Automated Attack Orchestration Lab

Table of contents

List of Figures

List of Tables

1. Lab Objective

The lab demonstrates a complete Caldera-driven emulation: initial access via a macro phishing attachment, execution and staging on a Windows host (SANDCAT agent), automated archiving of user data, and exfiltration to an attacker HTTP listener. The exfil stage uses an automated red-team technique (chunking/encoding + HTTP PUT) implemented as a Caldera ability with a small Python PUT server to receive data.

2. Tools Used

- Caldera (abilities, adversary, operations)
- RTA-style techniques implemented as Caldera abilities
- SANDCAT / PowerShell, Python PUT/POST listener

3. Lab Setup

- ***Kali : 192.168.1.48*** -Attacker (caldera,metasploit,pyphsisher)
- ***Windows 10 : 192.168.1.46 -*** Victim

4. Phase : technique mapping (MITRE ATT&CK)

- ***Delivery:*** Macro-enabled phishing document — T1204.002 (User Execution: Malicious File).
- ***Execution:*** PowerShell / SANDCAT agent starting the staging process — T1059.001 (PowerShell).
- ***Discovery:*** Enumerating files (targeting Downloads) — T1083 (File and Directory Discovery).
- ***Collection / Staging:*** Compress-Archive to create an archive — T1005 / T1560 (collect & archive).
- ***Exfiltration:*** HTTP PUT or chunked hex POST to my web listener — T1567 (Exfiltration Over Web Service) and, if via C2, T1041.
- ***Orchestration:*** Caldera abilities → adversary → operation (automated chaining inside Caldera).

5. Methodology

Step 1: Use Caldera for Adversary Emulation and login with red caldera with windows 10 using sand-cat agent

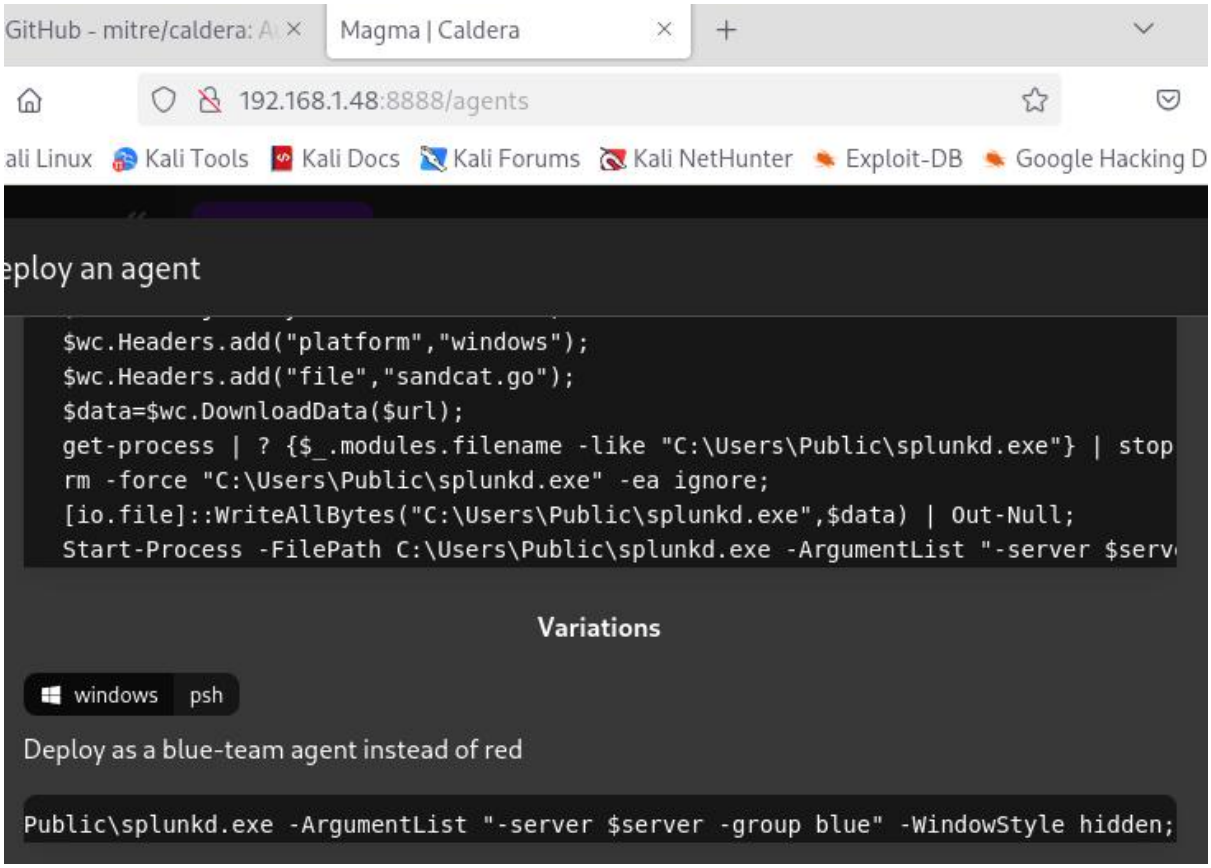


Figure 5.1 Shows agent deployment payload

Step 2: Copy the code for red team agent as paste on windows 10

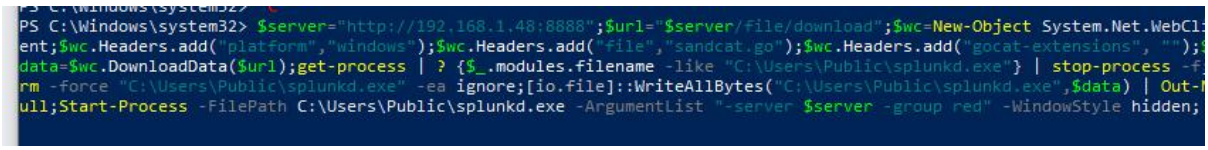


Figure 5.2 Shows payload being pasted on victim machine

Step 3: We see the agent successfully present in our red caldera

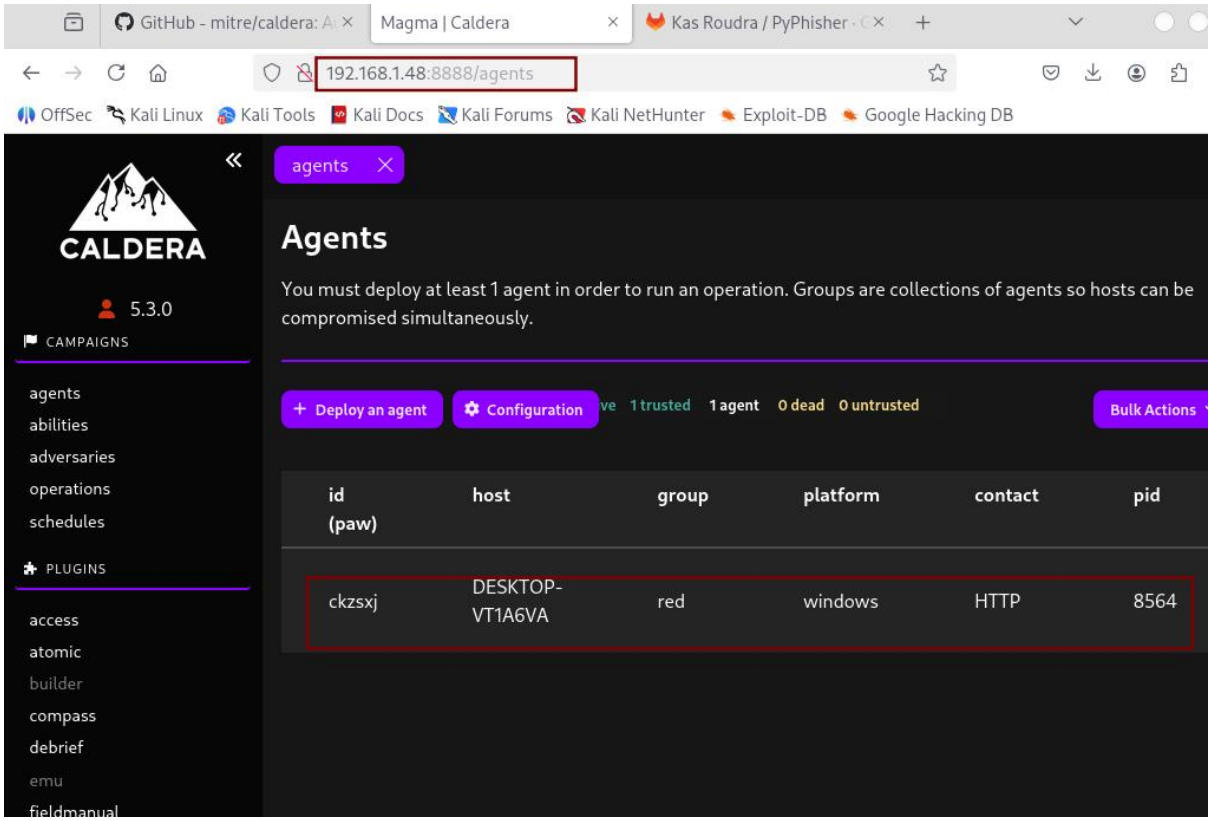


Figure 5.3 Shows agent being successfully deployed on caldera

Step 4: Once we have our agent ,lets start our attack ochestration ,

We are making use of below 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

Step 5: Start making necessary changes to the ability: ***Download Macro-Enabled Phishing Attachment*** and save it.

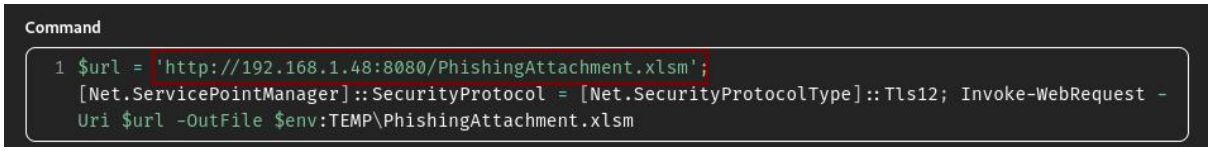


Figure 5.4 Shows making changes to macro phishing attachment

Step 6: For ability ***Zip a Folder with PowerShell for Staging in Temp*** make the following changes and save it

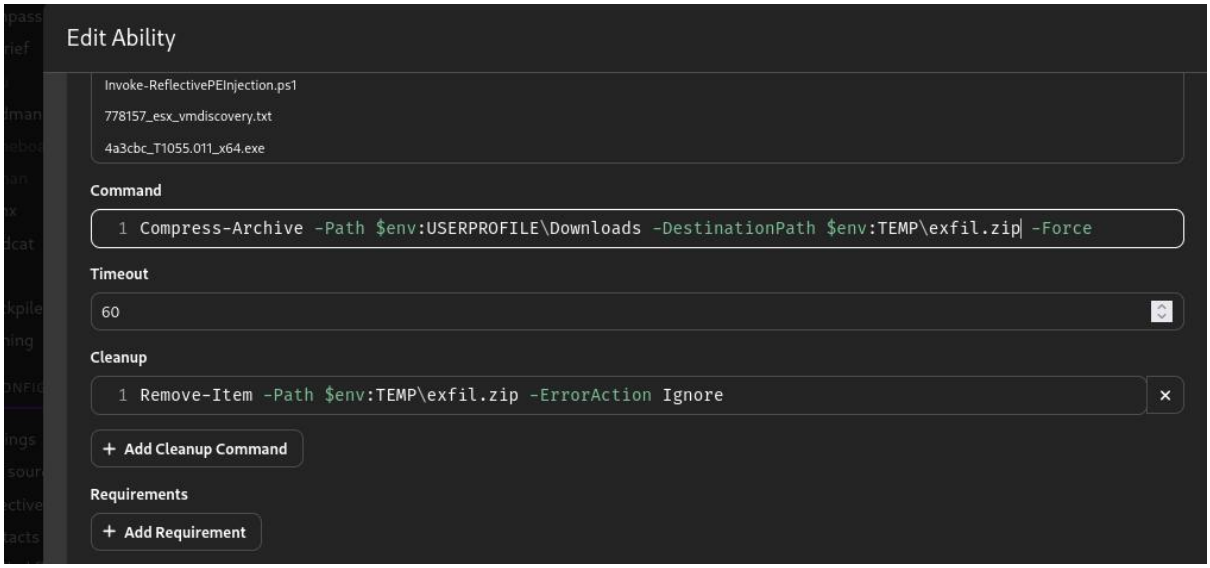


Figure 5.5 Shows making changes to zip a folder ability

Step 6: Exfiltrating Hex-Encoded Data Chunks over HTTP

Since this ability is not present we create a new ability and make the following changes

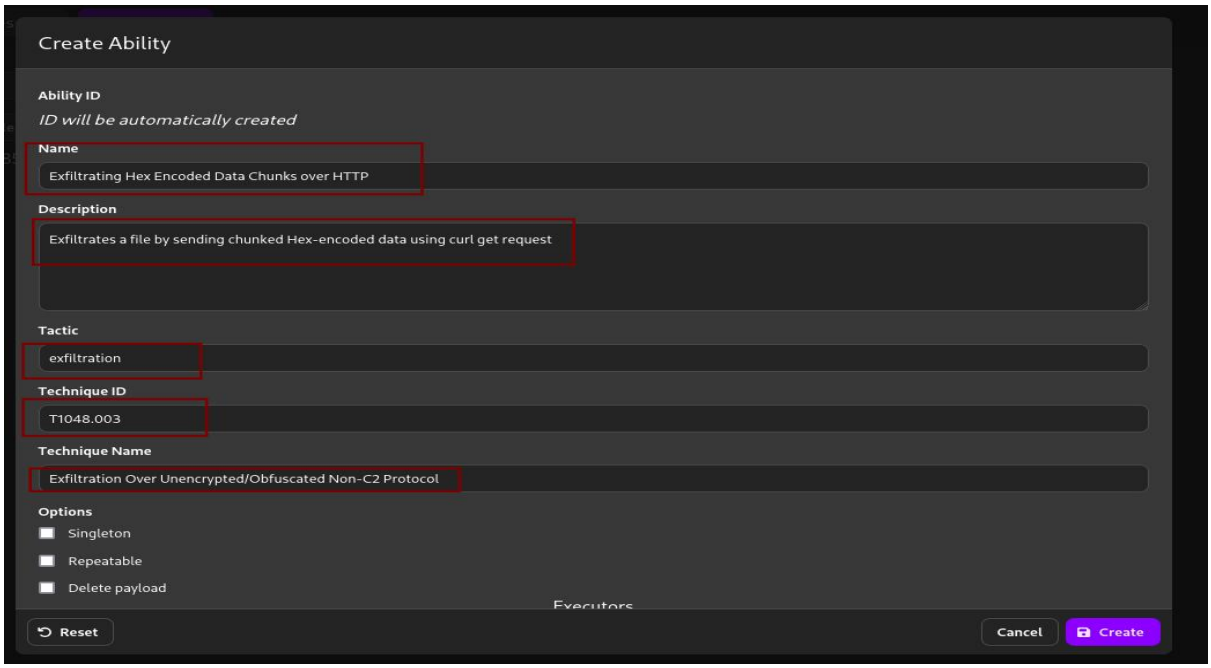


Figure 5.6 Shows creating a new ability

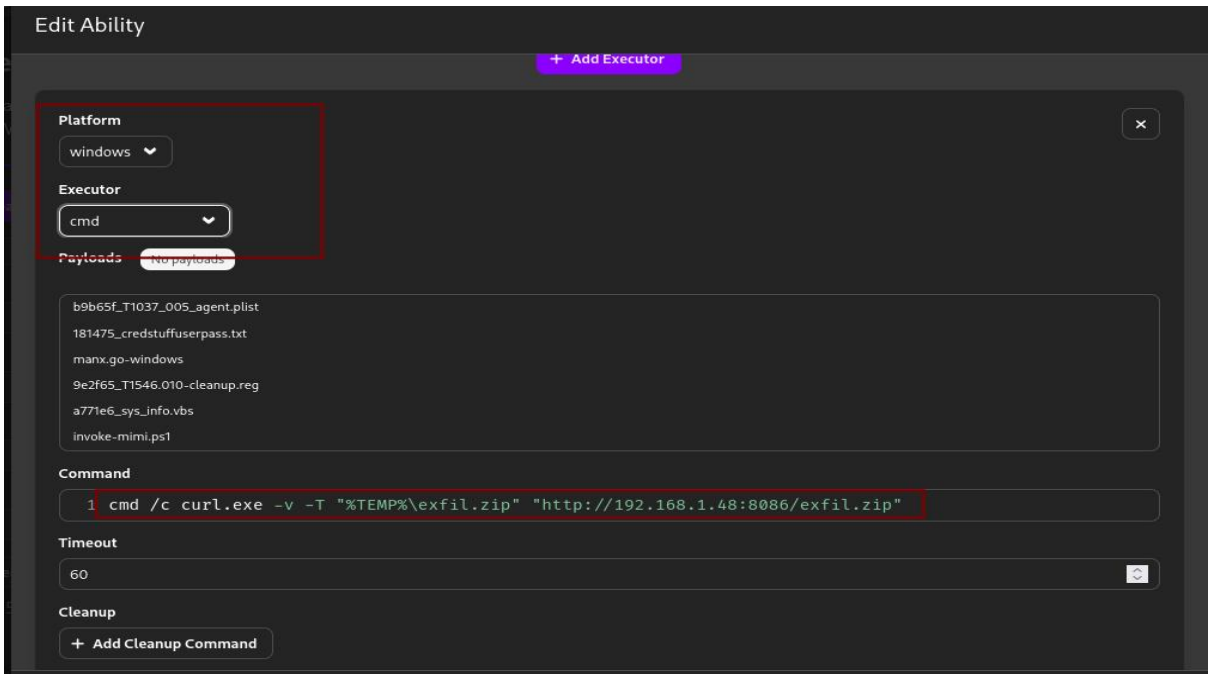


Figure 5.7 Shows making changes in executor in the new ability

5.1. RTA technique — Hex-encoded chunked exfil via curl

- The exfiltration technique uses native Windows tooling and a minimal HTTP listener to transfer a target archive from victim to attacker.
- The archive is prepared (compressed), optionally hex-encoded and split into chunks, and each chunk is uploaded from the victim using curl -T (HTTP PUT).
- The listener implements ***do_PUT*** to write received bodies to disk; after all chunks arrive, a reconstruction step concatenates the parts and decodes the hex back to the original binary archive.

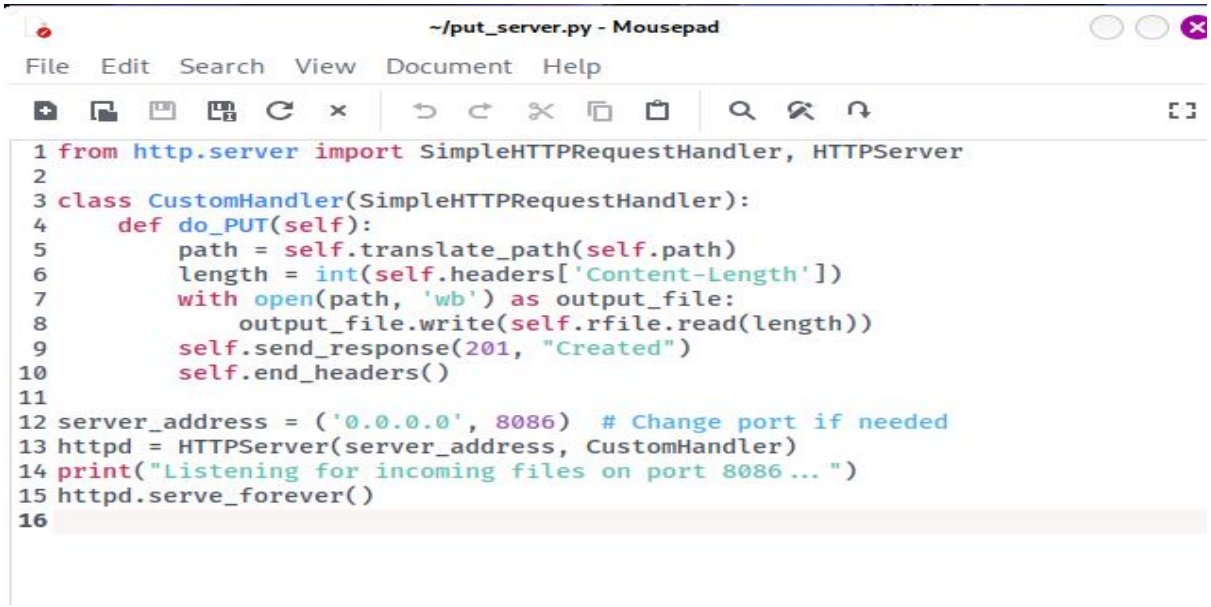


Figure 5.8 Shows python script for catching exfiltratig data

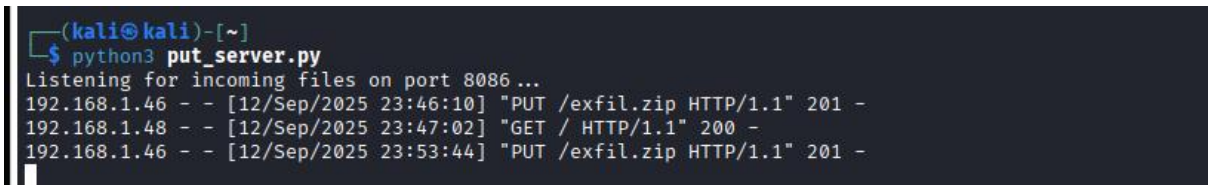


Figure 5.9 Shows python script running

Why this is an RTA technique?

- It automates multiple sub-steps end-to-end (staging → encoding → chunking → transport → reconfirm on listener).
- It uses scripted, repeatable behaviour (PowerShell + ability JSON) so the same exfil sequence can run without manual intervention.
- The chunking + hex-encoding over HTTP is a common red-team exfil pattern because it evades simple file-upload signatures and supports automation

Step 7: Creating a ***Custom Adversary Profile***

Now that we have prepared all the abilities, the next step is to create a new adversary profile. Navigate back to the adversaries tab and click New Profile.

The list of the abilities that we are going to add to the Adversary Profile.

- 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

save this before ***operation phase***

After saving the profiles, it looks like the one displayed below.

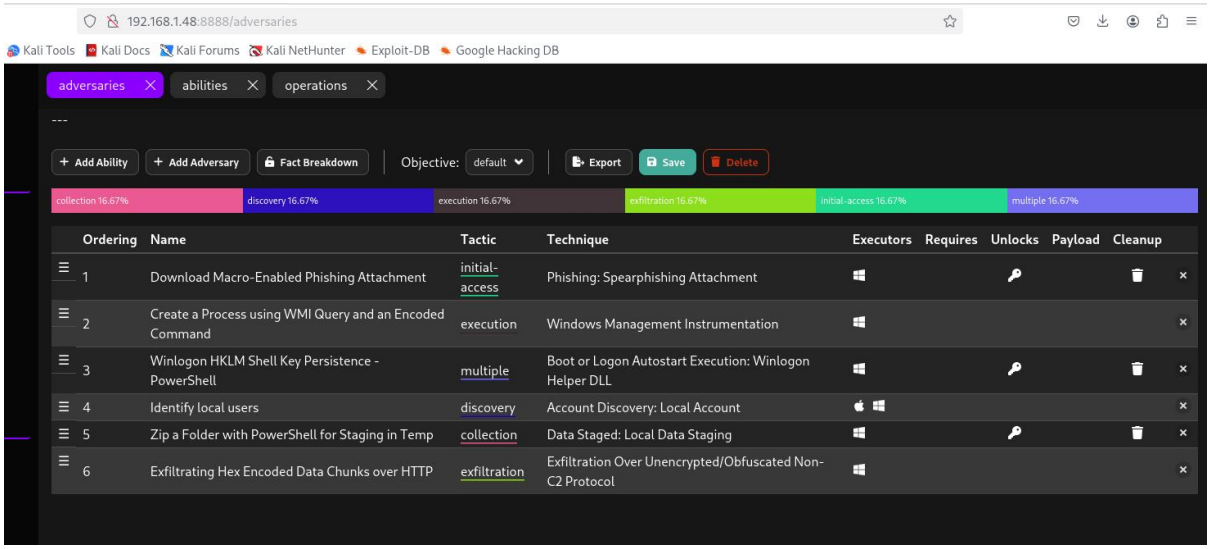
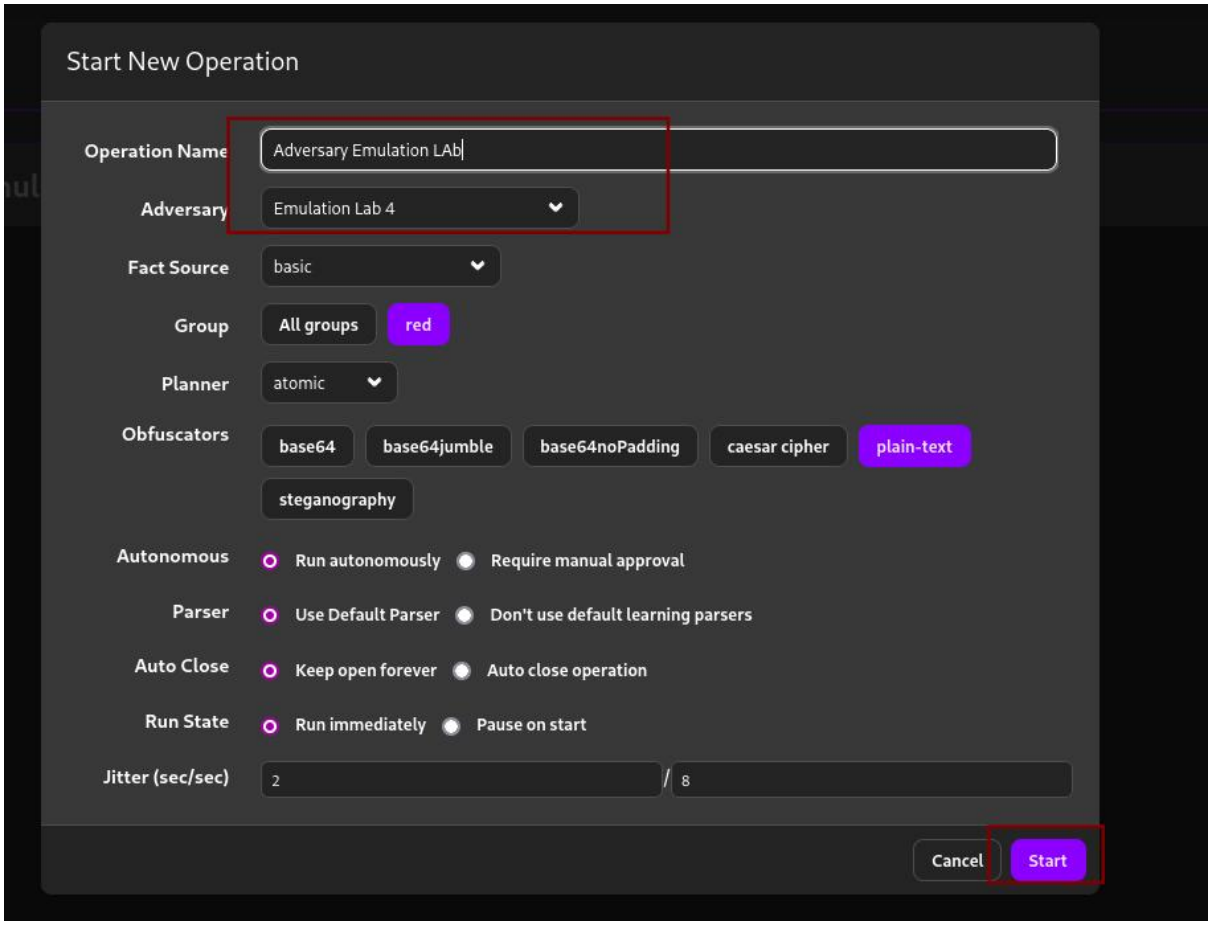


Figure 5.10 Shows adversary phases

Step 8: Running the Operation ,select the lab name we kept in above phase and add to the operations.After all the process successfully runs we get the following ,



Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output
9/12/2025, 11:49:36 PM EDT	success	Download Macro-Enabled Phishing Attachment	initial-access	ezvkta	DESKTOP-VT1A6VA	9784	View Command	No output
9/12/2025, 11:49:51 PM EDT	success	Create a Process using WMI Query and an Encoded Command	execution	ezvkta	DESKTOP-VT1A6VA	7368	View Command	View Output
9/12/2025, 11:50:21 PM EDT	success	Winlogon HKLM Shell Key Persistence - PowerShell	multiple	ezvkta	DESKTOP-VT1A6VA	9044	View Command	No output
9/12/2025, 11:51:16 PM EDT	success	Identify local users	discovery	ezvkta	DESKTOP-VT1A6VA	5880	View Command	View Output
9/12/2025, 11:51:56 PM EDT	success	Zip a Folder with PowerShell for Staging in Temp	collection	ezvkta	DESKTOP-VT1A6VA	8312	View Command	No output
9/12/2025, 11:52:51 PM EDT	success	Exfiltrating Hex Encoded Data Chunks over HTTP	exfiltration	ezvkta	DESKTOP-VT1A6VA	4104	View Command	View Output

Figure 5.11 Shows operation phase successfully created and executed

Step 9: Ex-filtrated file received in the Web-server and downloaded on attacker machine

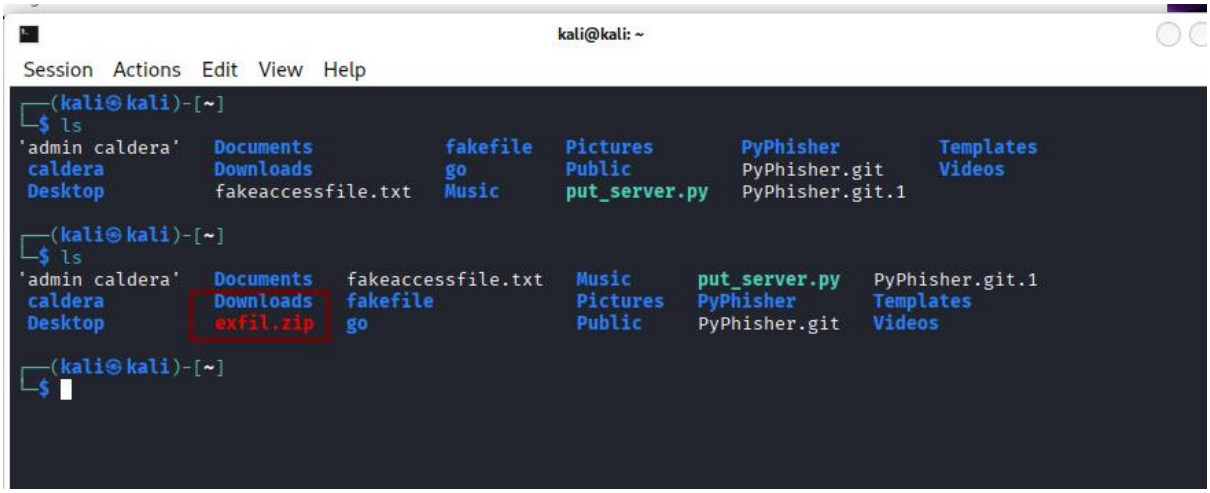


Figure 5.12 Shows exfil data successfully received on attacker machine

Step 10: Once all the operations are run successfully , go to temp folder find event logs ,here all the caldera logs are saved.

Figure 5.13 Shows caldera logs

6. Technique Mapping

Phase	*Observed technique*	*MITRE ATT&CK ID*
Delivery	Macro-enabled phishing attachment served via HTTP	T1204.002 (User Execution: Malicious File)
Initial Access (exploit/macro exec)	Macro triggers PowerShell / SANDCAT download & execution	T1204 (User Execution) / T1190 (Exploit Public-Facing App) context dependent
Execution	PowerShell execution of download/stager	T1059.001 (cmd/PowerShell)
Discovery	(implicit) enumerate user files ($env:USERPROFILE\Downloads)	T1083 (File and Directory Discovery)
Collection / Staging	Compress-Archive to create exfil.zip	T1005 (Data from Local System) / T1560 (Archive Collected Data)
Exfiltration	HTTP PUT upload or hex-chunk POST to attacker web listener	T1567 (Exfiltration Over Web Service) / T1041 (Exfil over C2)

Table 6.1 Shows technique mapping