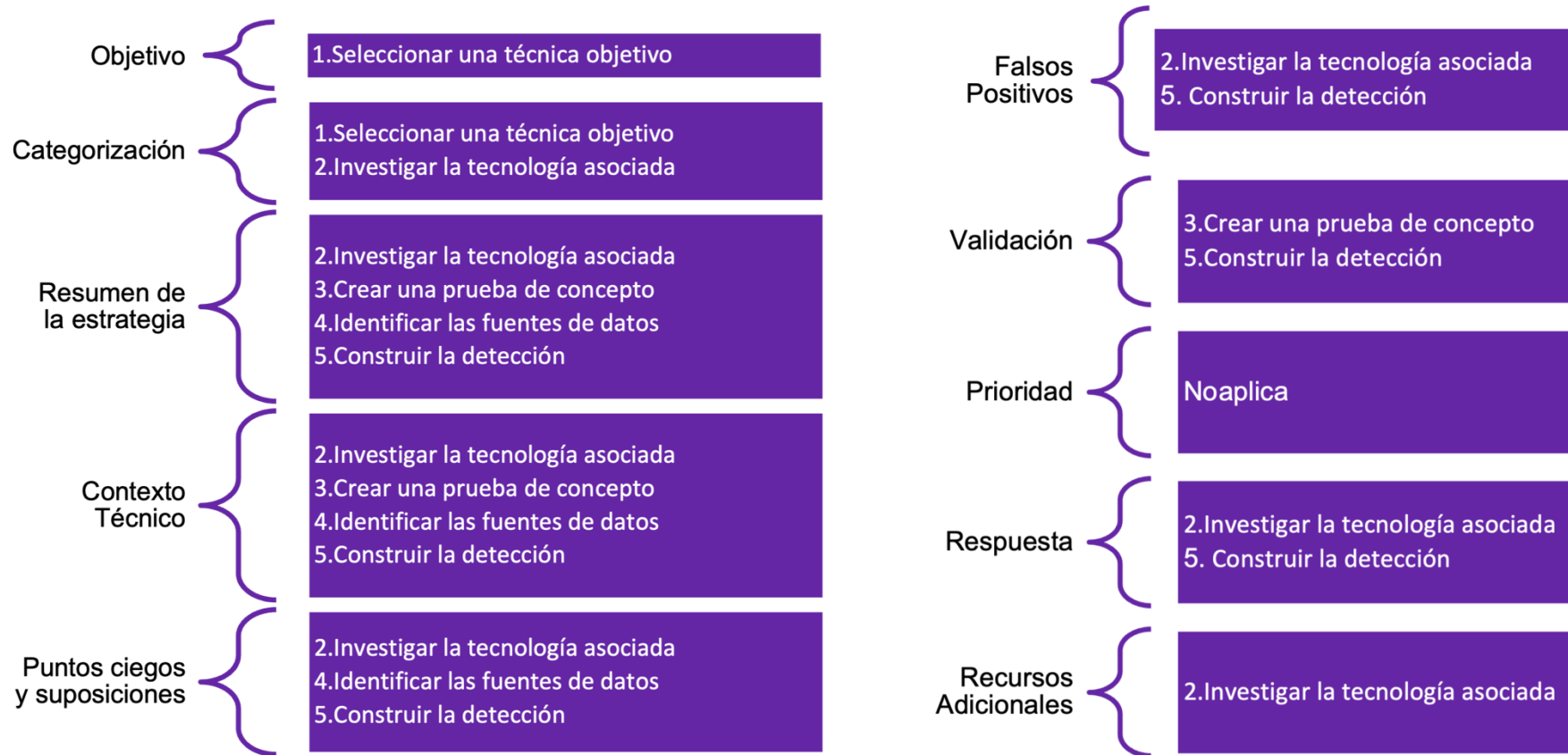
3  
Crear una prueba de concepto.

4  
Identificar las fuentes de datos.

5  
Construir la detección.

# Laboratorios (1.2)

## Alert and Detection Strategy - Ingeniería de Detección





# Laboratorios (1.2)

## Alert and Detection Strategy - Ingeniería de Detección

### Objetivos:

- Identificar y comprender las secciones que conforman el ADS.
- Generar un ADS a partir del trabajo realizado en cada etapa de la metodología de detección.

# Laboratorios (1.3)


¡Introducción a Jupyter Notebooks!




# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks


Aplicación open-source compatible con más de 40 lenguajes de programación como Python, R, PySpark, etc.



Input y output de sesiones interactivas, notas y paso a paso de metodologías o tareas específicas



Se guardan en documentos con formato JSON y extension .ipynb



Usos: Transformación de datos, modelado estadístico, simulaciones, visualización de datos, etc.

# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

Ejemplo: Jupyter Notebook para análisis exploratorio básico

## ➤ Conexión desde JN a Splunk



```
HOST = "Splunk IP"  
PORT = 8089  
USERNAME = "admin"  
PASSWORD = "changeme"
```

```
# Create a Service instance and log in  
service = client.connect(  
    host=HOST,  
    port=PORT,  
    username=USERNAME,  
    password=PASSWORD)
```



# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

Capturar campos específicos - todos los eventos



```
searchquery_oneshot = "search index=win EventCode=4688 | fields Account* process* parent*
```

```
oneshotsearch_results = service.jobs.oneshot(searchquery_oneshot, **kwargs_oneshot)
```

Limitar por tiempo



```
kwargs_oneshot = {"earliest_time": "2024-03-12T12:00:00.000-07:00",  
                  "latest_time": "2024-03-14T12:00:00.000-07:00",  
                  "output_mode": 'json'}
```

# Laboratorios (1.3)

## Tradecraft | Jupyter Notebooks

Filtrar por las columnas de interés

process	process_exec	process_id	process_name	process_path	parent_process	parent_process_id	parent_process_name	parent_process_path	process_command_line_arguments
(Program der\bin...	splunk-powershell.exe	0xce8	splunk-powershell.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x9a4	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-netmon.exe	0x139c	splunk-netmon.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x9a4	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-regmon.exe	0x884	splunk-regmon.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x9a4	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-powershell.exe	0x208	splunk-powershell.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x9a4	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	--ps
(Program der\bin...	splunk-MonitorNoHandle.exe	0x69c	splunk-MonitorNoHandle.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x9a4	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-powershell.exe	0x12bc	splunk-powershell.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x75c	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-netmon.exe	0x1730	splunk-netmon.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x75c	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-regmon.exe	0x12e4	splunk-regmon.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x75c	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na
(Program der\bin...	splunk-powershell.exe	0x1238	splunk-powershell.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x75c	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	--ps
(Program der\bin...	splunk-MonitorNoHandle.exe	0x15ec	splunk-MonitorNoHandle.exe	C:\Program Files\SplunkUniversalForwarder\bin...	C:\Program Files\SplunkUniversalForwarder\bin...	0x75c	splunkd.exe	C:\Program Files\SplunkUniversalForwarder\bin...	Na



# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

Listando columnas de interés

```
# Modify the query to your needs
searchquery_onehot = "search index=win EventCode=4688 | fields* | fields - _* " # <--- This One!
oneshotsearch_results = service.jobs.onehot(searchquery_onehot, **kwargs_onehot)

# Get the results and display them using the JSONResultsReader
reader = results.JSONResultsReader(oneshotsearch_results)
results = []
for result in reader:
    results.append(result)
df = pd.DataFrame(results)
for col in df.columns:
    print(col)
```

✓ 24.7s

```
Caller_Domain
Caller_User_Name
Channel
CommandLine
Computer
Error_Code
EventCode
EventData_Xml
EventID
EventRecordID
Guid
Keywords
Level
Logon_ID
MandatoryLabel
Name
NewProcessId
NewProcessName
Opcode
ParentProcessName
ProcessID
ProcessId
Process_Command_Line
```

# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

Query para buscar los eventos generados por el canal de Sysmon



```
searchquery_oneshot = "search index=win EventChannel=Microsoft-Windows-Sysmon/Operational | fields EventCode | fields - _*"
```



# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

Podemos saber el total de resultados o 'hits' y la información en tabla

```
# Get the results and display them using the JSONResultsReader
reader = results.JSONResultsReader(oneshotsearch_results)
results = []
for result in reader:
    results.append(result)
df = pd.DataFrame(results)
print(df['EventCode'].value_counts())
```

✓ 8.4s

EventCode	
1	59581
13	2019
11	975
22	396
3	83
18	12
17	12
6	9
12	9
5	7
255	6
4	3
8	1

Name: count, dtype: int64

# Laboratorios (1.3)

Tradecraft | Jupyter Notebooks

## Objetivo:

- Explorar las capacidades de búsqueda en Splunk a través de Jupyter Notebooks

## Extra:

- Genera un ADS haciendo uso de Jupyter Notebooks

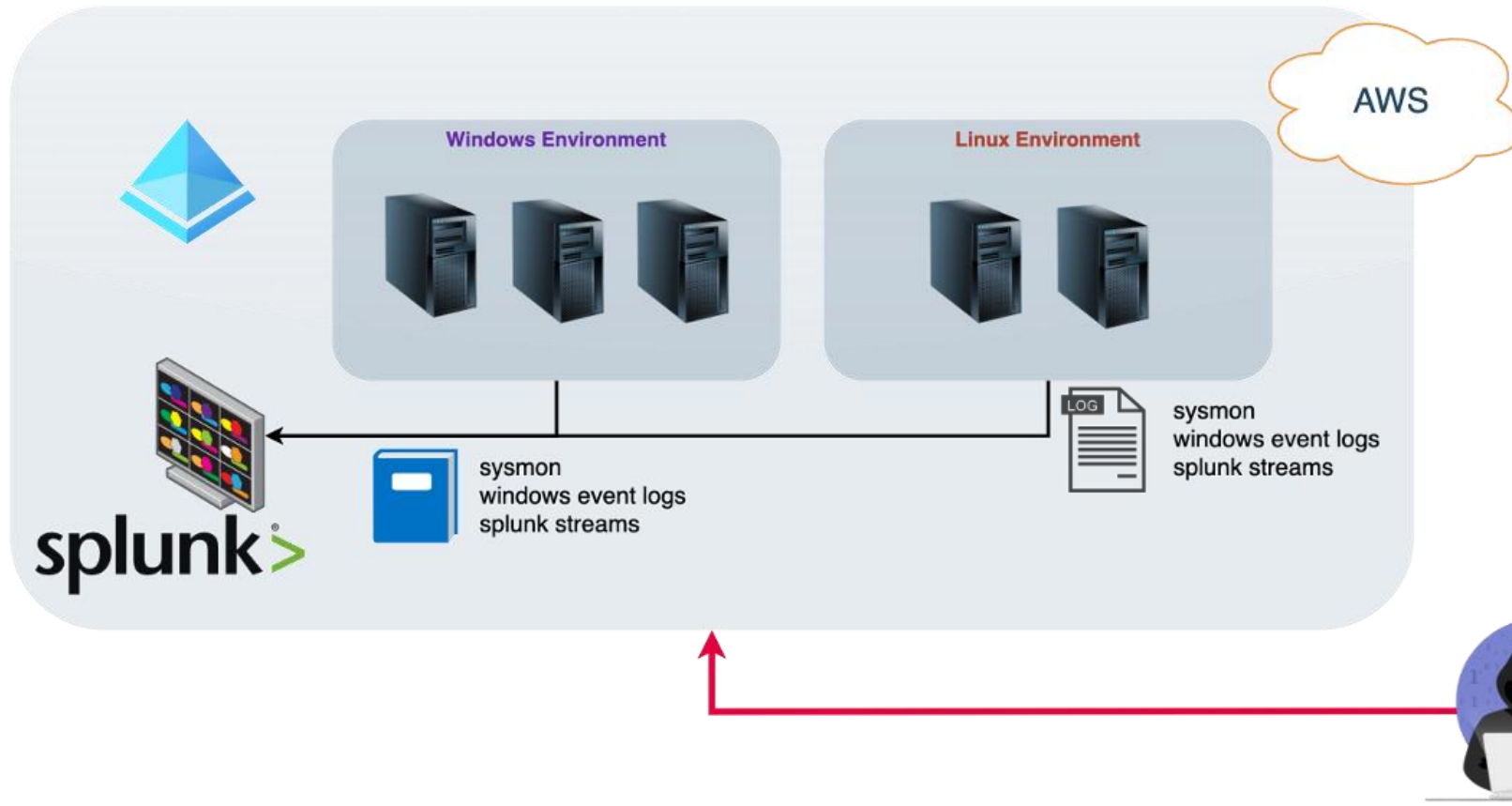


# Escenario de Kill Chain

Attack Range by Splunk

# Cyber Kill Chain Hunting

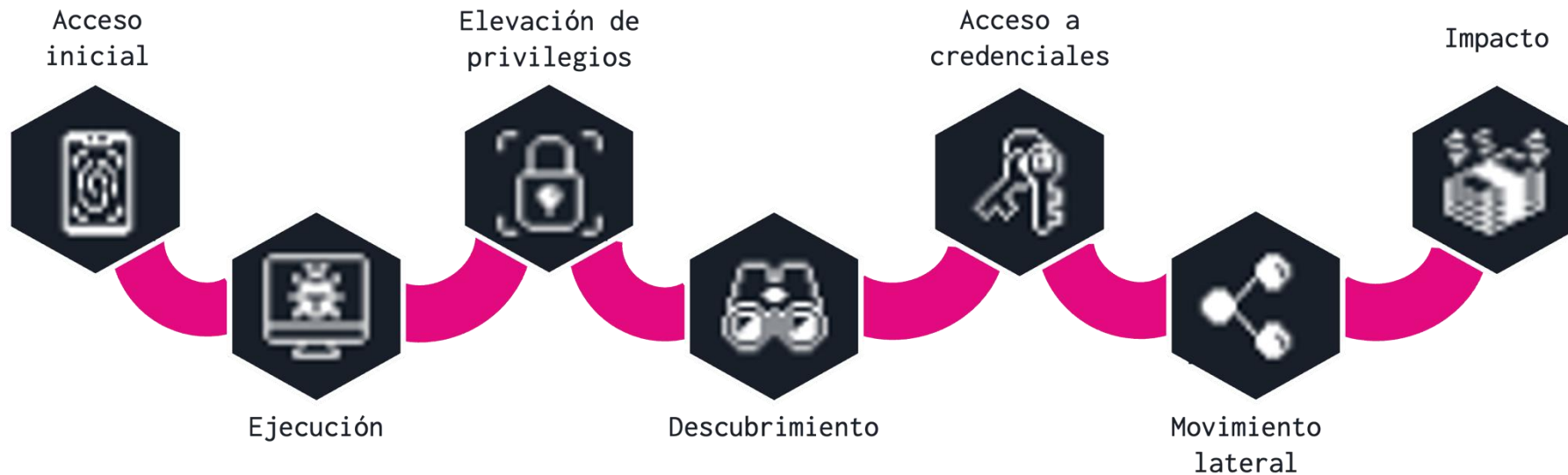
## Attack Range by Splunk





# Cyber Kill Chain Hunting

Attack Range by Splunk



# Cyber Kill Chain Hunting

Let's us hunt!

Hacer ingeniería de detecciones para:

- Identificar la TTP utilizada en cada fase del kill chain propuesto
- Crear un playbook de detecciones para la amenaza



# Conclusiones

¿Qué aprendimos?



# Conclusiones

Dentro de la metodología de Threat Hunting e Ingeniería de Detección; es importante:

- Conocer la importancia de contar con un entendimiento profundo del tradecraft empleado por los atacantes para detectar el uso de TTPs dentro del ambiente tecnológico que estemos defendiendo.
- Lograr familiaridad con conceptos clave para operacionalizar inteligencia provista por el Mitre ATT&CK. Esto permite lograr un mayor entendimiento de como puede ser usada para madurar nuestras capacidades y estrategias de detección y respuesta.



# Conclusiones

Ingeniería de Detección y Threat Hunting son dos disciplinas íntimamente relacionadas, conocer como influyen entre si nos ayuda a ser más efectivos en nuestros esfuerzos de identificación proactiva de amenazas. Threat Hunting e Ingeniería de Detección no se limitan al uso de herramientas, sino que emplean procesos iterativos, metódicos y analíticos.

Mejorar el tradecraft del Blue Team permite cubrir esos “blind spots” que las amenazas buscan en las herramientas de detección.

Threat Hunting e Ingeniería de Detección no se limitan al uso de herramientas, sino que emplean procesos iterativos, metódicos y analíticos.

Mejorar el tradecraft de del Blue Team permite cubrir esos “blind spots” que las amenazas buscan en las herramientas de detección.



# BUGCON



## ¿Preguntas?





¡Gracias!

@0ldbl4ck | @\_zkvL

