# ATTPwn: Adversary Emulation ATT&CK Automation Tool

*Pablo González (pablo.gonzalezperez@telefonica.com)*
*Francisco Ramírez (franciscojose.ramirezvicente@telefonica.com)*
*Victor Rodriguez (victor.rodriguez.practicas@telefonica.com)*

## Executive Summary

ATTPwn es una herramienta diseñada para la emulación de adversarios. El objetivo de la herramienta es acercar la emulación de una amenaza real con las implementaciones basadas en las técnicas y las tácticas del marco de trabajo de MITRE ATT&CK. La idea es emular cómo una amenaza opera en un escenario de intrusión, en el que la amenaza ha tenido éxito. La herramienta está orientada a sistemas Microsoft Windows a través del uso de la línea de comandos Powershell. Gracias a ésta se implementan las diferentes técnicas basadas en MITRE ATT&CK. La herramienta está destinada para que se pueda llevar a cabo la emulación de adversarios como para de un ejercicio de Red Team y poder verificar la eficacia y eficiencia de los controles en la organización ante una amenaza real.

# 1. Introducción

La emulación de adversarios ha tomado una gran relevancia en el mundo del Red Team. Es un ejercicio que aporta una visión sobre las posibilidades de que una amenaza real, parecida a otras muchas que tanto las empresas como la sociedad han sufrido, pueda afectar a una organización. El objetivo es poder comprobar si los controles que están implantadas en la organización son eficientes y eficaces, detectando la amenaza o mostrando debilidad ante ella.

Los objetivos de un ejercicio de Red Team son los que se pueden enumerar a continuación:

- Demostración del nivel de exposición y riesgo.
- Demostración del impacto de negocio.
- Demostración de las capacidades de prevención.
- Demostración de las capacidades de detección.
- Demostración de las capacidades de reacción o respuesta ante incidentes.

MITRE ATT&CK [1] es una base de conocimiento de acceso global de tácticas y técnicas adversarias basadas en observaciones del mundo real. La base de conocimientos de ATT&CK se utiliza como base para el desarrollo de modelos y metodologías de amenazas específicas en el sector privado, en el gobierno y en la comunidad de productos y servicios de ciberseguridad.

El marco de trabajo MITRE ATT&CK asume la brecha de seguridad, por lo que el punto de comienzo es la táctica de intrusión inicial. Cualquier actividad que se haya realizado previamente estará cubierta por un marco de trabajo llamado PRE-ATT&CK.

Las tácticas son utilizadas para la descripción de los diferentes pasos de un ataque a alto nivel. Estos pasos son utilizados por un adversario.

Las técncias describen cómo se ejecuta una determinada táctica. En otras palabras, una táctica puede ser implementada o llevada a cabo por una variedad de técnicas. El marco de trabajo MITRE ATT&CK incluye una descripción, recomendaciones de detección y mitigación y amenazas conocidas que utilizan la técnica. En la siguiente dirección URL puede conocerse mayor detalle sobre el marco de trabajo: https://mitre-attack.github.io/attack-navigator/enterprise/# [2]

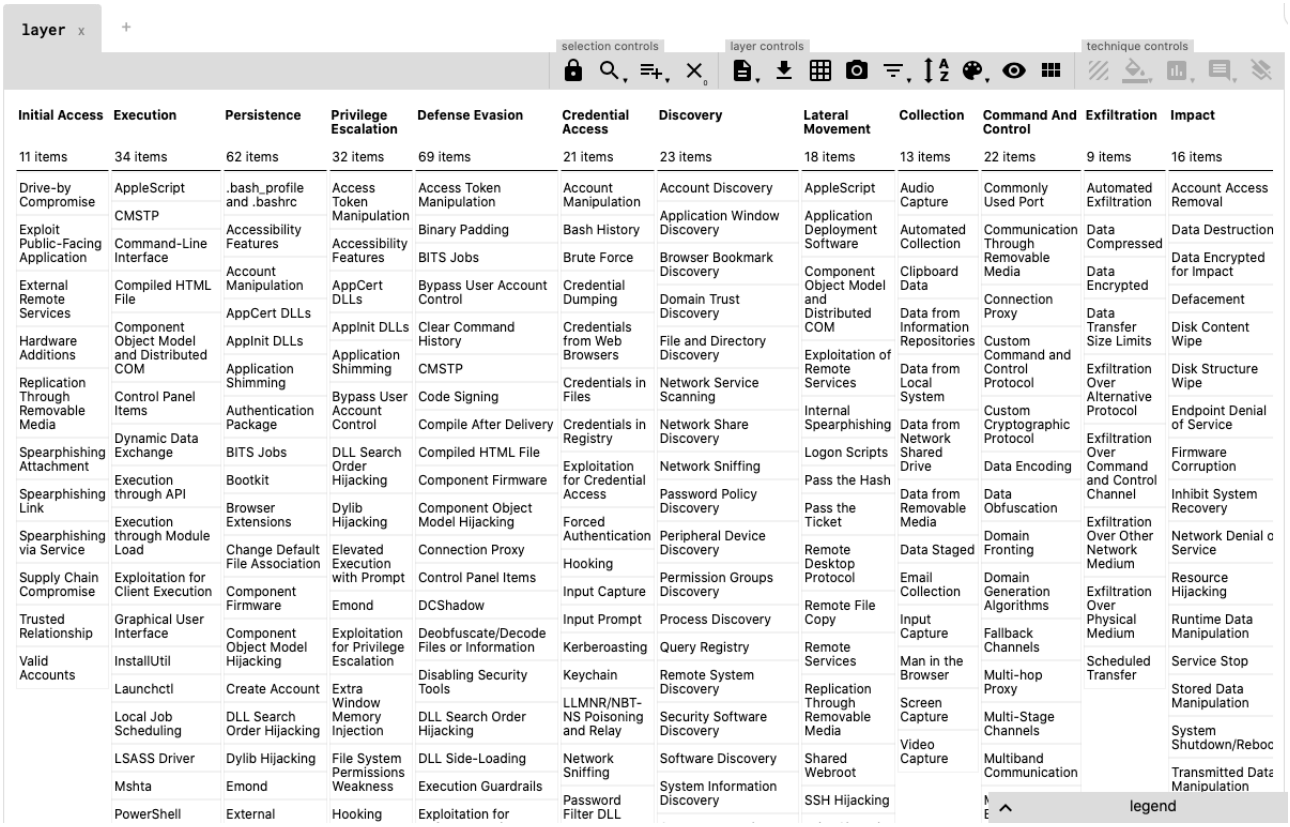



*Figura 1: Tácticas y Técnicas del marco de trabajo ATT&CK*

## 1.1. Plan de emulación (Adversary Emulation Plan)

Con el objetivo de que personal ofensivo y defensivo pudiera poner en marcha el marco de trabajo ATT&CK desde un punto de enfoque práctico, el MITRE creó lo que se denomina el Plan de Emulación Adversaria [3].

Hay dos vertientes bien diferencias en el plan de emulación adversaria. La primera es que el personal dedicado a la parte defensiva o Blue Team pueda probar de forma eficaz y eficiente el nivel de defensa de las redes de la organización. La segunda es que el personal dedicado a la parte ofensiva o Red Team pueda definir y modelar el comportamiento de un amenaza o adversario, tal y como se describe el marco de trabajo ATT&CK.

La idea es sencilla. Se conocen datos sobre inteligencia de amenazas, pero no hay un gran detalle sobre cómo los atacantes están encadenando las técnicas en los ataques. El marco de trabajo ATT&CK es innovador en este hecho, proporcionando la visión de cómo se encadenan técnicas sobre una base de datos de amenazas reales.

El plan de emulación de adversarios proporciona la información sobre las técnicas que una amenaza encadena para llevarse a cabo. En otras palabras, cuando se utiliza una emulación basada en ATT&CK lo que se está llevando es una agrupación a los TTPs (Tácticas, Técnicas y Procedimientos) por parte de los operadores de la emulación sobre el escenario de la red de la organización.

## 2. Trabajos anteriores

Existen tres trabajos previos que son la base de la implementación de técnicas de ATTPwn en la actualidad. El primer trabajo se donominó *"Give me a PowerShell and i will move your world"* realizado entre finales de 2014 y mayo de 2015, dónde se presentó en *Qurtuba Security Congress* [4]. La idea surge de la no existencia de herramientas de *pentesting* en el equipo o la no posibilidad de instalar este tipo de herramientas en un equipo. Gracias a la existencia y no prohibición del uso de *PowerShell* en el equipo se disponía de un script cuyo objetivo era hacer un *bypass* de la política de ejecución en *Windows* y lograr ejecutar scripts de *PowerShell* en el ámbito del *pentesting*.

Las funciones eran cargadas desde ficheros de disco, por lo que existía el riesgo de que un antivirus pudiera detectar por firma, de manera sencilla, el script como una amenaza. El script principal podría cargar las funciones y ejecutar las instrucciones a través de *Twitter* y mensajes directos. Es decir, se podía utilizar un *Covert Channel*.

El segundo trabajo se denomina *"PSBoT: No tools, but not problem!"* y se presentó en septiembre de 2016 en el evento *Rooted CON Valencia* [5]. Este segundo trabajo era una evolución del anterior, partiendo de nuevo de la hipótesis de que el *pentester* no disponía de la posibilidad de ejecutar herramientas o de instalaras. Esta evolución cargaba funciones dinámicamente a memoria, sin que éstas estuvieran en disco. Este mecanismo se denomina *Fileless*. Además, el bot permitía la ejecución a través de mecanismos de explotación, por lo que se podía aprovechar de un exploit para ser el código ejecutado. El control del bot se realizaba a través de un panel escrito en *PowerShell* a modo de línea de comandos. Las funciones se obtenían desde un servidor externo configurado por el usuario.

El tercer trabajo denominado iBombShell [6] fue presentado en BlackHat Europe 2018. La herramienta permite disponer de una shell dinámica de pentesting y mediante el manejo de dos modos de ejecución proporciona al pentester la posibilidad de llevar a cabo diferentes acciones de explotación y post-explotación.

## 3. Powershell

*PowerShell* aparece con la liberación de *Windows Vista* por parte de *Microsoft*. Viene de forma nativa con el sistema operativo, hecho que hace que sea de mucho interés, tanto para administradores IT como para *pentesters*. En su versión 1.0, *PowerShell* era compatible con *Windows XP*. A continuación, se puede visualizar la aparición de las diferentes versiones de *PowerShell*. Cada nueva versión incluía un gran número de funcionalidades y módulos que ayudaban a integrarse más y más con el sistema operativo y las diferentes herramientas de éste.

- *Monad Manifest*. Esto fue el comienzo de la idea de la *PowerShell*. Publicada por *Jeff Snover* en el año 2002 [1].
- La versión 1.0 apareció en el año 2006. La primera versión estable.
- La versión 2.0 apareció en el año 2009 con la liberación de *Windows 7*.
- La versión 3.0 apareció en el año 2012 con la liberación de *Windows 8*.
- La versión 4.0 apareció en el año 2013.
- La versión 5.0 apareció en el año 2016.
- La versión 6.0 apareció en el año 2017. Esta versión marca un hito ya que se libera la versión para *GNU/Linux* y *macOS*.
- La versión 7.0 apareció en el año 2020.


*Figura 2: Línea temporal de versiones de Powershell*

## 3.1. PowerShell EverySystem

La irrupción de *PowerShell* en otras plataformas marcó un hito para el uso de esta línea de comandos. El proyecto fue llamado por *Microsoft* como *"PowerShell for Every System"* [10]. La pieza fundamental es *PowerShell Core*, el cual permite ejecutar entre plataformas, en este caso, *Windows, Linux* y *macOS*.

El proyecto está optimizado para trabajar de la forma más eficiente con estructuras de datos como JSON, CSV, XML, etcétera. Además, el uso de objetos y de las REST API hacen de *PowerShell* una herramienta integradora de tecnologías comunes a las plataformas.

La aparición del proyecto *"PowerShell for Every System"* hace que la fase de post-explotación en diferentes plataformas se acerque y pueda ser unificada, flexibilizada y homogeneizada. Es un hecho que ha provocado que muchos usuarios prueben *PowerShell* en sistemas no *Microsoft*.

Existe una particularidad y es que la gran ventaja del uso de *PowerShell* en el ámbito del *pentesting* en sistemas *Microsoft* sigue siendo que la aparición de la línea de comandos en el sistema es nativa, mientras que en el caso de sistemas *GNU/Linux* o *macOS* se debe instalar previamente. Esto es un *hándicap*, pero no cabe duda de que es un paso a la posibilidad de aprovechar y homogeneizar procesos de post-explotación de sistemas a través de una herramienta.

# 4. ATTPwn: Emulación de adversarios

La idea detrás de ATTPwn es unir el marco de trabajo MITRE ATT&CK con técnicas implementadas a través de la línea de comandos de Microsoft Windows Powershell. La gran cantidad de técnicas implementadas con Powershell apoyan a que se puedan emular un porcentaje alto de técnicas indicadas en la matriz ATT&CK.

El proyecto es colaborativo, es decir, un usuario puede tener su base de conocimeinto inicial basado en ATT&CK, pero puede importar nuevas implementaciones de técnicas utlizando Powershell y relancionarlas con el identificador de las técnicas y de las tácticas.

Se ha buscado mediante de archivos con fomato JSON poder exportar ese nuevo conocimiento e importarlo en otros entornos dónde ATTPwn esté implementado. De esta forma el conocimiento colaborativo cobra gran importancia. Esto facilita que varios usuarios puedan compartir el conociminto entre diferentes entornos. Las técnicas son elementos dinámicos que van apareciendo con la evolución de la seguridad ofensiva.

## 4.1. Arquitectura

En este apartado se presenta la arquitectura global de ATTPwn. La arquitectura consta de tres elementos fundamentales:

- Consola. La consola es el código escrito en Powershell que se encargará de implementar los agentes, denominados 'Warriors', en las diferentes máquinas cuando la amenaza sea emulada. La consola se conectará al controlador de ATTPwn para solicitar las técnicas a ejecutar según su amenaza asignada.
- Funciones. En esta parte se almacenan las técnicas implementadas en código Powershell. Todas las técnicas van ligadas con el identificador de técnica y táctica perteneciente al marco de trabajo ATT&CK.
- MVC. Es la parte del modelo-vista-controlador. ATTPwn dispone de una aplicación web que hace uso de controladores que gestionan las diferentes peticiones y respuestas de las consolas desplegadas en una red. Además, ATTPwn hace uso de modelos para interactuar con la base de datos correspondiente, dónde se gestionan las amenazas asignadas, los resultados obtenidos y las relaciones entre el marco de trabajo MITRE ATT&CK y las técnicas implementadas. Por último, la aplicación dispone de diferentes vistas para el manejo sencillo de la emulación por parte del usuario.

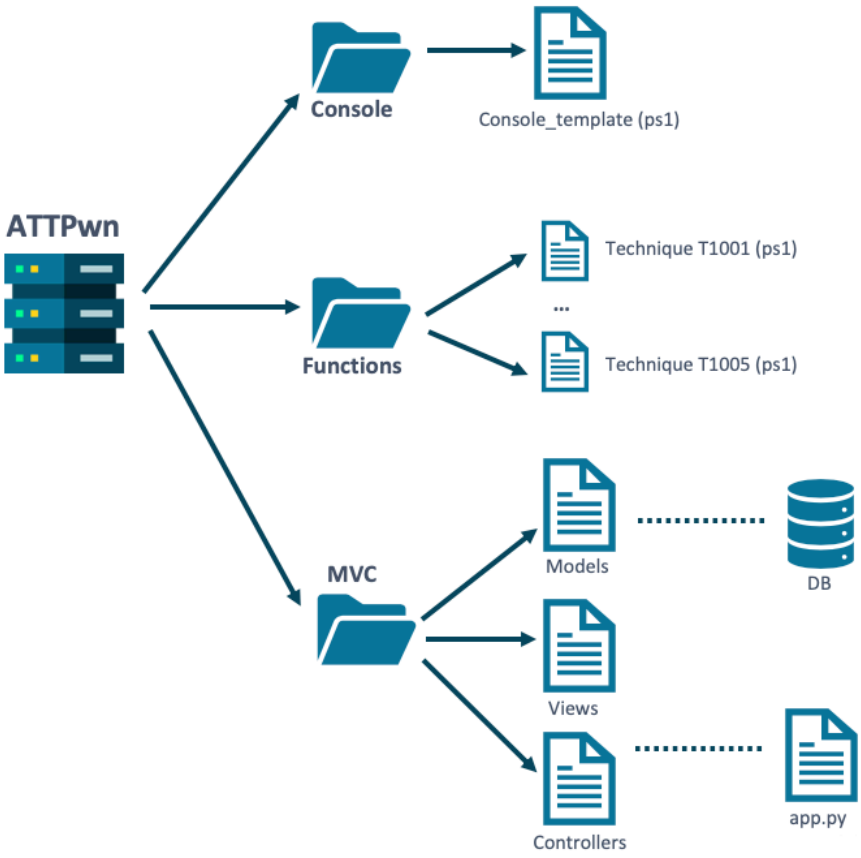



*Figura 3: Arquitectura ATTPwn*

Un esquema básico de la arquitectura y a grandes rasgos tendría dos elementos fundamentales desde el punto de vista de red:

- Agente, denominado 'Warrior'. Emulación del adversario. Código escrito en Powershell. Lo representaremos con este icono:

*Figura 4: Representación de warrior de ATTPwn*

- Nodo raíz o *Command and Control*. Desde esta aplicación web se puede gestionar la aplicación de amenazas para ser emuladas por los diferentes Warriors desplegados. Además, se puede gestionar los resultados de la emulación y sacar conclusiones sobre los controles. Todo alineado con los identificadores que marca la matriz ATT&CK. Lo representaremos con este icono:

*Figura 5: Representación del nodo raíz de ATTPwn*

Desde el punto de vista de red el despliegue de Warriors en la red se hará a través de diferentes posibilidades: invocación remota a través de priviliegios en red, invocación en máquinas remotas sin privilegios, invocación local, etcétera. Se recomienda no incluir a más de 10-15 máquinas en este proceso, tal y como indica ATT&CK. Estos ejercicios son emulaciones, pero hay que tener un entorno controlado y lo más real posible.

*Figura 6: Warriors conectándose al nodo raíz de ATTPwn*

## 4.2. Flujos de comunicación

A continuación se muestran los flujos de comunicación entre los Warriors o agentes desplegados en las máquinas Windows y el nodo raíz de ATTPwn.

El primer flujo es el de conexión.

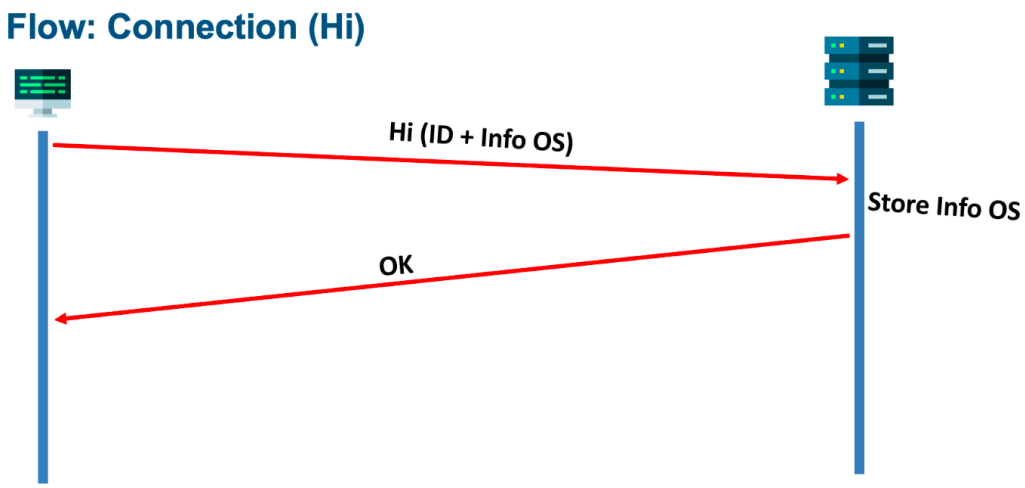



*Figura 7: Flujo de conexión de un warrior*

En la generación de un warrior se indica la dirección IP a la que éste se debe conectar. Esto se verá más adelante. Cuando el warrior quiere registrarse como 'máquina comprometida' envía una petición por el método POST de HTTP al nodo raíz indicando el ID Warrior que tiene y la información recopilada de la máquina.

Cuando esta información es recibida por el nodo raíz se almacenan los datos en la base de datos y se pone a disposición del usuario para que pueda asignar el plan definido en una amenaza y llevar a cabo su ejecución.

El segundo flujo es el de planificación y despliegue. Este flujo constará de N pasos, en función del número de tareas de las que disponga una amenaza.

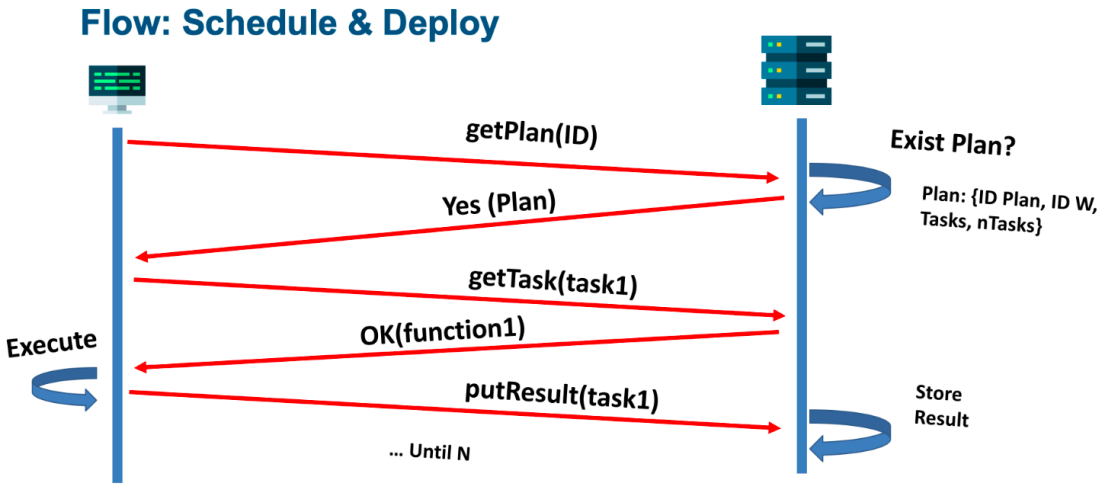



*Figura 8: Flujo de planificación y despliegue*

Cuando el warrior se ha registrado solicita un nuevo plan de amenaza a través de la función getPlan(). Si el usuario ha asignado algún plan al warrior se responde de forma afirmativa y se entrega un plan de la amenaza. Este plan está generado en formato JSON. Se dará mayor detalle más adelante.

Una vez que se dispone del plan de la amenaza en el warrior, éste llevará a cabo la solicitud de cada función o tarea a ejecutar. Esto se realiza a través de la función getTask(). Se ejecutará un getTask() por cada implementación de técnica que el warrior necesite ejecutar para cumplir con el plan de la amenaza.

Por cada ejecución anterior, se devolverá a través de la función putResult() dos cosas:

- Si la ejecución de la función que implementa la técnica de ATT&CK ha sido ejecutada correctamente o no. Aquí entra en juego los diferentes controles y defensas que la organización pueda tener implantados.
- En segunda instancia, se devuelve la salida de la ejecución de la función que implementa la técnica. Esto es para que un analista técnico pueda comprobar el resultado de dicha ejecución.

El tercer flujo sería el cierre de la ejecución del plan de la amenaza.

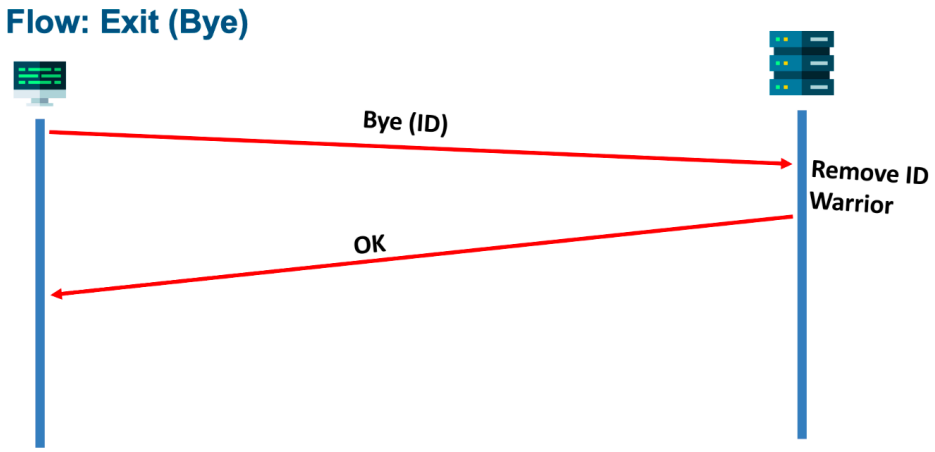



*Figura 9: Flujo de cierre de warrior*

Cuando el plan de amenaza, es decir, todas las funciones que implementan las técnicas propuestas son ejecutadas se envía una petición por POST denominada bye(). De esta forma el nodo raíz puede registrar que el warrior ha finalizado.

El nodo raíz también dará por "muerto" al warrior si durante un largo período de tiempo no indica actividad. El warrior indicará actividad si antes de tener asociado un plan de amenaza, el warrior se encuentra solicitando un plan.

## 4.3. Consola

La consola será generada desde la aplicación web indicando para ello la dirección IP desde dónde se descargará el fichero consola. Existe un fichero denominado consola_template el cual alberga las instrucciones que harán la gestión de la conexión, del plan de amenazas y del cierre de la conexión comentado en el apartado anterior. Cuando se ejecute la instrucción proporcionada en una máquina Windows se ejecutará el código de Powershell necesario para llevar a cabo la instanciación del warrior.

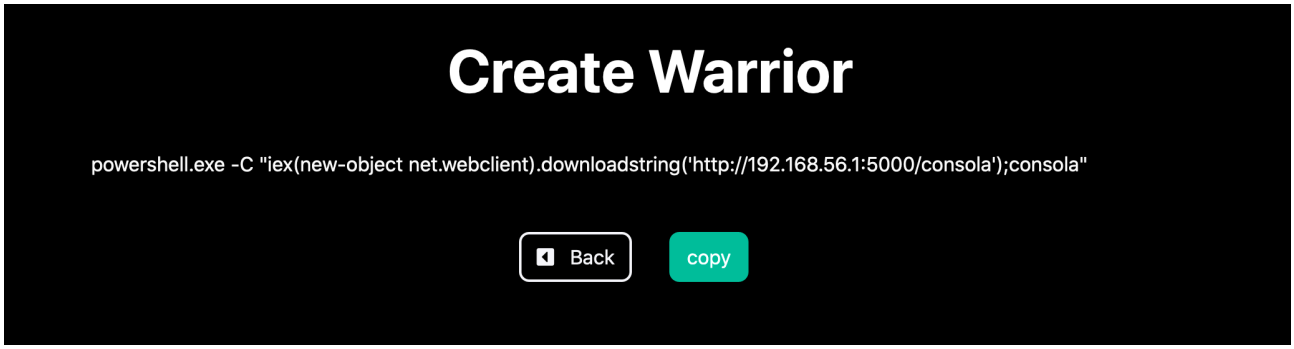



*Figura 10: Generación de warrior*

La consola dispone de dos partes con un gran "if" central:

- La primera parte o el camino básico de inicio es el comienzo de una amenaza. Como se ha visto en el apartado anterior, se realizará una petición de conexión. Antes se crea un ID único para esta instancia de forma que pueda ser identificado en el nodo raíz. Además, se recopila información del sistema operativo donde se está ejecutando.

```
$global:id = -join ((65..90) + (48..57) + (97..122)|Get-Random -Count 5 |% {[char]$_})
$global:id = "P"+$global:id

$nameOS = (Get-WmiObject Win32_OperatingSystem).Name.Split("|")[0]
$archOS = (Get-WmiObject Win32_OperatingSystem).OSArchitecture
$machine = [System.Net.Dns]::GetHostByName(($env:computerName)).HostName
$hi = @{id=$global:id;name=$nameOS;arch=$archOS;machine=$machine;pid=$PID}

$resp = invoke-webrequest -UseBasicParsing -Method POST -Body $hi "http://$IP:5000/hi"
```

*Figura 11: Generación ID Warrior, recopilación información y conexión*

Después el código de la consola solicitará una petición para conocer si hay algún plan para ejecutar. En caso de que exista plan éste será solicitado. La consola entra en un bucle para ir pidiendo cada función del plan de la amenaza. Cuando esto termina la consola ejecutará un "bye" o desconexión.

- La segunda part viene determinada por un caso que puede ocurrir en muchas emulaciones de amenazas. Se ha denominado "desdoblamiento" al proceso en el que mediante una escalada de privilegio o tras un movimiento lateral a otra máquina se genera un nuevo proceso con priviliegio o con posibilidad de ejecución en otro entorno. En este caso, la consola será invocada con una herencia de ID, es decir, no tendrá que generar el ID Warrior que identifica al warrior ante el nodo raíz. En otras palabras, si hay un plan con 3 tareas y la segunda tarea es una escalada de privilegios, y ésta tiene éxito, entonces la tercera tarea será ejecutada en otro proceso a nivel de sistema operativo, pero desde el punto de vista lógico de ATTPwn es el mismo warrior, es decir, hereda el mismo ID Warrior.

```powershell
function consola
{
    param(
    [String] $id
    )

    $global:id = ""
    $global:remoteip = "$IP"
    if($id)
    {
        sleep 5
        $global:id = $id
        #Spawn Process (same ID warrior)
        #Trace ON
        $nameOS = (Get-WmiObject Win32_OperatingSystem).Name.Split("|")[0]
        $archOS = (Get-WmiObject Win32_OperatingSystem).OSArchitecture
        $machine = [System.Net.Dns]::GetHostByName(($env:computerName)).HostName
        $hi = @{id=$global:id;name=$nameOS;arch=$archOS;machine=$machine;pid=$PID}

        $resp = invoke-webrequest -UseBasicParsing -Method POST -Body $hi "http://$IP:5000/hi2"
```

*Figura 12: Desdoblamiento de proceso con herencia de ID Warrior*

Hay una parte central de la consola cuando se empieza a ejecutar funciones que implementan técnicas de ATT&CK de la que se hablará más adelante. La consola ejecuta las funciones y espera un resultado en forma de *'hashtable'* dónde se le indique si la función de ATT&CK correspondiente se ha ejecutado de forma satisfactoria o no. Además, se obtiene el resultado de la ejecución.

```powershell
#request function to givemetask
$function = invoke-webrequest -UseBasicParsing -method POST -body @{id=$global:id;plan=$idplan;
    funcion=$task.function} "http://$IP:5000/givemetask" | ConvertFrom-JSON
$id = $function.id
$plan = $function.plan
$funcion = $function.funcion
$execute = $function.funcion | iex

$success = 1
if($execute.success)
{
    $success = 0
}
$exec = invoke-webrequest -UseBasicParsing -method POST -body @{id=$global:id;plan=$idplan;
    idfunction=$task.idfunction;funcion=$task.function;resultado=$execute.results;good=$success} "http://$IP:5000/putresult"
```

*Figura 13: Ejecución de la función y llamada a putResult()*

Más adelante se hablará del formato de las funciones que serán invocadas por la consola. Se le ha llamado *'Skeleton function'*, ya que se proporciona una guía para que otros usuarios puedan desarrollar sus funciones y las incorporen a ATTPwn.

## 4.4. Formato del plan de amenaza

El plan de amenaza se genera en el nodo raíz en función de las necesidades del usuario. El usuario puede utilizar amenazas reales ya generadas o crear sus propias amenazas y ver los resultados de sus técnicas de control. El plan de amenaza se proporciona a la consola a través de la llamada giveMePlan(), una vez que se ha validado que existe un plan de amenaza para dicho warrior.

El formato del documento entregado con el plan es JSON. Este documnto JSON tiene la siguiente estructura:

```
{
        plan : ID_Plan,
        id : ID_Warrior,
        tasks : {
                        t0: {
                        function : function_name,
                        idfunction : ID_Function,
                        die : [ 0 | 1 ]
                        },
                        ...
                        tN: {
                                function : function_name,
                                idfunction : ID_Function,
                                die : [ 0 | 1 ]

                        }
                }
        ntasks : Tasks_Number
}
```

En este documento hay que tener en cuenta que:

- El parámetro plan indica el identificador interno del plan de la amenaza.
- El parámetro id indica el identifiador unívoco del warrior.
- El parámetro tasts es un diccionario con varias técnicas a ejecutar. Cada técnica tiene:
    - El nombre de la función que se solicitará para su descarga.
    - El identificador de la función.
    - El parámetro ‘die’ que indica si la función debe desdoblarse o no.

## 4.5. Diseño de base de datos

En este apartado se muestra el diseño de la base de datos. La base de datos es un elemento central ya que permite que los resultados persistan y que se pueda gestionar de manera sencilla. El fichero que alberga la base de datos de ATTPwn se denomina 'mydatabase.db'. Está consituito en SQLite.

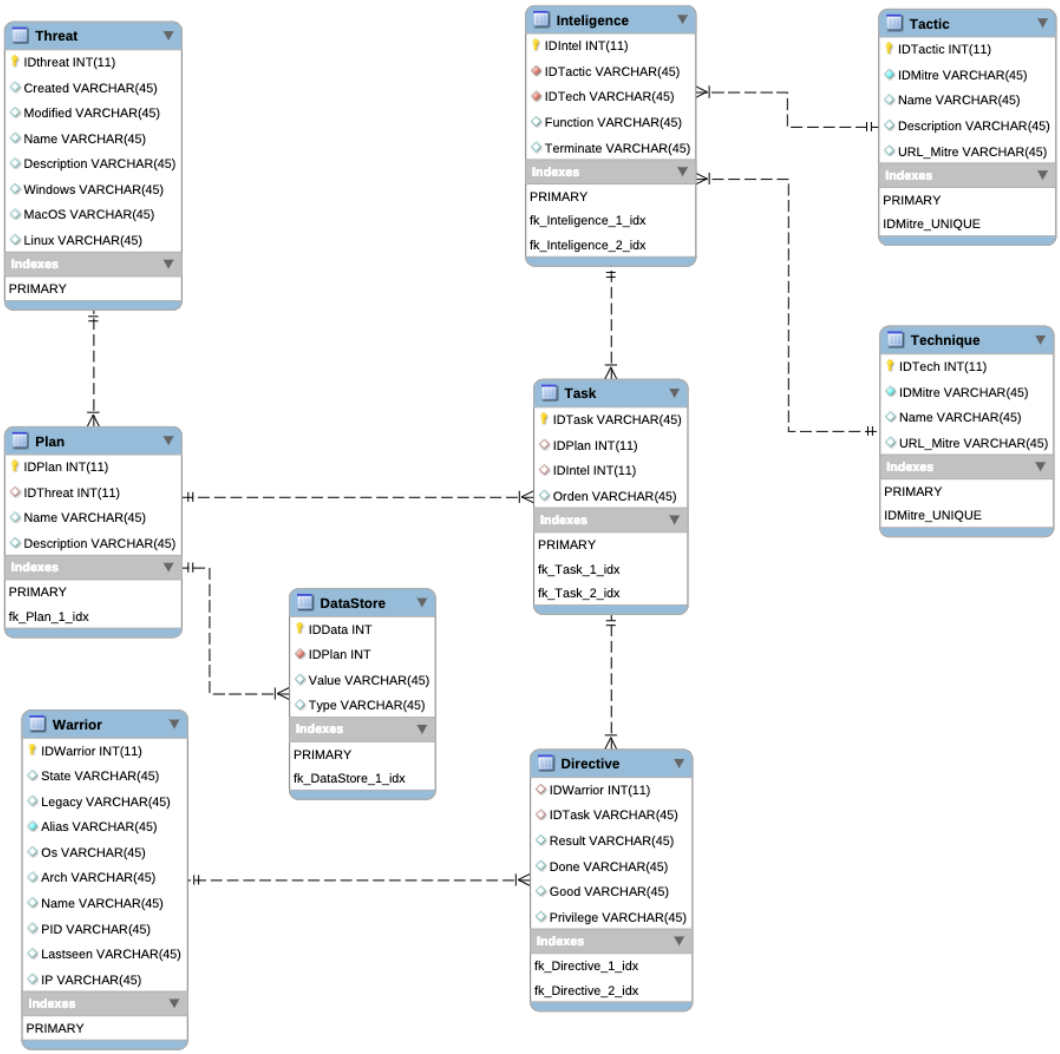



*Figura 14: Esquema de base de datos de ATTPwn*

## 4.6. Crea tu propia función para ATTPwn: 'Skeleton function'

Cualquier usuario puede crear o añadir funciones que implementen técnicas del marco de trabajo MITRE ATT&CK. En este apartado se muestra el esqueleto de una función que puede ser añadida a la herramienta.

El esqueleto de la función a implementar es el siguiente:

```
Function name_technique_implemented
{
                    #instructions
```

```
                    …
                    #instructions

                    #return data
                    return   @{results=$dataDisplayResults;success=$executionSuccessOrNot;[data=$data
for Datastore ATTPwn}}
}

#Main

#First, if your script need data…
#getData()
#give me data (1 or more types)
$resp = invoke-webrequest -UseBasicParsing -Method POST -Body @{id=$global:id;[type_data]=""} $uri

#execution
$execute = invoke-function [params]

#Send data for Datastore. Others techniques can use this data
#putData()
invoke-webrequest -UseBasicParsing -Method POST -Body @{id=$global:id; [type_data]=$data} $uri

#end technique
return @{results=$dataDisplayResults;success=$goodOrNot }
```

Cada función implementada está asociada a una técnica y a una o varias tácticas del marco de trabajo. Los pasos que se pueden estudiar del esqueleto de función presentado son:

1. Se dispone una función que es la técnica que se quiere lanzar.
2. Se dispone de un #main que hará las veces de programa principal. Se encargará de controlar el flujo de la técnica que se quiere ejecutar. Se encarga de:
    a. Solicitar datos necesarios al nodo raíz para la ejecución de la técnica. Esto es un prerrequisito de ejecución.
    b. Controlar si la ejecución de la ténica ha sido satisfactoria o no. Se almacena en una variable booleana.
    c. Si hay datos obtenidos como, por ejemplo, un volcado de credenciales, el descubrimiento de direcciones IP o enumeración de usuarios que, posteriormente otra técnica puede utilizar son volcados a un Datastore que dispone el nodo raíz.

Este esquema puede visualizarse en cualquiera de las funciones que se pueden encontrar en ATTPwn.

## 4.7. Panel web

En este apartado se muestran las funcionalidades básicas para el manejo del nodo raíz de ATTPwn. Desde este nodo raíz se puede llevar a cabo el despliegue de las amenazas y las configuraciones de éstas. Es decir, se puede hacer la planificación del plan de amenaza y llevar a cabo las diferentes ejecuciones de la emulación del adversario.

### 4.7.1. Inicio

En la pantalla de inicio se puede observar una vista ejecutiva de qué Warriors están vivos o preparados para ejecutar una amenaza. Se puede ver un histórico de los Warriors que se han ido generando, así como todos los que se han ido creando, tanto los que se encuentran vivos como los que terminaron su ejecución.

Por el icono que muestra el warrior se puede saber el sistema operativo que ejecuta, ya que en el caso de ser ejecutado en un sistema Windows 10, éste mostrará el icono más representativo de la versión del sistema operativo.

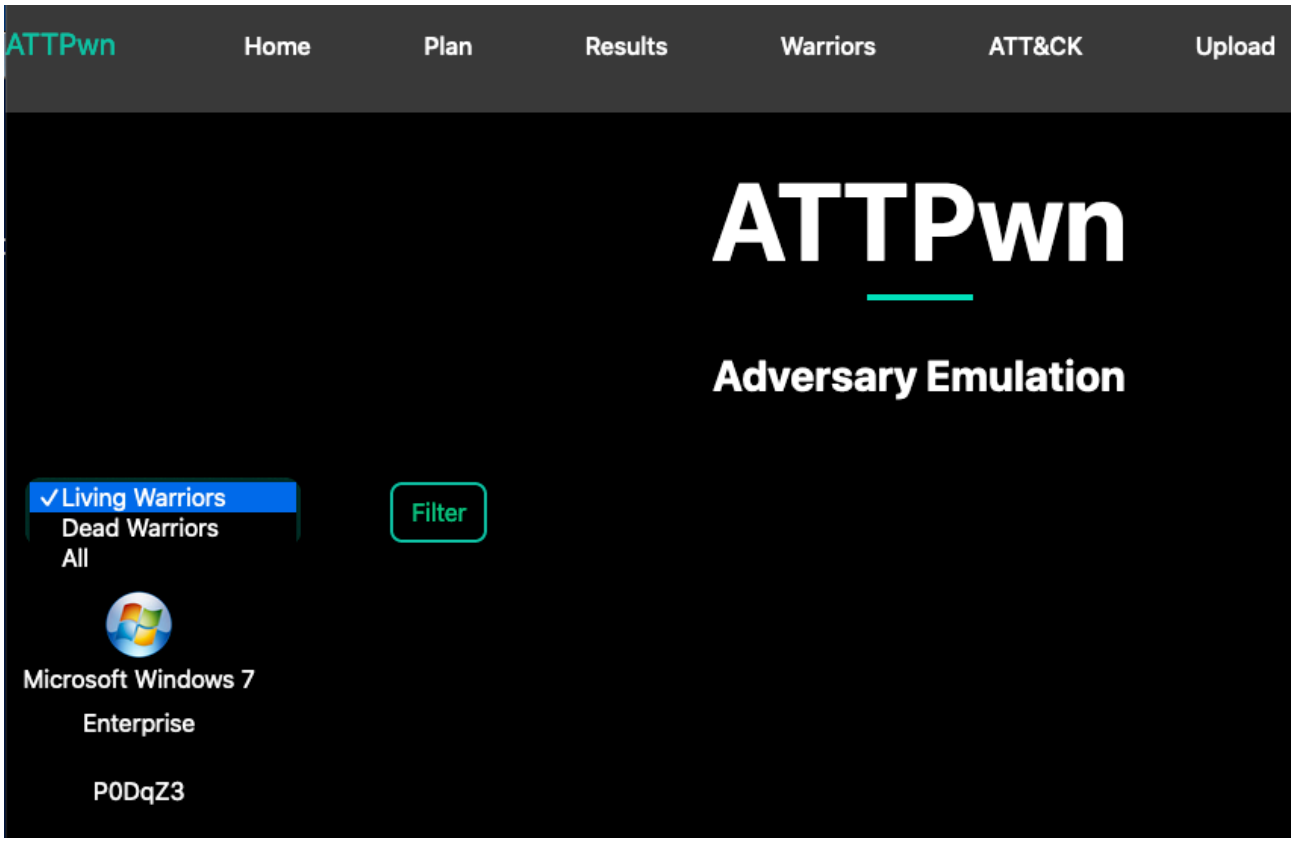



*Figura 15: Pantalla de 'Home' con los warrios vivos*

Cuando se pulsa sobre el warrior se puede obtener un mayor detalle:

- Identificador de Warrior o ID Warrior.

- Versión del sistema operativo.
- Arquitectura.
- PID o identificador del proceso dónde se está ejecutando la consola de ATTPwn.
- El campo de última visita. Un warrior estará realizando peticiones sobre ATTPwn hasta que finaliza su trabajo o hasta que éste muere por alguna causa. ATTPwn controlará la vida útil de dicho warrior mediante el uso de dicho campo.
- Dirección IP desde dónde se ha conectado el warrior.


*Figura 16: Información del warrior*

### 4.7.2. Plan

La asignación de una amenaza a un warrior y la generación de un nuevo plan de amenaza se llevan a cabo a través de la vista "Plan".

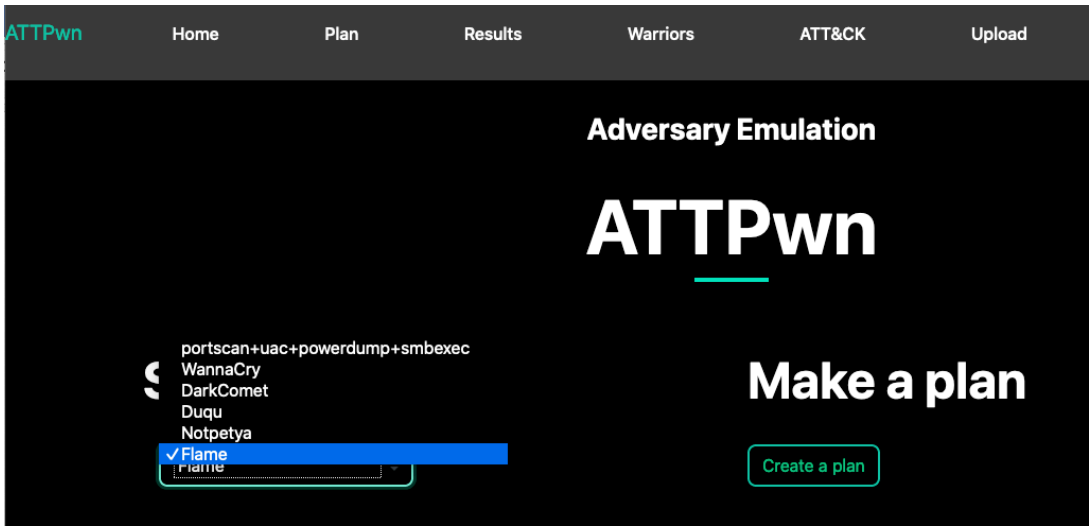



*Figura 17: Gestión de amenazas y planes*

Se puede crear una nueva amenaza pulsando sobre “Create a plan”. Aparecerá una nueva vista en la que se puede ir seleccionando diferentes tácticas del marco de trabajo ATT&CK y sus técnicas asociadas, tal y como se puede visualizar en la imagen. Una vez seleccionadas las técnicas de nuestra amenaza se pulsar el botón “Insert Plan”.

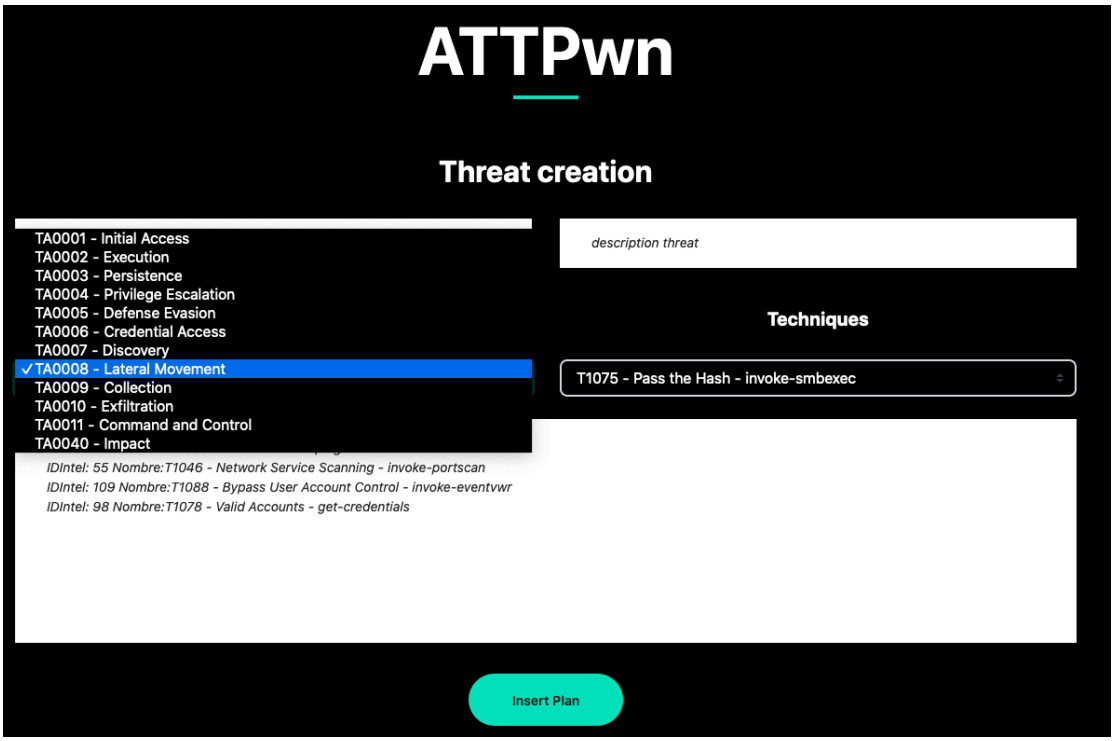



*Figura 18: Creación de una nueva amenaza*

Cuando se accede a una amenaza ya creada se puede ver el listado de técnicas que la componen. Estas se irán ejecutando paso a paso e irán devolviendo resultados y datos que pueden ser utilizados por otras técnicas. Esto es uno de los grandes valores que tiene ATTPwn. En el combo “Warrior ID” se dispone de los Warriors que están vivos y a los que se puede asociar un plan a través del botón “Allocate Plan”.

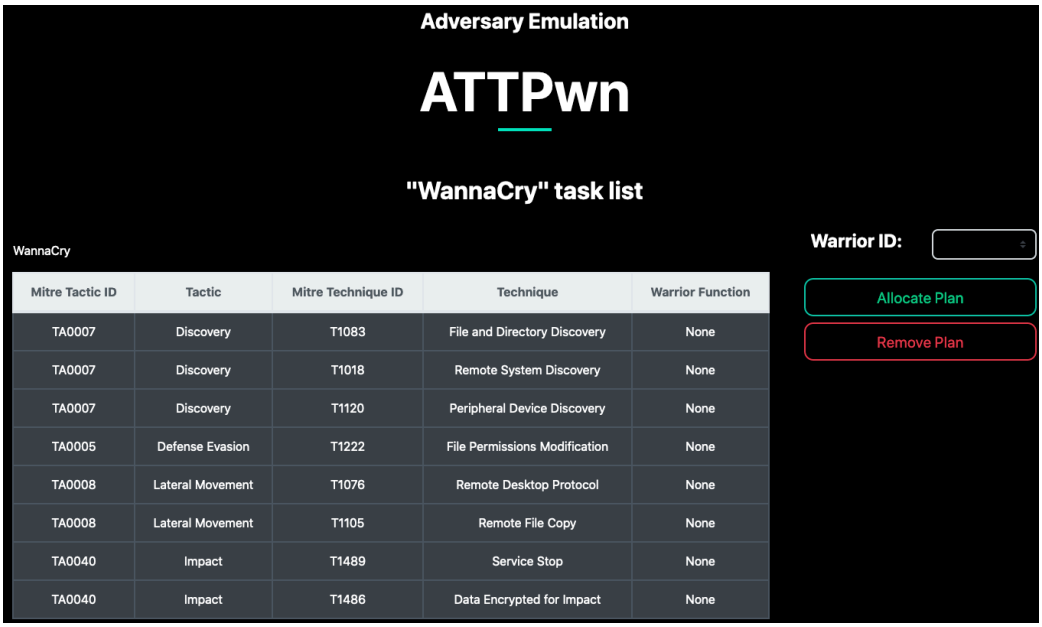



*Figura 19: Asociación de warrior a un plan*

### 4.7.3. Resultados

En la vista de resultados se pueden visualizar dos partes: el resumen ejecutivo y el resumen técnico. En la parte ejecutiva se presentan las diferentes instancias de la consola que se ha ejecutado en las máquinas Windows. En la parte técnica se proporciona el acceso a la información de las diferentes ejecuciones de las técnicas en su conjunto.

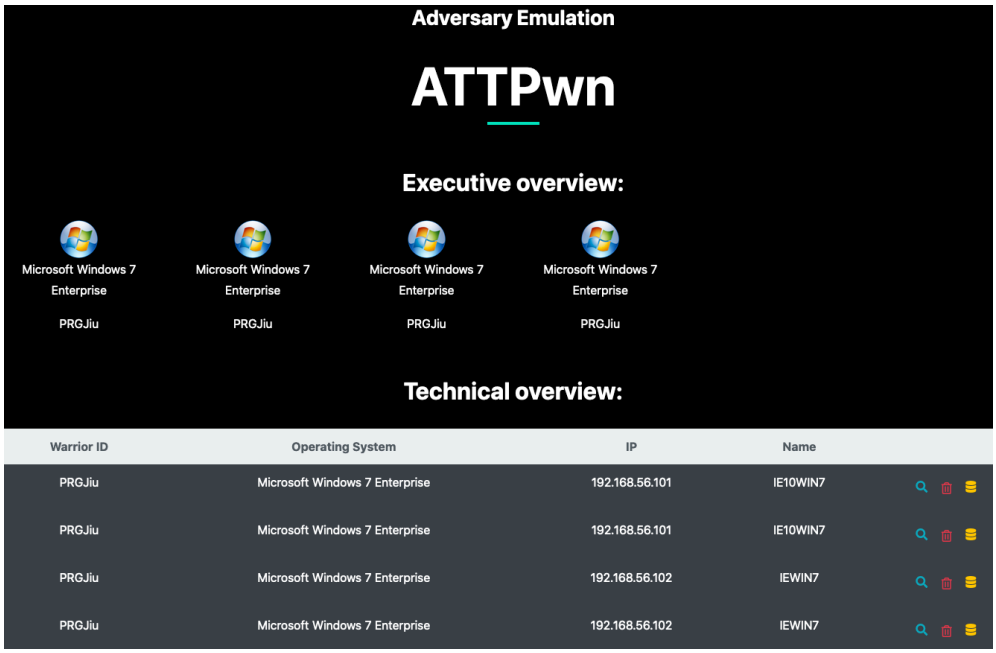



*Figura 20: Acceso a los resultados ejecutivos y técnicos*

En la parte ejecutiva, si se accede a la información se puede visualizar el siguiente contenido:

- ID Warrior.
- Sistema operativo.
- Nombre del plan de la amenaza.
- Tabla con la técnica ejecutada por este warrior, si se ha finalizado y el resultado de ésta.

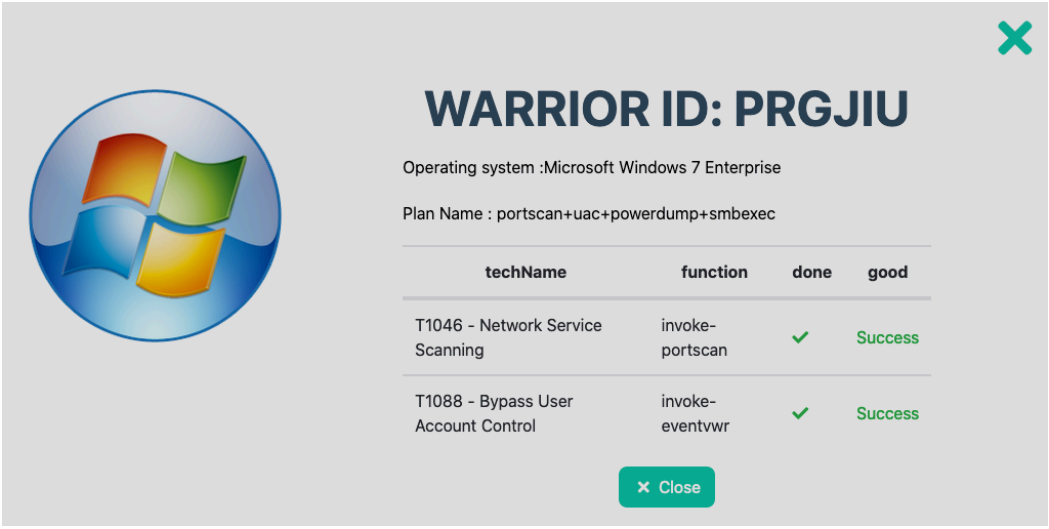



*Figura 21: Resultado ejecutivo*

Si se accede a la parte técnica hay varias opciones:

- El icono de la “lupa” proporciona información sobre los resultados de las ejecuciones. Se puede ver la salida por pantalla de cada técnica ejecutada.
- El icono de la “papelera” permite eliminar la información de un warrior.
- El icono de la “base de datos” da acceso al “datastore”. Un elemento que es utilizado para intercambiar información entre Warriors que están ejecutando la misma emulación de la amenaza.

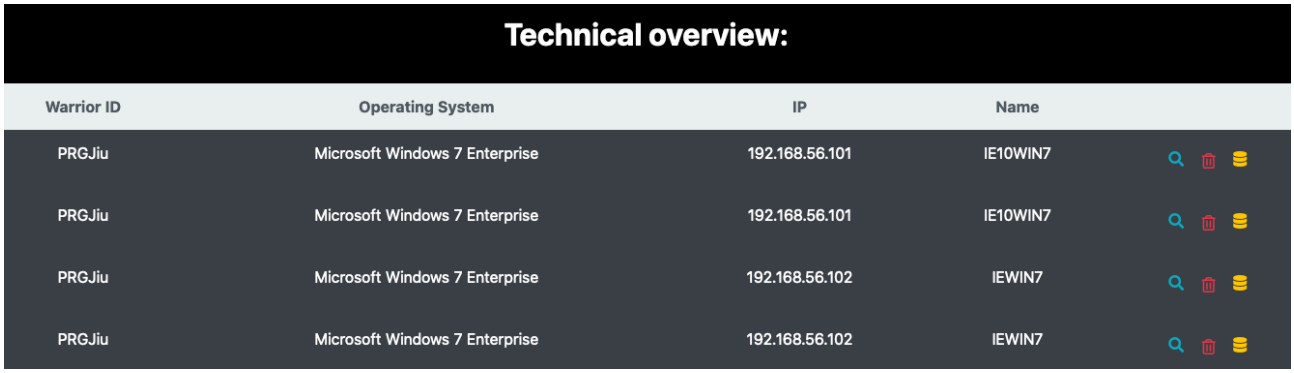



*Figura 22: Panel del resultado técnico*

### 4.7.4. Generación de un warrior

Para la generación de un warrior se utiliza un panel denominado “Warriors”. Su funcionamiento es muy sencillo: se introduce la dirección IP del nodo raíz, es decir, de ATTPwn y se genera una pequeña instrucción que invocará una Powershell y hará que se descargue el fichero ‘consola’.

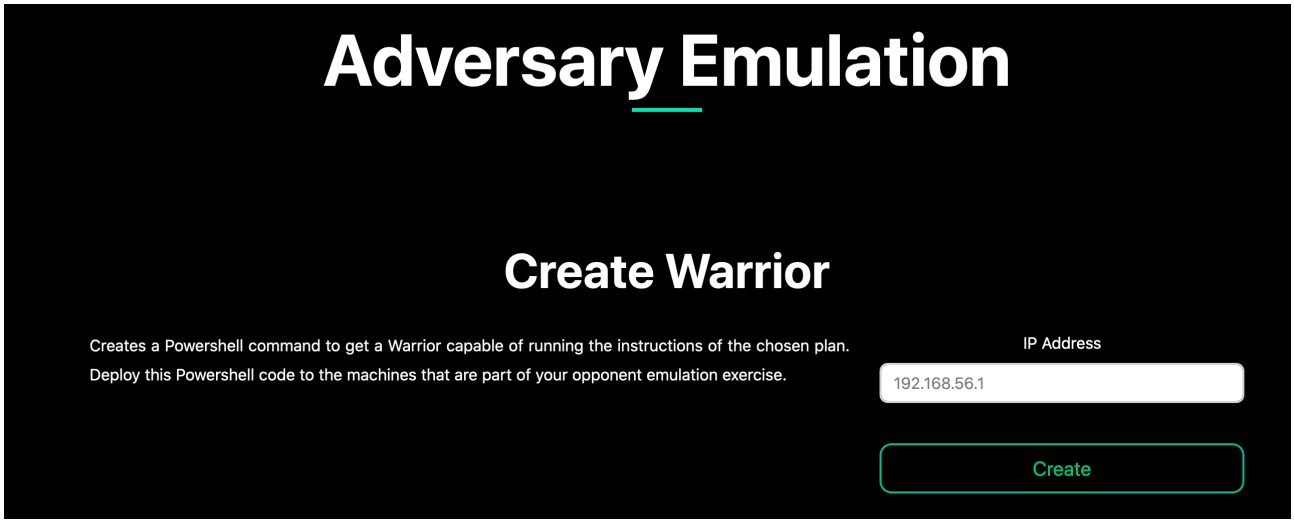



*Figura 22: Creación de warrior*

Una vez obtenida la instrucción se puede desplegar en las máquinas que formen parte de la emulación del adversario. Cuando se ejecuta la consola, ésta ya es la instancia de un warrior, por lo que hará el flujo de registro contra el nodo raíz a través del método ‘Hi’ comentado en el apartado de ‘Flujos de comunicación’.

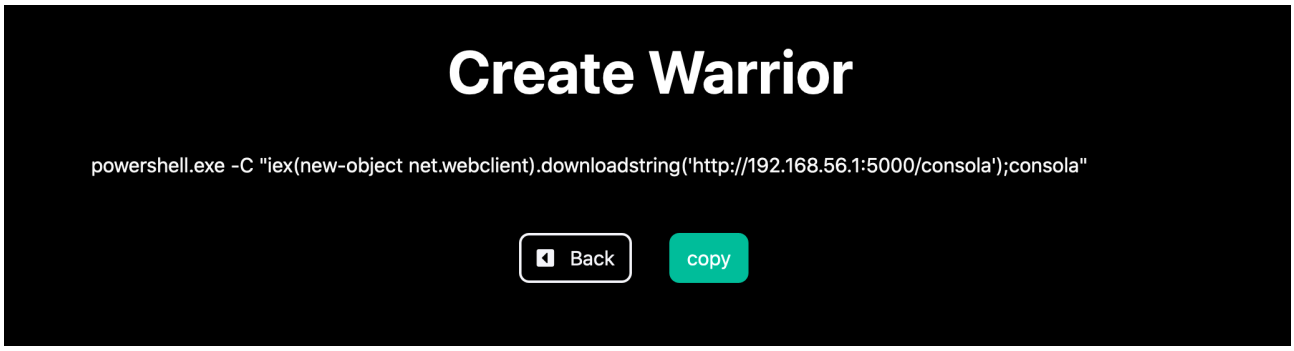
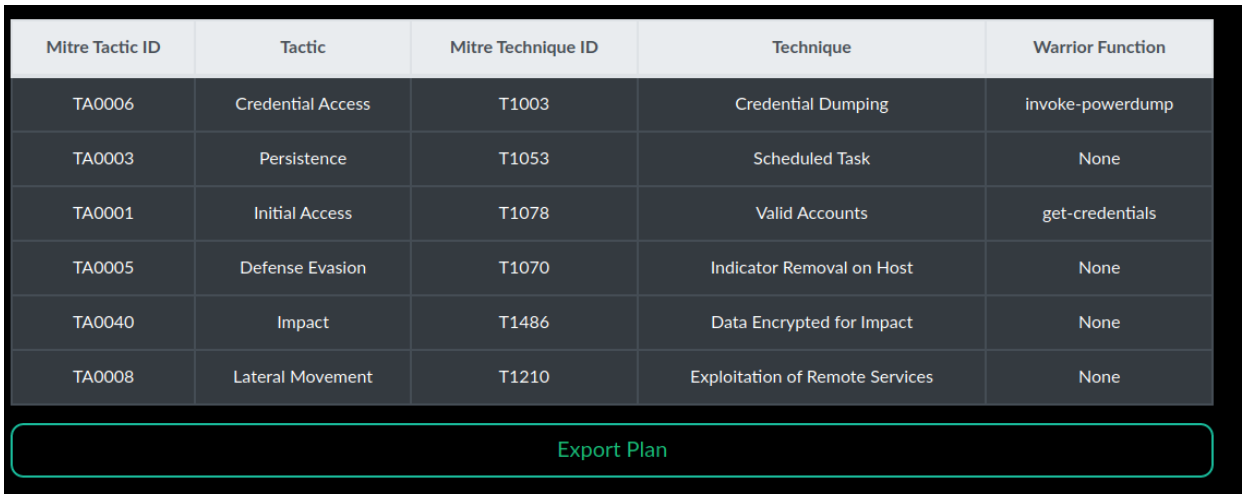



*Figura 23: Warrior creado y listo para desplegarse*

### 4.7.5. Exportación / Importación de amenazas (Plan colaborativo)

ATTPwn es un proyecto colaborativo dónde se pretende compartir el conocimiento a través de los planes de amenazas. Por esta razón, la herramienta permite la exportación de amenazas generadas por el usuario e importación en otras instancias de ATTPwn.

Durante la generación o consulta del plan de amenazas se puede hacer uso de la opción "Export Plan", tal y como se puede visualizar en la imagen.



| Mitre Tactic ID | Tactic | Mitre Technique ID | Technique | Warrior Function |
|---|---|---|---|---|
| TA0006 | Credential Access | T1003 | Credential Dumping | invoke-powerdump |
| TA0003 | Persistence | T1053 | Scheduled Task | None |
| TA0001 | Initial Access | T1078 | Valid Accounts | get-credentials |
| TA0005 | Defense Evasion | T1070 | Indicator Removal on Host | None |
| TA0040 | Impact | T1486 | Data Encrypted for Impact | None |
| TA0008 | Lateral Movement | T1210 | Exploitation of Remote Services | None |

Export Plan

*Figura 24: Exportación de un plan de amenaza*

Cuando se almacena el fichero en formato JSON, éste puede ser compartido con la comunidad. El fichero JSON alberga toda la información necesaria para reconstruir el plan de amenaza con todas las tareas e implementaciones de técnicas necesarias, aunque éstas no se encuentren disponibles en el nuevo entorno.

En el entorno nuevo desde el recurso "Upload" se puede llevar a cabo la carga de datos del plan de amenaza, tal y como se puede visualizar en la siguiente imagen.

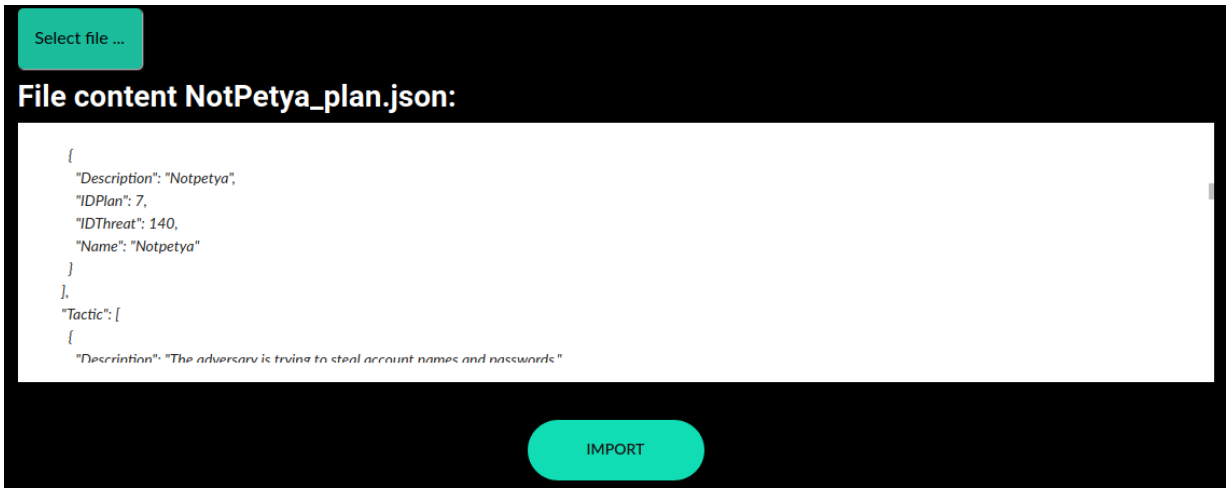



*Figura 25: Importación de un plan de amenaza colaborativo*

A partir de este momento, ATTPwn proporciona el nuevo plan de amenaza para el usuario.

# 5. Escenarios de emulación de adversarios

En este apartado se muestran diferentes ejemplos de planes de amenaza con diferentes técnicas que implementan diferentes escenarios de emulación de adversarios.

## 5.1. Amenaza con escalada de privilegios

En este apartado se muestra la creación de una amenaza que genera una escalera de privilegios y, posteriormente, realiza un volcado de credenciales de la máquina. ATTPwn proporciona un sistema que permite "desdoblar" un proceso, es decir, de un proceso con PID 'x' sale un nuevo proceso con PID 'y', aunque desde el punto de vista lógico ambos tendrán el mismo ID Warrior.

El objetivo de esta técnica es que el segundo proceso sea ejecutado en diferentes escenarios como:

- Una escalada de privilegios. Del proceso sin privilegios se explota la vulnerabilidad o debilidad y se consigue un nuevo proceso con privilegios.
- Un movimiento lateral. Ejecución de código en otro sistema gracias a alguna técnica de movimiento lateral.

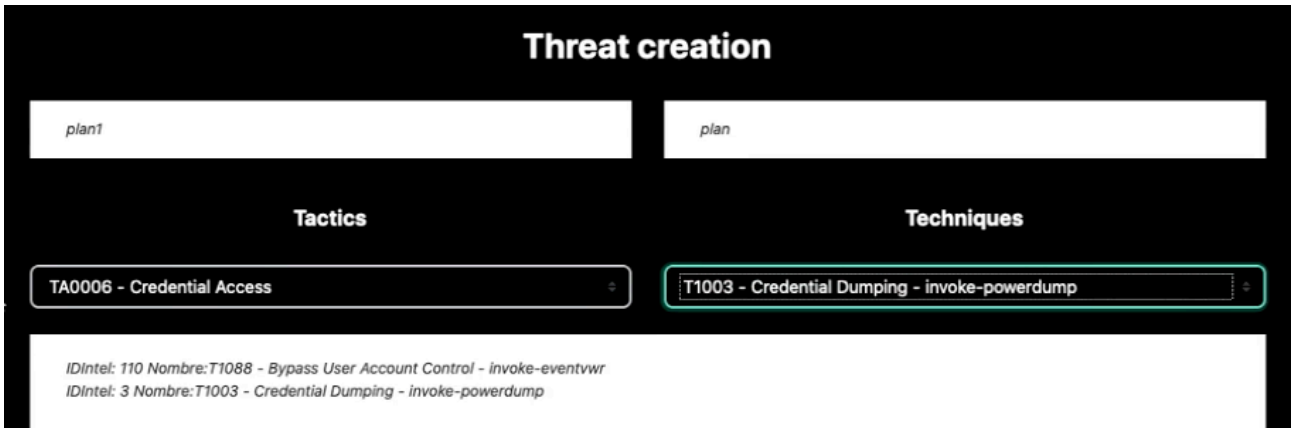



*Figura 26: Generación de un plan con elevación de privilegios*

Según el plan creado, se ejecutará una técnica de escalada de privilegios y, posteriormente, si ésta ha tenido éxito se podrá ejecutar un volcado de credenciales. La segunda tarea depende de la primera, es decir, si la primera tiene éxito la consola de ATTPwn se ejecutará en un nuevo proceso privilegiado y solicitará el resto de tareas del plan que quedan por ejecutar al nodo raíz. Si la primera tarea falla debido a controles de defensa de la organización, la primera tarea devolverá un error y la segunda tarea será ejecutada sin privilegios, por lo que o se ejecutará de forma errónea o se abortará su ejecución.


*Figura 27: Bypass UAC logrado y volcado de credenciales pendiente o en ejecución*

Cuando la escalada de privilegios se ha llevado a cabo, se puede observar en el nodo raíz como se dispone de un nuevo proceso con el mismo ID Warrior.

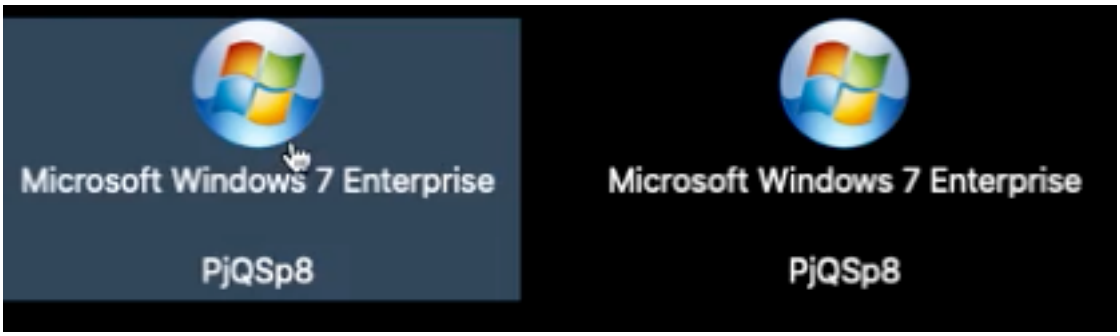



*Figura 28: "Spawn procces"*

En el segundo proceso se puede ver el resultado de la ejecución del volcado de credenciales.

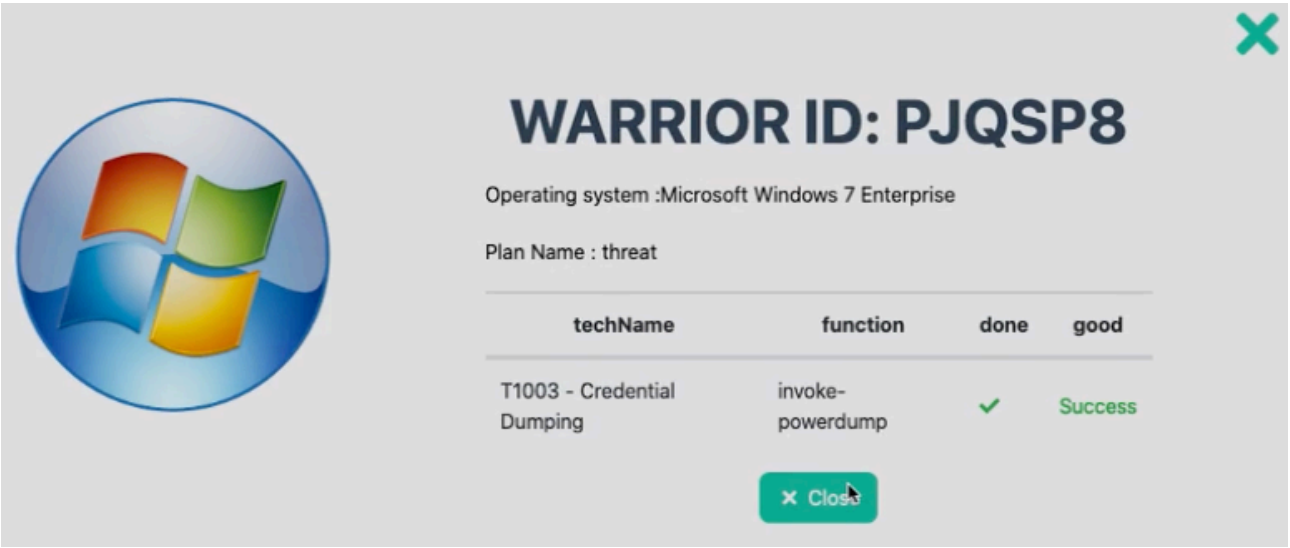



*Figura 29: Credential Dumping correcto*

## 5.2. Amenaza con movimiento lateral

En este apartado se muestra cómo generar un plan avanzado de amenaza emulando el funcionamiento de un adversario. El objetivo es desplegar la amenaza en un warrior ejecutado en una máquina Windows 7 y que éste consiga desplazarse a una máquina de la red con Windows 10. El escenario es el siguiente:

- Se ejecutará una técnica de descubrimiento de máquinas en la red. Esta técnica devolverá al 'Datastore' una serie de direcciones IP descubiertas con las que la máquina dónde se ejecuta el warrior tiene conectividad.

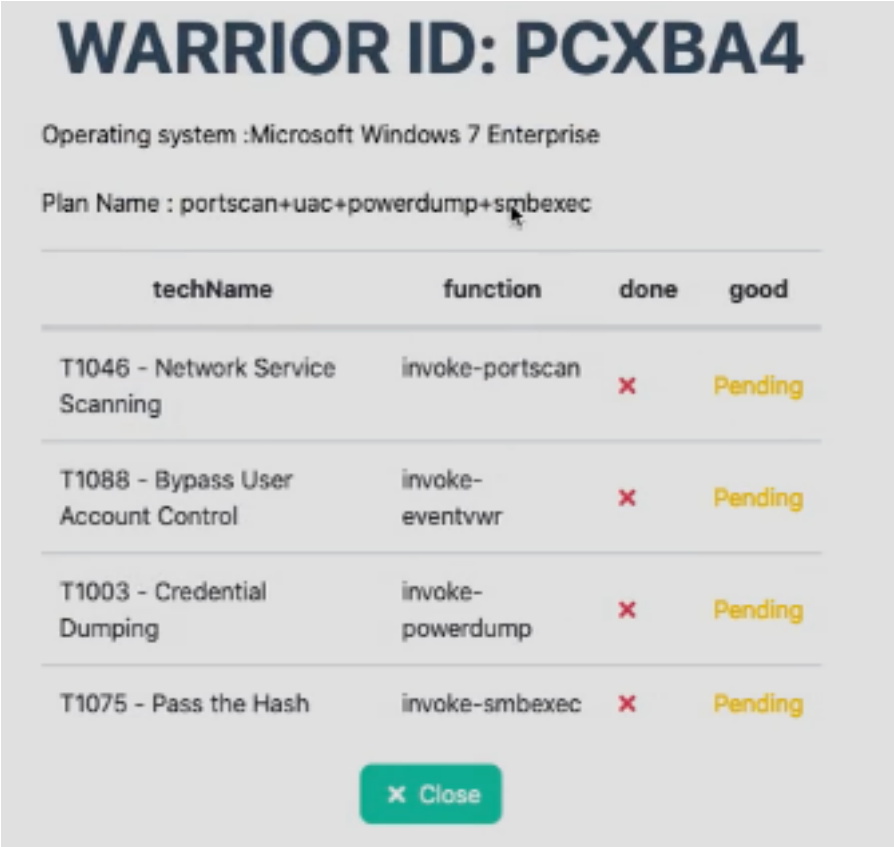



*Figura 30: Planning de la amenaza*

- Posteriormente, se llevará a cabo una escalada de privilegios con el objetivo de poder disponer de credenciales lo suficientemente elevadas como para poder realizar un movimiento lateral a otra máquina.

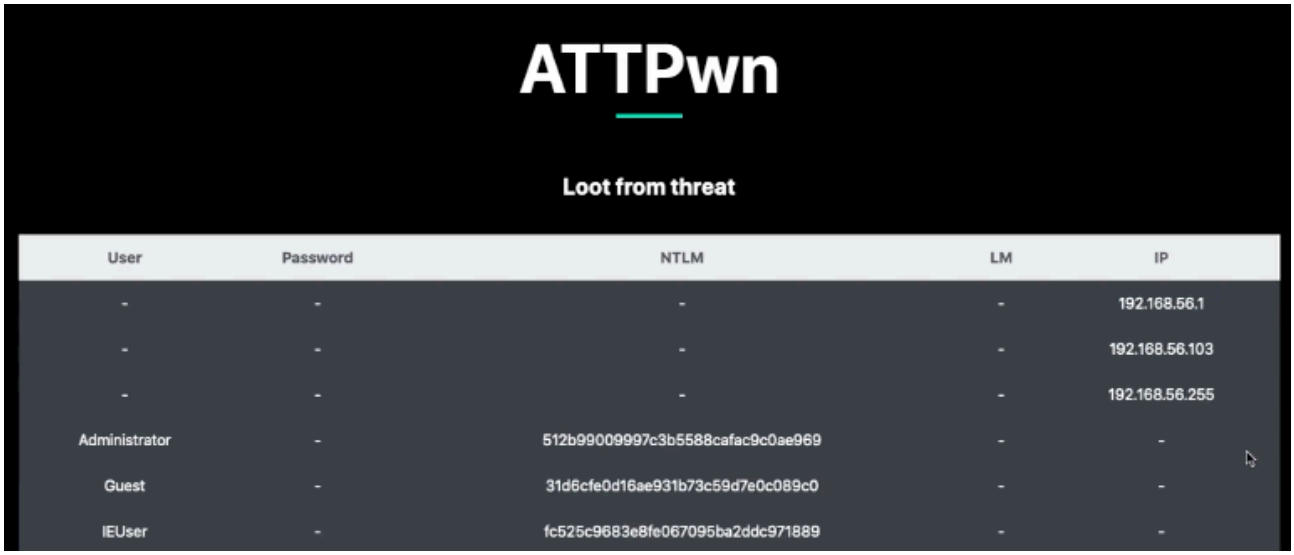



*Figura 31:Datastore con datos recogidos en técnicas anteriores de la amenaza*

- Tras la escalada de privilegios se realiza un volcado de credenciales. En esta técnica el objetivo es obtener credenciales que permitan desplazarnos a otra máquina de la red.
- Obteniendo credenciales, que son almacenadas en el 'Datastore' en pasos anteriores, se utilizará una técnica que obtenga dichas credenciales y vaya probando con las direcciones IP anteriores. Se utiliza una técnica de movimiento lateral, en esta ocasión, basada en pass-the-hash.


*Figura 32: Mismo Warrior en dos máquinas diferentes (Windows 7 y Windows 10)[7][8][9]*

## 6. Referencias

[1] https://attack.mitre.org

[2] https://mitre-attack.github.io/attack-navigator/enterprise/#

[3] https://attack.mitre.org/resources/adversary-emulation-plans/

[4] Qurtuba Security Congress 2015. https://qurtuba.es/2015/sessions/pablo-gonzalez/

[5] RootedCON Valencia 2016. https://rootedcon.com

[6] iBombShell. https://github.com/elevenpaths/ibombshell

[7] ATTPwn Example 1. https://www.youtube.com/watch?v=2Y3F5uxXXSM

[8] ATTPwn Example 2. https://www.youtube.com/watch?v=VQHVgfgdJwM

[9] ATTPwn Example 3. https://www.youtube.com/watch?v=1y2D8801jRI

*Icons designed by Freepik from Flaticon

*Icons made by Freepik, Smashicons and Dave from www.flaticon.com