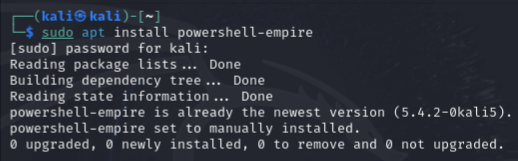
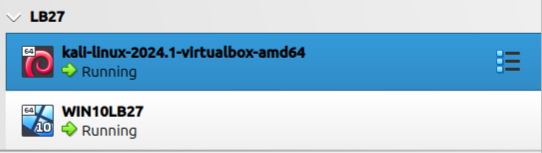
Hélio Ferreira 01/07/2024

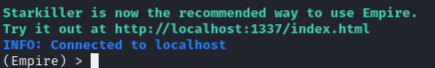
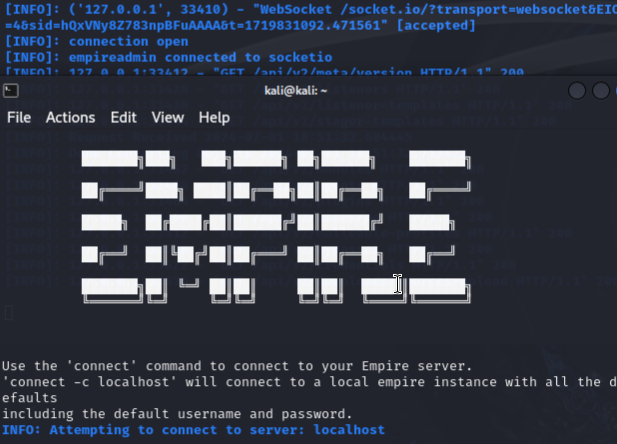
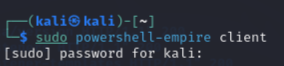
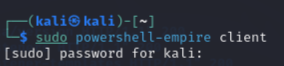
# **Lab: Persistence**

### **Part 1: Staging**

For today’s lab you’ll need a Windows 10 VM and a Kali Linux VM with Empire installed. Kali will be our C2 server. To install PowerShell Empire on your Kali VM, simply run the command: **sudo apt install powershell-empire**

### **Start PowerShell Empire**

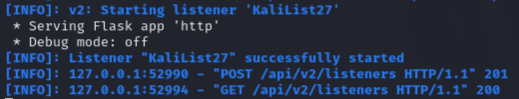
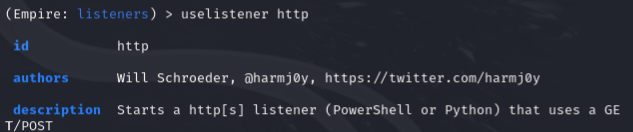
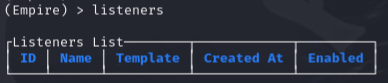
Open up a terminal and start the Empire server by running **sudo powershell-empire server**. When the Empire server starts successfully, open up another terminal and start the Empire client by running **sudo powershell-empire client**. You should now have the Empire interactive prompt

.

### **Part 2: Setup a Listener**

To prepare the attack, you’ll need to setup a listener in Empire. Reference the provided resources as to how to execute the procedures in this lab. Make sure the listener is running before proceeding to the next stage fo the lab. Take a look at these commands, then access Empire and attempt to setup your own listener. Note that these are not sequential lab instructions.

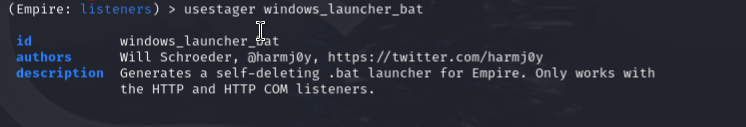
* **listeners** will say no listeners currently active and enter listeners mode
* **back** will exit listeners mode
* **uselistener [protocol]** will add a listener for the entered protocol.
* **info** prints information about your listener. Key attributes to take notice of include:
  + Name
  + BindIP
  + Port
* **set Name** can change the name of your listener
* **set Port [number]** can change the port number of your listener
* **execute** will run the listener; a listener must be running in order to capture inbound data from the victim

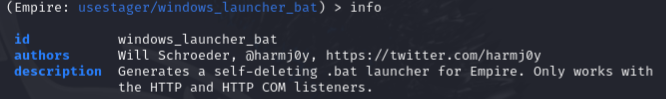


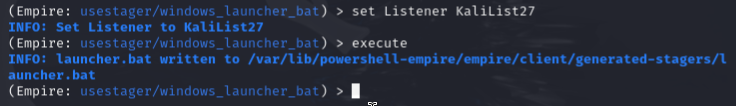
Once your listener is up and running on a given port and protocol, proceed.listeners

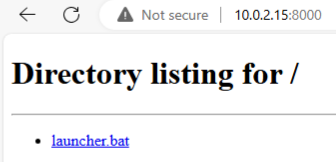
### **Part 3: Setup a Stager**

Next, it’s time to setup a stager. Generate **launcher.bat**, copy it to the victim host, then run it from the terminal (you won’t see the output if you run it in File Explorer). If you encounter security systems blocking the execution of the script, feel free to disable those systems (and ensure they remain disabled, and don’t turn themselves back on…), but remember to document any such actions in your submission. You should see the connection established to the victim PC and be able to interact with it via the Empire shell. Here are some useful commands you can use (again, note that these are not sequential lab instructions):

* The **usestager [stager]** command will setup a stager for use.
  + Example, **usestager windows/launcher bat** then type **info**

****

* **set Listener test** then **info** will show the updated listener name
* **execute** gives “Stager output written out to: /tmp/launcher.bat”

* **agents** lists compromised systems that you now have access to.
* **rename [current] [new]** changes the agent’s name.
* **interact [agent name]** launches the shell.

Were you able to issue commands to the Windows 10 victim computer? Make sure you establish a working shell before proceeding.  
  
No, I’ve ran the script on victim computer without any problem, howerever, , our host Kali didn’t “caught” any movement.

### **Part 4: Reporting**

Discuss in your own words the following:

* Why would an APT want to establish persistence on the Cyberdyne network?  
    
  An APT would want to establish persistence on the Cyberdyne network for several key reasons:

1. **Intellectual Property Theft**: To steal valuable technology and proprietary information.
2. **Espionage**: To gather intelligence on sensitive projects, especially if related to defense.
3. **Sabotage**: To disrupt operations or create chaos.
4. **Financial Gain**: To engage in financial fraud or ransomware attacks.
5. **Credential Harvesting**: To continuously collect credentials for broader network access.
6. **Supply Chain Attacks**: To compromise connected organizations through Cyberdyne.
7. **Long-Term Control**: To maintain ongoing surveillance and control.
8. **Data Manipulation**: To alter or corrupt data, impacting product integrity and operations.

* What kind of threat actor are we dealing with here?  
    
  We are dealing with an **Advanced Persistent Threat (APT)**, characterized by:

1. **Sophistication**: Uses advanced techniques and custom malware.
2. **Persistence**: Aims for long-term network access.
3. **Resourceful and Well-Funded**: Often state-sponsored with significant resources.
4. **Targeted Attacks**: Focuses on high-value targets like defense contractors and tech firms.
5. **Goal-Oriented**: Motivated by strategic objectives, such as espionage or sabotage.
6. **Stealthy and Patient**: Operates covertly to avoid detection and achieve long-term goals.

* Based on your reproduction, how could the batch file payload have been transmitted to the victim and executed the first time?  
    
  The batch file payload could have been transmitted and executed through:

1. **Phishing Emails**: Containing a malicious attachment or link.
2. **Malicious Websites**: Exploiting browser vulnerabilities for drive-by downloads.
3. **Social Engineering**: Tricking the victim into downloading and running the file.
4. **Removable Media**: Using a USB drive with the batch file.
5. **Exploiting Vulnerabilities**: Leveraging software or OS vulnerabilities to deliver the file.
6. **Malware Bundles**: Bundling with legitimate software as a Trojan horse.
7. **Remote Desktop Protocol (RDP)**: Using weak or compromised credentials to transfer and execute the file remotely.

* Reboot the Windows 10 VM. Is the agent still responding to Empire? Explore the limitations of such a technique.  
    
  As I said before, I couldn’t find a way our victims respond to Kali .
* Is there a way to configure the batch file to not delete itself after executing? Explain.  
    
  By default, a batch file cannot delete itself while it's running. This is a safety measure to prevent accidental deletion or malfunction. However, there are a few workarounds to achieve the effect you want:

1. **Remove the delete command:** The simplest solution is to remove the del %0 command from the end of your batch file. This line specifically targets the file itself for deletion.
2. **Redirect delete output:** You can suppress the error message that appears when the script tries to delete itself while running. Add del "%~f0" >nul at the end. Here, >nul redirects any output from the del command to oblivion (represented by nul).
3. **Two-step deletion:** Create a separate short batch file with just the del "%~f0" command. In your main script, use start /b "hidden name" del\_script.bat to launch the deletion script in the background. /b hides the console window and "hidden name" sets a temporary name for the process. This achieves deletion after the main script finishes.
4. **Rename instead of delete:** If you just want the script to be inactive after running, consider renaming it with a .bak extension at the end. Use ren "%~f0" "%~f0.bak" to achieve this.

* Evaluate technique T1037. As the threat actor, what kind of synergy does this technique offer alongside T1059.003?  
    
  Here are some approaches organizations can take to counter this combined attack:

**Mitigating T1037:**

* **Group Policy:** Utilize Group Policy Objects (GPOs) to restrict users from modifying their logon scripts or limit scripts to run from approved locations.
* **Script Signing:** Implement script signing to ensure only authorized and signed scripts can run during logon.
* **Endpoint Detection and Response (EDR):** Employ EDR solutions to monitor for suspicious modifications to logon scripts or unauthorized script execution.

**Mitigating T1059.003:**

* **Application Whitelisting:** Implement application whitelisting to restrict execution only to approved programs, preventing malicious tools downloaded through the script.
* **Network Traffic Monitoring:** Monitor network traffic for suspicious activity related to downloading tools or communicating with command and control servers.
* **Advanced Threat Detection (ATD):** Utilize ATD solutions to identify and block malicious commands or scripts attempting to exploit the system.

**Additionally:**

* **User Education:** Train users to be cautious about opening attachments or clicking on links, especially in emails.
* **Regular Security Updates:** Ensure systems are patched with the latest security updates to address vulnerabilities attackers might exploit.

By implementing a layered approach that combines these mitigation strategies, organizations can significantly reduce the risk of successful attacks using T1037 and T1059.003 together.

* At a high level, brainstorm what we could potentially do as defenders to protect Cyberdyne against this type of threat. Explore the various types of security controls:
  + Preventative
  + Detective
  + Corrective  
      
    **Protecting Cyberdyne from T1037 and T1059.003: A Multi-Layered Defense**

**Scenario:** A threat actor aims to gain persistence and evade detection on Cyberdyne's systems by combining T1037 (Boot or Logon Initialization Scripts) and T1059.003 (Command and Scripting Interpreter: Windows Command Shell).

**Our Goal:** Implement a layered security approach using preventative, detective, and corrective controls to mitigate this threat.

**Preventative Controls:**

* **System Hardening:**
  + Restrict user privileges to minimize the ability to modify logon scripts.
  + Disable unused ports and services to limit potential attack vectors.
  + Implement application whitelisting to restrict script execution only to approved programs.
  + Enforce strong password policies and multi-factor authentication.
* **Script Security:**
  + Implement script signing to ensure only authorized and signed scripts can run during logon.
  + Restrict users from modifying their logon scripts or limit scripts to run from approved locations using Group Policy Objects (GPOs).
  + Monitor script execution for suspicious activity using Endpoint Detection and Response (EDR) solutions.

**Detective Controls:**

* **Log Monitoring:**
  + Monitor system logs for suspicious activity related to logon script modifications or unauthorized script execution.
  + Analyze network traffic for indicators of compromise (IOCs) such as downloading tools from remote servers or communicating with command and control (C2) servers.
* **Security Information and Event Management (SIEM):**
  + Utilize SIEM to aggregate logs from various sources and correlate them to detect potential attacks using T1037 and T1059.003.
  + Leverage threat intelligence feeds to identify known malicious scripts or tools that might be used in the attack.

**Corrective Controls:**

* **Incident Response:**
  + Develop a well-defined incident response plan to identify, contain, and eradicate threats if an attack occurs.
  + Train personnel on incident response procedures to ensure a swift and coordinated response.
* **Vulnerability Management:**
  + Implement a vulnerability management program to identify and patch vulnerabilities in operating systems, applications, and scripts that could be exploited by attackers.
  + Regularly scan systems for vulnerabilities and prioritize patching critical vulnerabilities.
* **Threat Hunting:**
  + Proactively hunt for threats within the network by analyzing logs, network traffic, and system behavior for signs of malicious activity.
  + Conduct simulations and red teaming exercises to test the effectiveness of security controls and identify potential weaknesses.

**Additional Considerations:**

* **User Education:** Train employees on social engineering tactics and the importance of cyber hygiene to prevent them from falling victim to phishing attacks that could lead to script execution.
* **Cyber Threat Intelligence:** Stay up-to-date on the latest cyber threats and trends to adapt your security controls and improve your defenses.