

Using Cloud Analytics with Caldera

Using the emu plugin for Caldera makes it easy, but it is not obvious how it works. The high level process is as follows:

NOTE: Caldera 4.0.0-beta was used for the following instructions. Not tested on other versions.

- Using Cloud Analytics with Caldera
- Initial Setup
 - Setup Caldera Server
 - Setup Windows Guest
 - Add Windows VM as Caldera Agent
- Post-Install Setup
 - Ensure EMU Plugin is Enabled
 - Install New Adversary Emulation Plan
 - Option A: Install From AEP Archive
 - Option B: Create Directory Layout Manually
 - Activate New Adversary Emulation Plan
- Validate
 - Adversary Profile
 - Fact Sources
- References

Initial Setup

Setup Caldera Server

NOTE: Assumes user has setup Vagrant and Virtualbox. Vagrant provides a straightforward workflow to create a reusable, repeatable environment shareable by vagrant users.

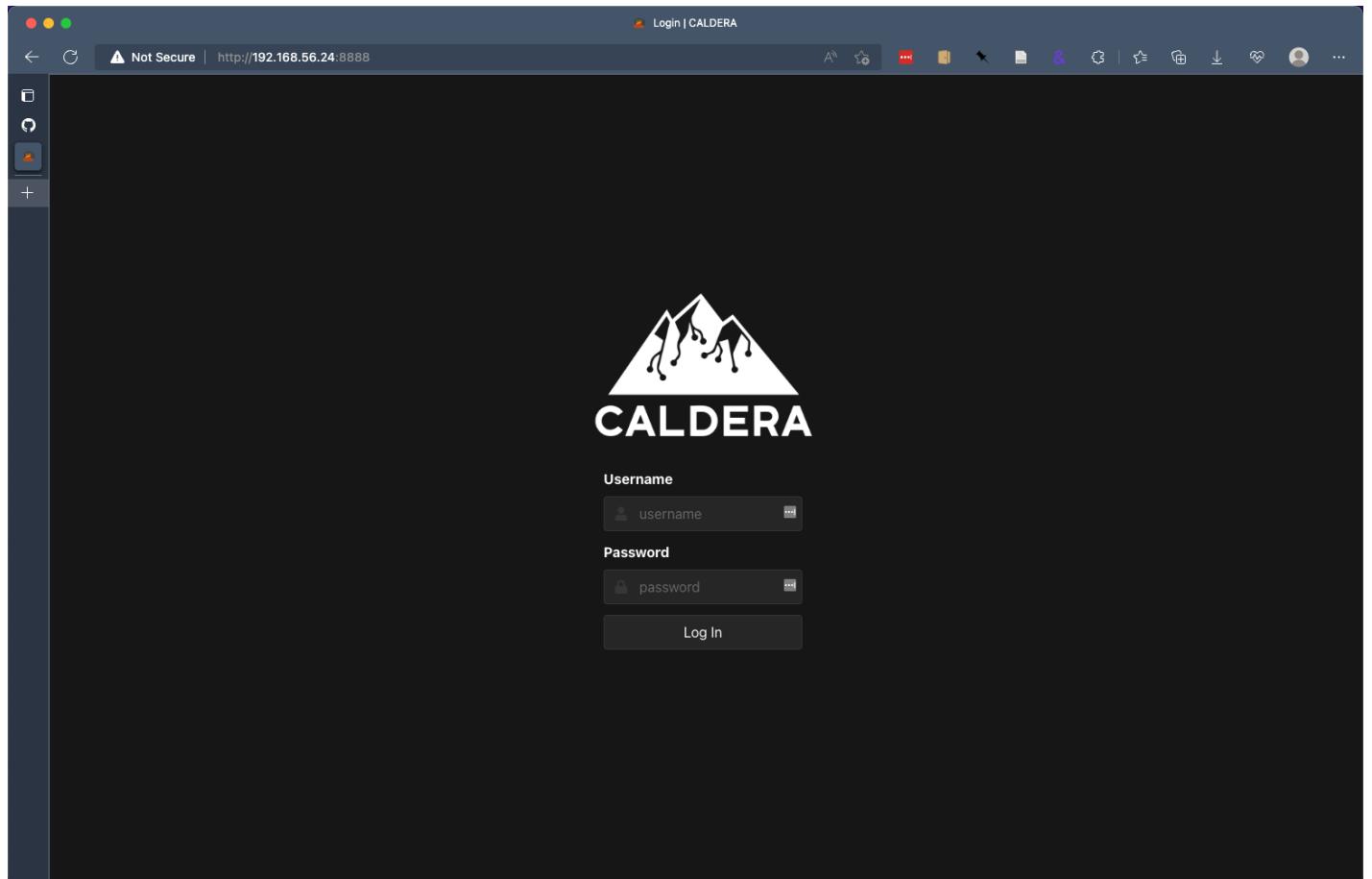
Within the `cloud-analytics/emulation/caldera-vagrant/` directory, of the [Cloud Analytics](#) project, there is a vagrant configuration which will install Caldera from scratch on a new virtual machine instance.

1. Open a terminal window, and change to the `caldera-vagrant` directory: `cd ./cloud-analytics/emulation/caldera-vagrant/`
2. Run `vagrant up` to initialize the vagrant environment. How long this takes is highly dependent on your network connection. Vagrant will first perform a one-time download of the base box, `ubuntu/focal64`, and then provision the VM by installing and configuring Caldera.
3. Once Caldera is fully provisioned, you should see a banner similar to the following, with a URL to connect to the Caldera web interface.
4. **NOTE:** Due to a quirk in the Caldera 4.x beta, after vagrant is complete and the system boots up, you should wait approximately 3 minutes, then run `vagrant reload` from your host system to restart. Otherwise, Caldera may hang on plugin initialization and not fully startup the web interface. After waiting a few minutes and running

vagrant reload , Caldera should properly start on all startups going forward. If you encounter a ERR_CONNECTION_REFUSED in your browser, you have encountered this issue. Just run vagrant reload and the problem should be permanently fixed going forward.

```
```
#####
#
#
#
#
Connect to Caldera at http://192.168.56.24:8888
#
#
#
#
#####
```
```

5. Open a web browser and connect to Caldera at the URL specified in the terminal, as shown in the previous step. You should see a login screen similar to the following.



- 6.
7. Login with the default credentials

8. username: red
 password: admin

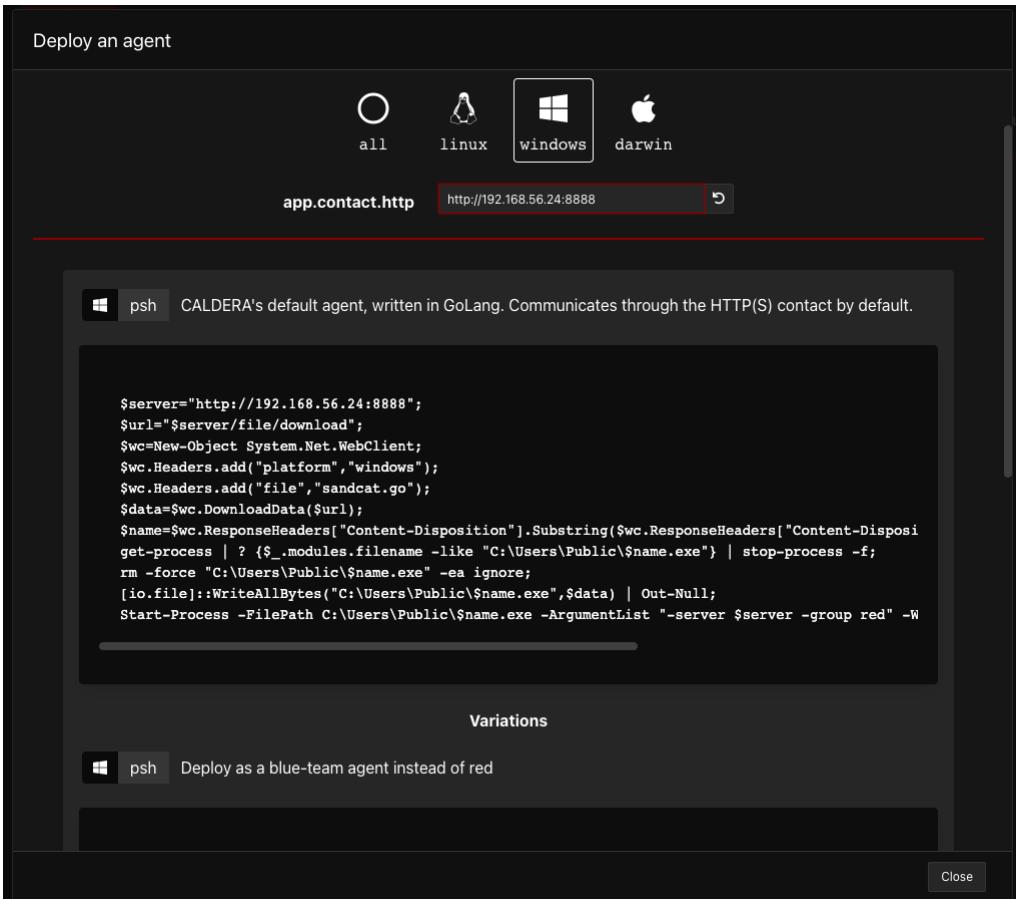
Setup Windows Guest

The Caldera server application will act as the emulation controller, however we need *Caldera Agents* to perform the actual executions. In this example, we will deploy a Windows VM using Vagrant. The Windows instance will use a temporary evaluation license by default. Make sure this meets your organizational licensing requirements or install an appropriate license as needed.

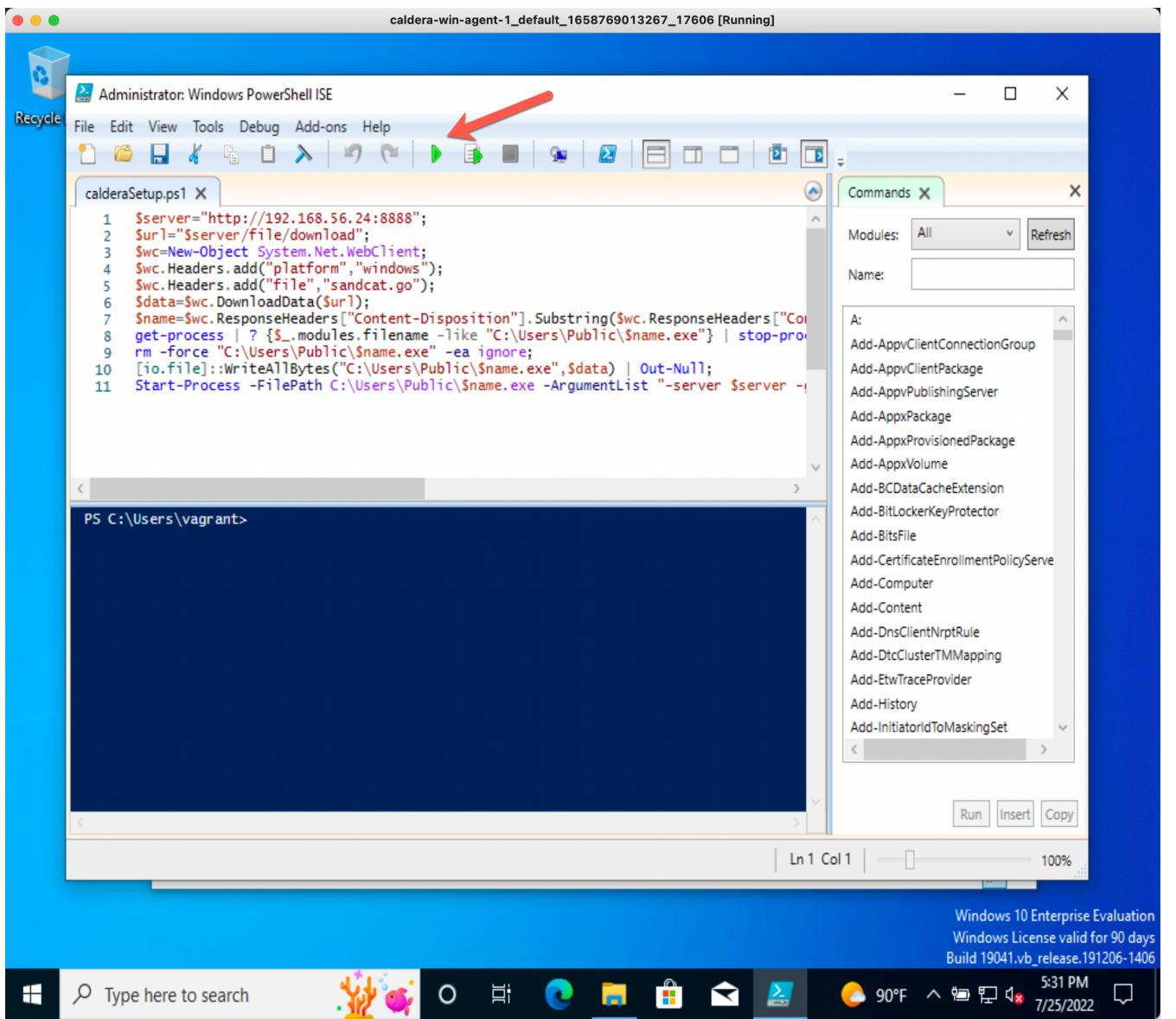
1. Open a separate terminal, and navigate to the `cloud-analytics/emulation/caldera-win-agent-1` directory.
2. Run `vagrant up`
3. After the Windows system is fully booted, continue with the next section to add the Windows system as a Caldera agent.

Add Windows VM as Caldera Agent

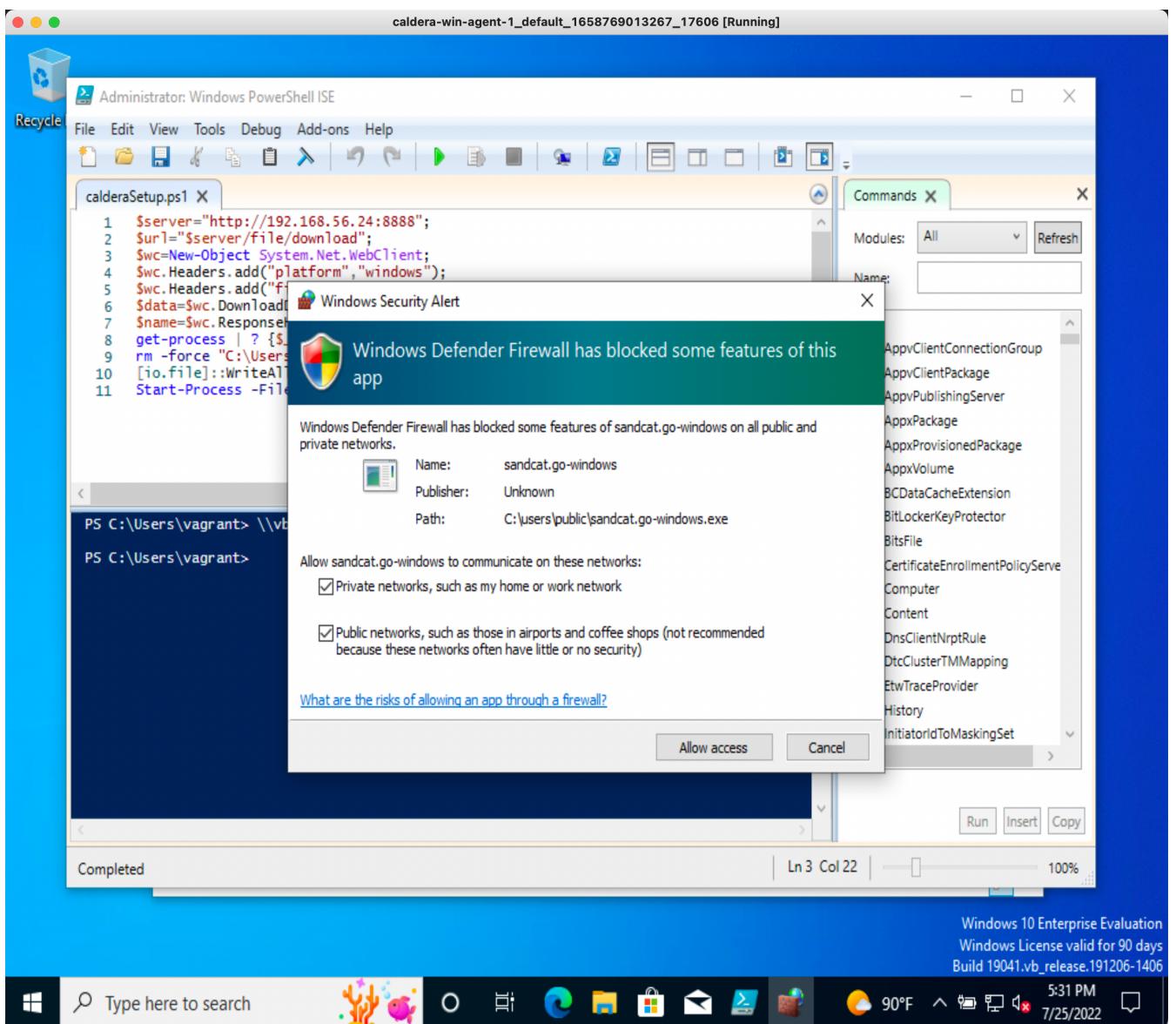
1. Within Caldera, navigate to the `Campaigns -> agents` section.
2. Click the `Deploy an Agent` button.
3. Select `Sandcat` from the dropdown menu.
4. Select `windows` under `Platform`.
5. Edit the `app.contact.http` setting from `http://0.0.0.0:8888` to the URL printed out earlier on the terminal when Caldera started up. In the earlier example, the URL is `http://192.168.56.24:8888`.
- 6.
7. Copy the PowerShell code from the first section, with the title of `CALDERA's default agent, written in GoLang`.
8. Create a new file in the `caldera-win-agent-1` directory, titled `calderaSetup.ps1`. That directory should now have two files, `Vagrantfile` and `calderaSetup.ps1`.
9. Open a GUI console session to the Windows VM.
10. Open the Virtualbox application.
11. Look in the list of VMs for a name that begins with `caldera-win-agent-1-`. Vagrant appends additional characters to the name, but you only have to match the initial section.
12. Select the VM on the left with a single click.
13. Click the green `Show` button in the toolbar in the top right.



- i. 14. You should be logged in to a Windows VM. Use the Virtualbox -> View menu if you need to modify the display settings.
15. If needed, the default Windows username and password are vagrant and vagrant , respectively.
16. From within the Windows guest, from the Start Menu, open the Windows Powershell ISE application **as an Administrator**.
17. Select File -> Open from the menu, and navigate to C:\vagrant\ .
18. Open the file calderaSetup.ps1 .
19. Click the Play icon to run the script.



20. Windows Firewall will generate a notification due to the network access. For the Windows Firewall prompt, check both boxes and click Allow Access .

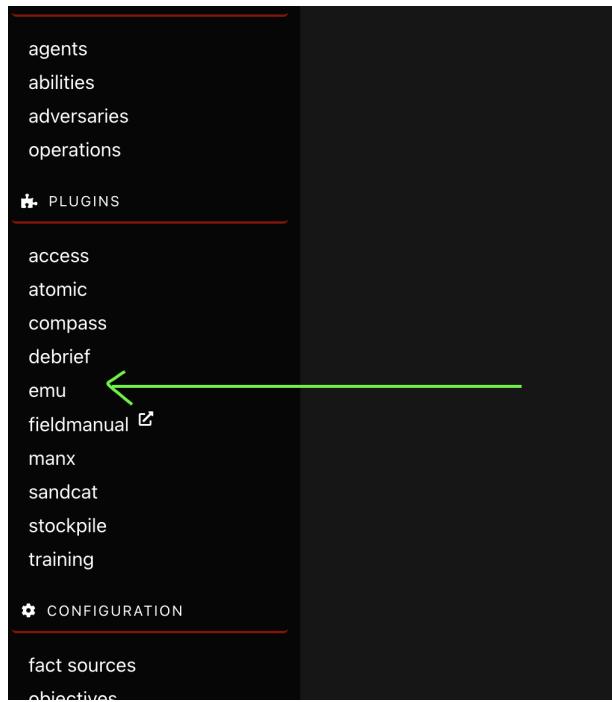


21. Navigate back to the Caldera agents webpage on your host computer, and the new Windows agent should show in the Agents list.

Post-Install Setup

Ensure EMU Plugin is Enabled

Make sure the emu plugin is enabled within Caldera. If not, navigate to *Configuration -> configuration-> Plugins*, and enable the emu plugin, and restart Caldera. When enabled, you should see `emu` on the left side menu.



Install New Adversary Emulation Plan

NOTE: If you are using the Vagrant Caldera setup installed earlier, read the following:

- You can ssh to the Caldera instance by `cd cloud-analytics/emulation/caldera-vagrant`, then running `vagrant ssh`.
- To copy the adversary emulation plan, copy the `aep1-package-caldera.tar.gz` package to the vagrant directory. For example, `cp cloud-analytics/emulation/aep1-package-caldera.tar.gz cloud-analytics/emulation/caldera-vagrant/`.

Option A: Install From AEP Archive

1. On the command line on the Caldera system, navigate to the following directory (`CALDERA_HOME` denotes the home directory of the Caldera installation).
2. `cd CALDERA_HOME/plugins/emu/data/adversary-emulation-plans`
3. Copy the attached file to the Caldera system, and decompress while in the directory in the previous step.
4. `tar -zvxf /path/to/aep1-package-caldera.tar.gz`
5. Vagrant users: If you followed the steps at the beginning of this section, you can run `tar -zvxf /vagrant/aep1-package-caldera.tar.gz`.
6. The resulting directory layout should look similar to the following:

```
vagrant@ubuntu-focal:~/caldera/plugins/emu/data/adversary-emulation-plans$ tree aep1/
aep1/
└── Emulation_Plan
    └── yaml
        └── aep1.yaml
.
2 directories, 1 file
```

Option B: Create Directory Layout Manually

Alternatively, you can manually recreate the same structure.

1. cd CALDERA_HOME/plugins/emu/data/adversary-emulation-plans
2. mkdir -p aep1/Emulation_Plan/yaml/
3. cp /path/to/aep1.yaml ./aep1/Emulation_Plan/yaml/

Activate New Adversary Emulation Plan

After completing one of the above versions, restart Caldera.

Validate

NOTE: The Cloud Analytics adversary name is currently CAP, short for Cloud Analytics Project.

Adversary Profile

Within Caldera, *Adversary Profiles* allow for collecting ATT&CK TTPs for a specific effect or scenario, such as an offensive or defensive scenario.

To validate the CAP profile is setup, within the Caldera web interface, navigate to *Plugins -> emu -> Adversaries -> Select a profile -> CAP*.

A screen similar to the following should be displayed.

Adversary Profiles

Adversary Profiles are collections of ATT&CK TTPs, designed to create specific effects on a host or network. Profiles can be used for offensive or defensive use cases.

Select a profile

CAP



+ New

CAP

Adversary CAP, or Cloud Analytics Project, is a simulated adversary created to meet the goals of the MITRE Engenuity Cloud Analytics Project. Adversary behavior was based off of TeamTNT, and augmented with additional TTPs to provide a cloud specific adversary. Requires Azure CLI installation. (Emu)

				Objective: default	Change	Save Profile	Delete Profile					
Ordering	Name	Tactic	Technique	Executors	Requires	Unlocks	Payload	Cleanup				
≡ 1	Initial Access Login to Cloud	initial-access	Valid Accounts	Windows					x			
≡ 2	Initial Access External Remote Services	initial-access	External Remote Services	Windows					x			
≡ 3	Initial Access Valid Account	initial-access	Valid Accounts	Windows					x			
≡ 4	Persistence	persistence	Implant Internal Image	Windows					x			
≡ 5	Spawn container	defensive-evasion	Defense Evasion - Deploy Container	Windows					x			
≡ 6	Extract Metadata	credential-access	Credential Access - Cloud Instance Metadata API	Windows					x			
≡ 7	Discovery Groups	discovery	Permission Groups Discovery - Cloud Groups	Windows					x			
≡ 8	Discovery Container and Resources	discovery	Permission Groups Discovery - Cloud Groups	Windows					x			
≡ 9	Lateral Movement	exfiltrate	Taint Shared Content	Windows					x			

Fact Sources

Within Caldera, *Fact Sources* allow for using variables within an execution plan. Multiple fact source configurations can be setup for a profile, such as a fact source for the test environment. Along with Adversary Profiles, Fact Sources allow for executing predefined scenarios customized to a particular environment.

To validate the CAP Adversary Fact Source has been setup, within the Caldera web interface, navigate to *Configuration* -> *fact sources* -> *Select a source* -> **CAP**.

A screen similar to the following should be displayed.

emu ✕ adversaries ✕ fact sources ✕

Fact Sources

Facts are identifiable pieces of data, collected by agents or loaded when the server starts. A source is a collection of facts. Rules are boundaries to ensure specific traits cannot be used.

Select a source		CAP (Emu)	+ Create Source
CAP (Emu)			Duplicate Delete Source
Facts (15) ▾			
+ Add Fact <input type="text"/> Find a trait or value...			
Order	Origin	Fact Trait	Value
0	IMPORTED	identity.app.id	unknown
1	IMPORTED	identity.app.pw	unknown
2	IMPORTED	identity.tenant	unknown
3	IMPORTED	cloud.resource.group	casandbox
4	IMPORTED	container.name	psdocker
5	IMPORTED	container.registry	justsomeregistry
6	IMPORTED	container.upstream.image	mcr.microsoft.com/powershell:latest
7	IMPORTED	container.upstream.image_b	docker.io/library/hello-world:latest
8	IMPORTED	container.upstream.image_b.tag	latest
9	IMPORTED	cloud.location	eastus
10	IMPORTED	container.image	justsomeregistry.azurecr.io/targetrepository:tar gettag
11	IMPORTED	vm.name	ca-linux-vm
12	IMPORTED	acr.username	justsomeregistry
13	IMPORTED	acr.password	secretpassword
14	IMPORTED	container.name_b	publicapp

References

- Official Caldera documentation: <https://caldera.mitre.org/>