CALDERA is an automated adversary emulation system produced by the MITRE Corporation.
CHAPTER 1

Getting Started

**CALDERA** is an automated cyber adversary emulation system. **CALDERA** can be used to simulate the post compromise behavior of a cyber adversary on a Windows Enterprise network.

If you want to dive right in, start with the *Requirements* and then read *Installation*.

If building a custom test environment for **CALDERA**, read *Setting up a Network Environment* to see how to set up a network for **CALDERA** to run in.

After installing **CALDERA**, read *Running Your First Operation* to see how to run your first Adversary Emulation operation.

We also recommend taking a look at **CALDERA’s Philosophy** to understand why **CALDERA** was designed.

If you run into trouble, *Configuration* may have settings that you need to tweak to get **CALDERA** to work in your environment.

If interested in the internal workings of **CALDERA** see the *Logic in CALDERA* and *API Information*.

And finally, for information on how to add new adversary techniques to **CALDERA**, read *Adding New Adversary Techniques*. 
CALDERA is a tool that can perform automated adversarial assessments against Windows enterprise networks, requiring zero prior knowledge about the environment to run. CALDERA works by leveraging its built in semantic model for how Windows enterprise domains are structured, an adversary model describing an attacker’s goals and actions, and an artificially intelligent planner that makes decisions about which actions to perform. CALDERA does this all by performing real actions on systems similar to how an adversary would so that the same kinds of data gets generated: CALDERA features a remote access tool (RAT) that performs adversary actions on infected hosts and copies itself over the network to increase its foothold. To most realistically emulate an adversary, CALDERA’s model uses common Windows domain elements – users, shares, credentials – and features a library of executable techniques curated from ATT&CK, including favorites such as running Mimikatz to dump credentials and remote execution with WMI.

As a fully automated tool, defenders can use CALDERA to verify their defenses are working appropriately and as a resource to test defensive tools and analytics. Additionally, CALDERA’s modular design allows users to customize each individual operation and provides a flexible logic so that users can incorporate their own techniques into CALDERA’s automated assessments.
3.1 Operating System Requirements

CALDERA has both a Server and an Agent, the requirements for each are different. The agent is installed on each computer that will be involved in the emulation exercise. The agent supports 64 bit versions of Windows 7 and higher.

The server must be installed on a computer that is accessible over the network to all agents. The server will run on either Windows or Linux; however, it needs access to a compiled Windows binary, CraterMain.exe, so that it is able to serve this file to the agents. If running the server on Linux, CraterMain.exe can be built on a Windows machine and copied to the Linux machine, or a precompiled executable may be downloaded from here.

See Installation for information on installing the Server and Agents.

3.2 Network Requirements

CALDERA operates in a Windows Enterprise network environment. At a minimum this will contain a Domain Controller running Windows Server 2008 R2 through 2016 and two Windows Enterprise computers joined to that domain.

See Setting up a Network Environment for more information on creating a Windows Enterprise test environment.

3.3 Internet Requirements

An Internet connection is required to install CALDERA, however once installation is complete an Internet connection is no longer required. It is possible to install CALDERA without Internet by manually copying CALDERA’s dependencies to the install server. Contact CALDERA’s developers for more information.
This contains the installation instructions for CALDERA. CALDERA consists of three separate pieces of software:

**CALDERA server**  The server controls the execution of CALDERA and contains a web interface for administration

**CALDERA agent**  A Windows service that communicates to the CALDERA server, the CALDERA Agent is installed on every computer partaking in adversary emulation activities.

**Crater**  A Windows executable that is used as an implant for Adversary Emulation exercises

These instructions have two sections: the first section dictates how to install the CALDERA Server along with Crater. The second section details how to install the CALDERA Agent on each system taking part in the red team.

### 4.1 CALDERA Server Installation

The CALDERA server is installed on a single central server. It should be accessible over the network to all computers that are taking part in the emulated adversary operation. Both Windows and Linux are supported, however installation on Windows requires extra installation steps.

Several options exist for installing the server.

#### 4.1.1 Option 1: Install with Docker Compose

The easiest way to install CALDERA is with Docker Compose. After installing Docker and Docker Compose, from the top-level directory simply run the following:

```
docker-compose up
```

This will start the Caldera server and an instance of MongoDB. Follow the Post-install steps to login to the Caldera server and change the Administrator password.

For advanced options, customize the `docker-compose.yml` file. If you are behind a proxy, uncomment three lines in the `build` section and edit the lines to include the correct information for your proxy. If you need to specify a different version of Crater (for example, if you need the Windows 7 version), you can specify it with another item under
**4.1.2 Option 2: Building the Docker Container By Itself**

Alternatively, if you want to connect to an already existing MongoDB instance, you can build the server container by itself:

```bash
docker build . -t caldera
```

If you are behind a proxy, provide proxy information to the build process:

```bash
docker build . -t caldera --build-arg http_proxy=http://proxy.example:80 --
   --build-arg https_proxy=http://proxy.example:80
```

If you need to specify a different version of Crater (for example, if you need the Windows 7 version), you can specify it with another `--build-arg`:

```bash
docker build . -t caldera --build-arg http_proxy=http://proxy.example:80 --
   ->github.com/mitre/caldera-crater/releases/download/v0.1.0/CraterMain7.exe
```

Then follow the instructions above regarding MongoDB.

Next run the container:

```bash
docker run --net=host caldera
```

If you need to change any configuration parameters, for example to use a different port for MongoDB, you can do the following:

```bash
docker run --net=host caldera --database.port 27020
```

After installation, follow the *Post-install steps* to login to the Caldera server and change the Administrator password.

**4.1.3 Option 3: Installing Without Docker**

If you would like to install without docker, please follow the below instructions.

1. **Install Python 3.5.4 or later**  
   Python 3.5.4 or later can be acquired from the Operating System’s package manager or from https://www.python.org/. The standard installation is straightforward.

   **Note:** On Linux, the development package for Python is needed. For example (may vary based on distribution and version):

   ```bash
   apt-get install python3-dev
   ```

2. **Upgrade to setuptools 24.0 or later**  
   The python package, setuptools, version 24.0 or later must be installed.  
   Setuptools comes with Python 3.5, but some versions may not be up to date. To update it, run

   ```bash
   pip install --upgrade setuptools
   ```
3. [Windows Only] Install Visual C++ 2015 Build Tools  
Install the Visual C++ 2015 Build Tools. During install check Windows 8.1 SDK and Windows 10 SDK options.

**Note:** If Visual Studio 2015 is already installed, Visual C++ 2015 Build Tools should not need to be installed

4. **Install Python libraries**  
Within a command prompt navigate to the caldera/caldera folder and run the command

```
pip install -r requirements.txt
```

This will download and install the Python libraries necessary to run the CALDERA server.

**Note:** When sitting behind a proxy, specific steps must be taken for pip to understand proxies. See Setting your pip configuration file. An example configuration file might look like the following:

```yaml
[global]
  proxy = http://my.proxy.name
  cert = C:\My\Path\To\SSL Certificates\chain.pem
```

5. **Install MongoDB**  
MongoDB 3.0 and later are supported. Most Linux distributions have MongoDB in their package manager. Otherwise both Windows and Linux installers can be downloaded from https://www.mongodb.com/download-center#community

6. **MongoDB Configuration**  
MongoDB must be configured to use a replication set. There are two ways to do this, either by modifying MongoDB’s configuration file or by adding an additional command line flag when starting the MongoDB daemon. On Linux it is typically easier to edit the configuration file (typically located in `/etc/mongodb.conf`). On Windows it is typically easier to add an additional command line flag. Both methods are detailed below.

**Method 1: Edit Configuration File (Recommended for Linux)**

Depending on the version of MongoDB that you have installed, the configuration file uses two different formats. Old style formatting typically contains equal signs. If you see no equal signs you probably have a new style configuration file (See https://docs.mongodb.com/v3.2/administration/configuration/ for more information.)

After determining what style configuration file you have, make the following modifications to it:

Newer version of mongodb use YAML style formatting so the following lines should be added (indentation intended)

```
replication:
  replSetName: caldera
```

Older versions on mongodb use key value pairs. For this style the following line should be added

```
replSet = caldera
```

**Method 2: Command Line Flag (Recommended for Windows)**

Alternatively, the replication set parameter can also be passed in through the command line if running mongod from the command line by adding the flag `--replSet caldera` to the command to start mongod. This is the easiest way to configure replication sets for Windows installs.

7. **Start MongoDB**  
MongoDB must be started. If MongoDB was installed using the Operating System’s package manager, look for instructions on how to start the MongoDB service (typically
service mongod start). On Windows, MongoDB is installed by default in `C:\Program Files\MongoDB\Server\<version>\bin`. Navigate to this folder on a commandline and run `mongod.exe --bind_ip 127.0.0.1 --replSet caldera`

8. **[Optional] Install git**  
   Git can be installed for version tracking information. It is available from Linux distributions package maintainers or from [git](https://git-sCM.org/).

9. **Install CraterMain.exe**  
The `CraterMain.exe` binary needs to be accessible to CALDERA. It should be placed in: `caldera/dep/crater/crater/CraterMain.exe` on the computer that the CALDERA server is installed on. Pre-built copies of CraterMain.exe are available [here](https://caldera.org/).

10. **Start the CALDERA server**  
The Caldera server can now be started by navigating to the `caldera/caldera` directory and running `python caldera.py`. The first time CALDERA is run, it will generate an OpenSSL command line which can be run to create an SSL keypair for encrypted communication. If CALDERA is installed on a Linux machine, OpenSSL is typically already installed and available. On Windows computers, OpenSSL can be installed or, this command should be executed on a Linux computer (with the appropriate hostname substituted).

See the following section for login information.

### 4.1.4 Post-install steps

After installing, perform the below steps

1. **Login to the Caldera server**  
The CALDERA server exposes a web service available on [https://localhost:8888](https://localhost:8888). Navigate to this URL to access CALDERA’s administration panel. The default username and password

   ```
   username: admin  
   password: caldera
   ```

2. **Change the Administrator Password**  
   Click on the top right button labeled “admin (Admin)” and select “Change Password” to change the default password to something unique and secret.

CALDERA is now installed. Proceed to the next section to install CALDERA Agents.

### 4.2 CALDERA Agent Installation

The CALDERA Agent or cagent, is installed on every computer participating in the Adversary Emulation. It should be accessible over the network to the CALDERA server. Once configured, each cagent will register with the CALDERA server making its computer available as an option in an operation. Pre-built cagent binaries are available [here](https://caldera.org/).

#### 4.2.1 Operating System Support

*Windows 7, 8, 8.1 or 10, 64 bit*  
A 64 bit version of Windows 7, 8, 8.1 or 10 is required.

#### 4.2.2 Installation Instructions

1. If not already done, install the CALDERA server

2. Install the [Visual C++ Redistributable for Visual Studio 2015](https://visualstudio.microsoft.com/downloads/)
Note: The Visual C++ Redistributable may fail to install if Windows is not fully updated. If you encounter problems try fully updating Windows.

3. Download the latest release of cagent. Place cagent.exe in the desired installation location (the recommended location is C:\Program Files\cagent\cagent.exe)

4. In the same directory, place the conf.yml file which can be downloaded from the CALDERA server by navigating to

https://my-caldera-server:8888/conf.yml

Note: The conf.yml is unique to the CALDERA server. When migrating agents to a new server, you will have to update the conf.yml file

Warning: To prevent unauthorized users from modifying cagent.exe or conf.yml ensure the directory that contains these files is only editable by Administrators

5. In an Administrator command prompt install cagent with:

cagent.exe --startup auto install

6. In an Administrator command prompt start cagent with:

cagent.exe start

Agents that are connected to the CALDERA server are visible by checking the Debug>Connected Agents tab.
This section covers how to create a test environment for CALDERA to run in. Although CALDERA is designed to run in real Enterprise environments, it is possible to construct a lab environment to test CALDERA in.

Note: If CALDERA is being installed on a pre-existing network this section can be skipped

In this section we’ll also talk specifically about the techniques that adversaries use within Windows Networks.

## 5.1 Creating a Windows Enterprise Network

Windows Enterprise Networks (also called Windows Domains) are common in organizations. They allow network administrators to easily oversee large amounts of Windows computers, and let computers that are part of the domain share resources including user accounts, files and printers.

A minimal Windows Enterprise network consists of a Windows Server acting as a Domain Controller and two Windows Enterprise computers acting as Domain members.

Creating and configuring a Windows Domain is outside the scope of this documentation, but several instructions exist online for how to configure a Windows Server as a Domain Controller and how to join workstations to that Domain.

## 5.2 Understanding Credential Overlap

Many adversarial techniques, and thus those built into CALDERA, require credential overlap in the targeted Windows network. Credential overlap is a condition that’s common in Windows Enterprise networks and involves the following things interacting together:

1. A weakness in Windows that allows an adversary to recover the credentials of all users that have logged into a Windows computer since its last reboot
2. A Windows feature that allows users that are administrators to remotely interact with Windows computers (for example, to remotely run a process or copy a file)
3. A common issue in real world networks where the same accounts are administrators on many (or all) computers in the network
4. A tendency for people to log onto many different computers in the network without rebooting (and therefore make it easy to find their passwords in memory)

What all that means is that when an attacker gets on a Windows network, their high level courses of actions are:
1. Recover the passwords of all the users that have logged onto the system
2. See where those users are administrators
3. Use their administrative privileges to laterally move to new computers
4. See if the attacker’s current level of access accomplishes their goal (compromise a specific user, find some kind of documents to exfiltrate, or something else)
5. If their goal is still not accomplished, repeat

5.3 Creating a Network with Credential Overlap

Believe it or not, in many real networks this strategy is successful. But it takes some setup to make a test network behave this way. Luckily there are only a few things you need to do to emulate this kind of situation:
1. Create a domain user account (different from a local user account)
2. Make that user an administrator on both Windows 10 computers in your domain (the account doesn’t have to be an administrator on the Domain Controller, but it can be)
3. Log the user into one of the Windows 10 computers to place their password in memory

Steps 1 and 2 are permanent changes that only have to be done once whereas Step 3 must be done any time the computer is rebooted.

5.4 Considerations for Windows 8 and Above

Windows 8 and above contain some differences from Windows 7. On those systems, several commands must be run on each computer taking part in the operation to make configuration changes that reflect enterprise networks.

**Warning:** The below firewall and registry changes are intended to make a test network appear more like a real network. They should not be executed on a real network because they will weaken the security of that network

1. **Enable Firewall Rules** Several Firewall rules must be enabled to allow CALDERA techniques to operate correctly. The following commands modify the Windows Firewall to allow this traffic

   ```powershell
   powershell -command Enable-NetFirewallRule -DisplayName "'File and Printer Sharing (SMB-In)'"
   powershell -command Enable-NetFirewallRule -DisplayName "'Remote Scheduled Tasks Management (RPC)'"
   ```

2. WDigest caching must be enabled for mimikatz to detect plaintext credentials.

   ```reg add hklm\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\ /v UseLogonCredential /t REG_DWORD /d 1```
This section will show how to get your first operation up and running. An operation is the basic unit of Adversary Emulation.

**Note:** If you haven’t already done so, follow the instructions in the *Installation* section before proceeding.

If the **CALDERA** server isn’t running, start it by navigating to the caldera/ directory and executing

```
python caldera.py
```

Then navigate to the **CALDERA** interface at https://localhost:8888

The default user account is *admin* and the default password is *caldera*

If you’ve followed the *Installation* instructions, you should have some agents connected, check this by going to the *Debug > Connected Agents* tab. If no agents are connected, go back to the *Installation* instructions to ensure that the Agents have been properly installed.

## 6.1 Creating an Adversary

To perform an Operation, **CALDERA** needs an Adversary to emulate. In **CALDERA**, an Adversary represents a real adversary’s tactics and techniques. When we create our operation we will select an Adversary to use which will dictate what techniques **CALDERA** performs during the operation.

We will create a simple Adversary.

Click on *Threat > Create Adversary*. This will take you to the adversary creation page. Several fields exist for you to configure.

**Name** This is the name that you would like to give to the adversary, for this example name it whatever you would like

**Steps** These are the atomic actions that the adversary is allowed to perform. Steps are the main way in which you can change the behavior of your adversary. Each step gives the adversary new abilities to perform. Steps have a
name and additionally are listed with their ATT&CK ID and ATT&CK Tactics. A detailed listing of each step is available by going to Threat > Configure Steps. For this example, select the following steps from the list:

- copy_file: [T1105, Lateral Movement]
- get_creds: [T1003, Credential Access]
- get_admin: [T1086, Execution | T1069 & T1087, Discovery]
- get_computers: [T1086, Execution | T1018, Discovery]
- get_domain: [T1016, Discovery]
- net_use: [T1077, Lateral Movement]
- remote_process(WMI): [T1047, Execution]

Artifact Lists  Artifact lists allow you to customize the artifacts that your adversary leaves behind. These artifacts include things like file and service names. For now we will leave this field blank to use the default artifact list.

Exfil Method

**CALDERA supports exfiltration techniques. This field defines how the adversary will exfiltrate data.**

- raw_tcp - exfiltrate using a tcp based protocol
- http - exfiltrate using an http protocol
- https - exfiltrate using an https protocol

**Exfil Address**  This is the IP address that CALDERA will exfiltrate data to. Leaving it set as x.x.x.x will automatically use CALDERA’s runtime IP address automatically. Leave this set to x.x.x.x for now.

**Exfil Port**  This is the TCP port that CALDERA will exfiltrate to. Leave this set to its default.

After completing this, click “Submit”. You can now see the newly created Adversary. If you made a mistake click the small pencil icon in the top right to edit the adversary.

### 6.2 Creating a Network

Now we will create a network for the adversary to operate on. Networks are just collections of host. They are a simple way for CALDERA to organize and group together computers.

Navigate to Networks > Create Network. You will be brought to the adversary creation page. There are a few fields for you to configure.

**Name**  You can name the network you are creating. For this example give the network any name you would like

**Domain**  This will let you select the domain name of the computers that you would like to include in this exercise. Every computer in a Network must be from the same domain. For this example select whichever domain is available.

**Hosts**  Once you have selected your Domain, the Hosts field will populate. As Agents register with the CALDERA server, they are added to this field. Select as many hosts as you would like to participate in this game.

Click “Submit” at the bottom of this page to create the Network.

### 6.3 Create an Operation

Now that we have created both an Adversary and a Network, we can create our first operation.

Go to Operations > Create Operation.
This will take you to the Operation creation page, where there are a lot of values to customize. We will explain them all but most of them are advanced and we won’t need to configure them.

**Name** You can give the Operation a name so that you can remember it.

**Adversary** You must select an Adversary which will be the aggressor in this Operation. Pick the adversary that you created earlier.

**Network** You must select a network for this Operation which will limit the scope of the operation to the hosts contained in the Network you choose. Select the Network you created earlier.

**Starting Host** This is the first host that the Operation will start on. Select from one of the options.

**Start Method** This option lets you configure how the initial RAT will be created. Because CALDERA assumes that a network has already been compromised, CALDERA begins with a RAT running on the starting host. This field lets you configure how that RAT is created.

- Existing Rat - If a RAT is already connected to CALDERA, you can use it as the starting RAT. If you select this option, an additional field will appear called “Starting Rat” that will allow you to pick the RAT you would like to start with.
- Wait For New Rat - If you would like to launch the RAT manually, you may select this option to have CALDERA wait for a Rat to connect
- Bootstrap Rat - CALDERA can automatically start a RAT on the starting host for you. Select this option.

**Start Path** You may tell CALDERA where you would like the Rat’s executable file to be placed. Leave this empty to use the default location.

**Starting User** The Rat can be started in several different user contexts. This field lets you select the user context you would like the Rat to start as.

- System - This is the System account. Leave this option selected.
- Active User - This will start the Rat as the user account of whoever is logged in. If you select this option you will see a field called “Parent Process” appear, which will let you enter a process to use as the parent for the rat
- Logon User - This will let you enter a specific user account to use. If you select this option, two fields will appear for you to enter the user name and password of the account that you would like the Rat to run under.

**Auto-Cleanup** CALDERA has the ability to cleanup after itself at the end of an operation. With a few exceptions, every technique that CALDERA executes can be cleaned up. Checking this box will perform the cleanup automatically when CALDERA is finished with the operation. Unchecking this box will allow you to manually trigger cleanup after the operation is over.

**Command Delay (ms) & Command Jitter (ms)** CALDERA typically runs techniques very quickly. If you would like to introduce some variability in how quickly CALDERA operates, you can artificially slow it down by adding delay and jitter. Every time CALDERA tries to execute a command, delay and jitter are used to calculate a sleep function. The value for this sleep is defined by the expression:

\[
\text{delay} + \text{random}(-\text{jitter}, \text{jitter})
\]

where the function \text{random} returns a random number between its first and second parameter. For now leave delay and jitter set to 0.

**Clone Previous Operation** At the bottom of the screen you can also see that there is an option to clone a previous operation. This lets you quickly copy the settings from a previous Operation.

Now that you have configured the Operation select, “Submit”. 

6.3. Create an Operation
Warning: Creating an Operation will immediately start running it.

This will create and start the operation, and take you to the operation view.

6.4 Observing an Operation

In the Operation view, you can view the progress that CALDERA has made working on an operation.

The operation’s status is displayed at the top of the screen next to the Operation’s name.

Below the status, colored bubbles indicate the number of hosts and credentials that have been compromised during this operation.

The bubbles on the left indicate CALDERA’s progress in compromising the network. Each bubble represents a host within the network. Bubbles start out gray. When CALDERA discovers a host, that host’s bubble turns blue. When CALDERA gets a Rat on a host, that host turns red.

On the right is a pane called “Operation Details”. This has several tabs that let you explore the operation.

**Steps** The Steps tab shows all of the steps that have been executed. Clicking on a step will expand the step to show the exact commands that CALDERA executed.

**Jobs** The Jobs tab is used for debugging

**Artifacts** The Artifacts tab lists artifacts that CALDERA creates on the network. At the moment only files that CALDERA creates are listed here

**Cleanup Log** The cleanup log lists any errors that occurred during operation cleanup.
CALDERA’s Philosophy

CALDERA is an automated adversary emulation system produced by the MITRE Corporation. It is designed to operate on Windows Enterprise networks.

7.1 Adversary Emulation

Adversary Emulation is a branch of red teaming. The purpose of red teaming is to approach a problem or system as an adversary would, with the mindset of breaking the system, abusing it, or otherwise maliciously interfering with it. The practice of red teaming is commonly used as a method to test and improve the security of a system by iteratively breaking a system as an attacker would and mitigating those attacks. Red teaming is often focused on demonstrating the risk of an adversary’s impact to an organization. Engagements may take weeks or months and one of the goals of the red team is to not be detected.

Adversary Emulation is at its core red teaming, but rather than using the general mindset of a attacker, adversary emulation adopts the methodologies of a specific real-world adversary, complete with the adversary’s goals, methods, and techniques based on threat intelligence on how that adversary is known to operate. The focus of engagements is on having the emulation team and the defenders work together to improve the systems, network, and defensive processes to better detect the techniques used across the adversary’s lifecycle. Adversary Emulation grounds the process of red teaming by focusing on threats that are demonstratively real in a way that defensive improvements can be measured and verified.

7.2 Why Adversary Emulation?

The practice of emulating adversaries is designed to answer the question: “Is my network secure?”. Or more specifically: “Is my network secure against known threats?”.

As with all things, the best way to find out something is to test it. The same applies to networks: the best way to find out if a network is resilient to adversary attack is to actually enact adversary actions on that network and observe how the network responds.
7.3 Automating Adversary Emulation

CALDERA automates adversary emulation. CALDERA contains numerous built-in adversary techniques that are derived from ATT&CK. For each adversary technique, CALDERA contains a logical encoding that describes that technique’s requirements (preconditions) and the effects of the techniques (postconditions). CALDERA uses this information to figure out when and how to execute the actions that it is told to use.

7.4 Post Compromise

CALDERA is focused on adversary emulation “post compromise”. In other words, CALDERA assumes that an adversary already has an initial foothold on a network. CALDERA emulates adversary actions that occur after this point of initial compromise. This concept of “post compromise” has several important implications:

CALDERA doesn’t focus on the things an adversary will do to “get in”. Things like vulnerability scanning, penetration testing, intelligence gathering, and spearphising, that commonly occur as a precursor to an attack, are out of scope for CALDERA.

CALDERA’s behavior reflects what attackers do after initially compromising a network, which is significantly different from how they behave before compromising a network. For many, the mental model of how computer networks are compromised usually involves executing a string of exploits against systems to penetrate into a network. However, in reality, exploits are rarely used post compromise. Instead, attackers leverage built-in functionality and tools of the network. This means that unlike other automation systems CALDERA places a heavy emphasis on using and abusing these same constructs.

In assuming that an attacker has already compromised a network, CALDERA exercises areas of defenses that are commonly weak and untested within networks. Significant emphasis is usually placed on perimeter defenses (things like firewalls, boundary packet inspection, and maintaining DMZ patch levels) at the expense of post compromise defenses. In other words, defenses meant to prevent initial compromise are heavily stressed, often at the expense of defenses designed to prevent or detect post-compromise activity. CALDERA’s post compromise focus means that it tests areas of security that are typically neglected.
CALDERA is designed to be completely automated. In creating a completely automated system, there is a natural tension between letting the system decide what to do, and telling the system what to do. The design of CALDERA emphasizes the former: CALDERA decides for itself what to do.

CALDERA makes this decision based on an internal heuristic that grades possible courses of action. CALDERA can be customized by modifying this internal heuristic. At the moment this can only be done by editing the source code. CALDERA’s heuristic is straightforward, each step has a numeric score called a value. Higher values indicate a higher precedence step. Steps with higher precedence are prioritized over steps with lower precedence.

CALDERA’s built-in step values can be modified by editing the file /caldera/caldera/app/operation/operation_steps.py.
Chapter 9

Logic in CALDERA

Note: This section is subject to heavy development and likely to change in future versions of CALDERA

Logic is a central part of how CALDERA is able to operate automatically.

Every Adversary action, called a Step in CALDERA, contains a logical description of the Step’s requirements and effects. CALDERA parses these logical descriptions to both tell when it is possible to run a Step and to predict the outcome of a Step. This lets CALDERA generate plans by iteratively checking what Steps are executable given the current state, selecting a Step, and then generating the output state of that step, all according to the logical rules.

CALDERA’s behavior is based on the branch of Artificial Intelligence called Planning. An introduction to Planning is available on Wikipedia.
Limitations

CALDERA is a working prototype, several decisions have been made to limit its scope. This section details several of these decisions.

No Command and Control Emulation  We decided not to emulate Command and Control (C2) channels. This may seem like a glaring omission, but it was made for several reasons. For one, several tools already exist for simulating C2 network traffic. We felt that we could make a greater impact by focusing on other aspects of emulation, such as generating host-based artifacts.

From a practical standpoint, CALDERA was originally created to test host-based defenses and sensors. For this use, C2 emulation activity was unnecessary since host-based defenses mainly use activity on the host and not on the network.

From a philosophical standpoint, an adversary’s Command and Control protocol is easy to change and has a multitude of variations. Due to the wide variation in possibilities we thought our time would be better spent emulating other aspects of adversary behavior.

No Linux Support  CALDERA only supports Windows Environments, Windows is nearly ubiquitous in corporate and government environments. Furthermore, most publicly adversary reports detailed techniques used on Windows, which are very specific to given operating systems. Despite this, more information is becoming available on adversary behavior on Linux systems. We would like to eventually add Linux support to CALDERA.
11.1 Caldera Server Configuration

The server configuration is located in `caldera/caldera/conf/settings.yaml`. It will be created after the Caldera server is run once.

**Configurable options include:**

- the IP and port that the server will bind to
- the SSL certificate and key that the server will use
- the default username and password which is used to login to the web admin panel
- the SSL certificate and proxy needed to communicate with external sites

The server uses an SSL certificate and private key. Instructions to generate an SSL certificate and key are printed to the console if no SSL certificate is detected. The certificate and key location is configured within the settings.yaml file by editing the `crypto:cert` and `crypto:key` variables.

The default username and password is stored in the configuration file. If this value is modified, it does not affect existing user accounts, which will still remain and must be manually modified through the Settings interface within the Administration Panel.

The proxy settings for the server can be found in the settings.yaml, under the `proxy:default:http` and `proxy:default:https` variables. If a CA ssl certificate is necessary, reference its location in the `proxy:default:cert` variable. In the event that a site Caldera reaches out to requires a unique cert or proxy, simply duplicate a set of variables under proxy, replacing default with the site’s base, and configure the variables appropriately.
11.2 Caldera Agent Configuration

11.2.1 Customizing the conf.yml

Settings for cagent are stored in a file `conf.yml`. The CALDERA server generates a `conf.yml` file that is generally correct, however it may have to be modified if the caldera server cannot detect certain settings.

**url_root**  The `url_root` field in `conf.yml` is the hostname of the caldera server that the agent will connect to.

**cert**  The `cert` field should match the public certificate that the caldera server is configured to use, which by default is located at `caldera/caldera/conf/cert.pem`. Note that because the certificate is a multiline string, there is a hanging indent in the example configuration file.

**verify_hostname**  The `verify_hostname` flag can be set to true. This will cause the Agent to verify that the hostname on the Server’s certificate matches the hostname of the Server that it is connecting to. If you are not sure whether the certificate contains the appropriate hostname or you encounter problems connecting to the Server, you may set this to false.

**logging_level**  The `logging_level` field can be set to various levels of verbosity:

- `info`
- `debug`
- `warning`
- `error`

The default `logging_level` of `warning` is generally appropriate. Cagent logs are stored in the Windows Event Log and can be accessed using the Windows Event Viewer. Logs in the Event Viewer are stored under `Windows Logs>Application`.

11.2.2 Cagent Debug Modes

Normally cagent is installed and run as a service, however it can be run for debugging purposes without actually making it a service. This will also print out error messages to the console instead of the Windows Event Log:

```
cagent.exe debug
```

For running CALDERA operations, cagent should be installed as a service or executed in an elevated command prompt on each computer taking part in the adversary emulation exercise.
This page documents aspects of security and how they relate to CALDERA.

12.1 Security in CALDERA

While CALDERA has not undergone a formal security review, we have tried to make CALDERA as secure as possible. Connections between the Server and Agent are encrypted and the Server’s identity is cryptographically authenticated by the Agent using the Server’s SSL certificate.

However because of the way that the CALDERA Agents log data, it is possible that the logs may contain sensitive information discovered during an engagement. To mitigate this possibility we recommend that the Agent Logging sensitivity be set to Warning or Error (see Configuration for more information on logging levels).

12.2 Does CALDERA Contain Malware?

Most adversaries (and therefore CALDERA) use tools and features that are built into Windows. However CALDERA does contain some third party tools that are commonly used by the security community, but may be considered harmful by antivirus software:

- Invoke-Mimikatz.ps1
- PowerView.ps1
- PowerUp.ps1
- Invoke-ReflectivePEInjection.ps1

12.3 Securing the CALDERA Server

By design, the CALDERA Server is able to execute arbitrary commands on systems that have a CALDERA Agent installed. Access to the computer running the CALDERA Server as well as the CALDERA Server interface should be
protected.

**Warning:** Remember to change the default password to the CALDERA Server.
Adding New Adversary Techniques

Note: This page is a WIP. This is an advance copy of the final version.

In CALDERA an adversary’s smallest executable action is called a Step.

CALDERA pieces individual Steps together to form a sequence of activity that represents an adversary.

New techniques can be added to CALDERA by adding a new Step to the caldera\caldera\app\operation\operation_steps.py file.

In this tutorial we’ll show how this works. First by explaining one of the already existing Steps. Then we’ll walk through adding a new Step.

13.1 Understanding Steps

Let's take a look at a simple step:

This Step finds files that have “password” or “admin” in the name. When it’s run from the System user account, it collects from all user’s home directories otherwise it collects from the current user’s home directory. Let’s break down some of what is going on here.

A Step is a class that inherits from the parent class Step. The name of the Step is the Class name.

Warning: The class name should be selected carefully, because it is used to uniquely identify the Step. If any code within the step is updated (for example to fix a bug) the class name is used to find and update the old definition. If a class is renamed, it will be treated as a completely new Step. Meanwhile the old Step will be removed from the definition of existing adversaries. For example, if Step “A” is renamed to “B”. It is effectively treated as deleting step “A” (which removes it from any adversaries that may have been using it) and creating a new Step “B”.

The step can be documented with a multiline comment. This comment will appear in the CALDERA web interface under the Step’s detailed view. Typically it contains a brief description of the step and any requirements that might be
necessary for it to work.

The step can be tagged with ATT&CK tactics and techniques. This is represented as a list of tuples, where the first item in each tuple is the ATT&CK technique (stored as the ID) and the second entry is the ATT&CK Tactic. The list is saved in the `attack_mapping` class variable. In this case the DirListCollection Step is tagged with three ATT&CK techniques.

The `display_name` class variable stores the displayable name of the Step, which is listed on the web interface.

The `summary` class variable is a short, one sentence description of what the step does.

The `preconditions` class variable stores the Step’s requirements for execution.

The preconditions define the objects that must exist for the Step to be executable. Preconditions can be thought of as database queries, in that they define objects, and conditions on those objects, that the Step needs in order to run.

Preconditions are stored in a format that is easily machine readable (and not necessarily human readable) so we’ll spend some time explaining what they mean. Preconditions are stored as a list of tuples. Each precondition is its own tuple. In this case there are two preconditions (one on each line). The first item in the tuple gives the precondition a name; it must be a string. The second item in the tuple is an expression that represents an object and any conditions on that object.

That’s the formal definition, let’s talk about this example. Here’s the first precondition:

```
('rat', OPRat)
```

The first item is the name of this precondition: `rat`. The second item defines the precondition. In this case, it is a single object, `OPRat`, which is a type that represents a `Rat`. There are no other conditions on this precondition, meaning that any object of type `OPRat` which is known to CALDERA will match against the precondition.

Here is the second precondition:

```
('host', OPHost(OPVar("rat.host")))
```

The first item names the precondition: `host`. The second item starts with `OPHost`, specifying that this precondition refers to objects that are `Hosts`. The next part of the expression denotes conditions on the `OPHost`. `OPVar` is a special keyword which matches to a previously defined precondition. The entire expression, `OPVar("rat.host")`, matches to the `host` field of the previously defined `rat` precondition. In plain terms, the precondition `OPHost(OPVar("rat.host"))` matches to an object of type `OPHost` that is the same object as the `host` field of the `rat` precondition.

In psuedocode this would look something like this:

```
OPHost host = rat.host
```

Now that you understand these preconditions, we’re going to jump down a bit to a different line to show how they’re used.

This defines the Step’s `action` function. The action function is called when CALDERA decides to execute the Step. Notice the parameters `rat` and `host` these match the preconditions that are defined above (because they match the precondition’s names). When the Step is called, objects that match the `rat` and `host` preconditions will be passed into the `action` function.

There are some other parameters to the action function, for the moment you can ignore these, (we’ll get to them soon).

Moving back to where we left off, the next line is:

The class variable `postconditions` defines the effects of the Step. This is used by CALDERA to predict outcome of a Step, so it can build plans of potential Steps to execute.

The structure is similar to `preconditions`. The postconditions are a list of tuples. The first item in a tuple is the name of the postcondition, in this case `"file_g"`. The second item defines the object that will be created as a result
of the Step. In this case it is an OPFile with two properties (represented by the dictionary). Each key-value pair in the dictionary represents properties that will be in the object. So the OPFile will have a field use_case with a value of 'collect' and a 'host' field with a value of OPVar("host") which is refers to the object in the host precondition.

The postcondition is used to create the object within the action function:

Like preconditions, postconditions are passed as a parameter to the action function.

The class variable significant_parameters allows the user to specify which parameters to the action function are significant for tracking repeated actions.

By default CALDERA will not re-run an action if all of the parameters are the same as an action that has previously been executed. However, this behavior can be overridden using the significant_parameters class variable. Here we set the significant_parameter as “host” because we want this Step to only be performed once per host. If we had left the default behavior here, this Step would have been run for every host and rat pair. Because there can be multiple :code:`rat's running on a host, this step would have been run multiple times on the same host which is not what we want.

The last class variable in this example is the postproperties variable.

The is a list of the fields that will be set on postconditions that are defined in the postconditions class variable.

Every Step must have a description function:

The description function must return a string detailing what the step does, preferably with any runtime preconditions filled in. The description will be displayed in the operation view when the step is executed during an operation. Note that in this example the parameter names: rat and host match the names of the preconditions. When CALDERA runs this step, it will pass in the objects that it resolves for each precondition into the appropriate parameter.

Finally we have reached the last part of the Step, the action function. We have seen this multiple times already. Here it is in its entirety:

The function always takes at least one parameter: operation. This is an instance of the OperationWrapper class. It lets the Step perform actions on the Rat, like execute command lines.

The rest of the parameters are the names of preconditions or postconditions that need to be referenced.

The action function contains all the code for the Step to actually perform the action that it needs to. To do this, it will read preconditions (passed in as parameters), execute commands using the operation object, or create objects in the database using the postconditions.

Several other class variables exist that aren’t used in this Step:

- **value** - Default: 1 - This can be used to prioritize this step over others. Steps with a large value will be executed before others.

- **preproperties** - Default: [] - similar to postproperties, this is a list of strings that define fields that must be defined on any preconditions.

- **deterministic** - Default: False - this changes how CALDERA tracks whether a Step is redundant. When this is set to True CALDERA will ignore the significant_parameters and will instead use the postconditions to determine whether the step is redundant. More specifically, if the postconditions indicate that the step will add something new, like create a RAT on a host that doesn’t currently have a RAT, CALDERA will perform the Step, even if it has already done the exact step before. This is probably easiest to understand with an example; most Lateral Movement Steps are labeled as deterministic because this allows them to repeated if a defender interferes with CALDERA, for example, by killing a running RAT. In some cases, its not possible to represent the outcome of a techniques. For example, credential dumping techniques often don’t know ahead of time what credentials will be discovered. In this case, the postconditions can’t accurately repre-

## 13.1. Understanding Steps
13.2 Adding a New Step

Now that we’ve deconstructed a Step, we’ll walk through creating a new one. Our new step will search the current user’s home folder for files with a specific extension (.pem) and print the contents of the file.

To start, we’ll create a new Step, fill in the appropriate ATT&CK tags and give it a name and summary:

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106",
    'Execution']
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"
```

Next we will define the preconditions. We’ll start with just a simple requirement for a rat:

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106",
    'Execution']
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"
    preconditions = [('rat', OPRat)]
```

Next we will define the postconditions, in this case, we will say that we have collected files as a result of running this step. We also can define that the files are on the rat’s computer.

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106",
    'Execution']
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"
    preconditions = [('rat', OPRat)]
    postconditions = [('file_g', OPFile({'use_case': 'collected',
                                           'host': OPVar("rat.host")}))]
```

We only want this to be run once per host, so we’d like to set the significant parameters to be the host that the Step is being executed on, which is the Rat’s host (that is, `rat.host`), however we need to have the host as a named precondition in order to do this, so we will also have to modify our preconditions to create a new precondition to refer to the Rat’s host.

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106",
    'Execution']
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"
    preconditions = [('rat', OPRat),
                      ('host', OPHost(OPVar("rat.host")))]
    postconditions = [('file_g', OPFile({'use_case': 'collect',
                                            'host': OPVar("host")})
```

(continues on next page)
Next we will write the description function. We can refer to objects in the preconditions, which makes the description more tailored to the exact action that is being run.

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106", "Execution"]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"

    preconditions = [rat, OPRat, host, OPHost(OPVar("rat.host"))]
    postconditions = [file_g, OPFile({use_case: 'collect', host: OPVar("host"))}]

    significant_parameters = [host]

    @staticmethod
    def description(rat, host):
        return "Using cmd to recursively look for .pem files to collect on {}".format(host.hostname)
```

Now we will write the action function. We’ll just write a stub for now.

```python
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106", "Execution"]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"

    preconditions = [rat, OPRat, host, OPHost(OPVar("rat.host"))]
    postconditions = [file_g, OPFile({use_case: 'collect', host: OPVar("host"))}]

    significant_parameters = [host]

    @staticmethod
    def description(rat, host):
        return "Using cmd to recursively look for .pem files to collect on {}".format(host.hostname)

    @staticmethod
    async def action(operation):
        return True
```

At a minimum the action function takes a parameter for the operation and has to return a boolean indicating whether the action succeeded or failed. Next we’ll add parameters for our preconditions and postconditions (the keyword names must match the names of the preconditions and postconditions as defined in the class variables).
class PEMCollection(Step):
    attack_mapping = [('T1005', 'Collection'), ('T1083', 'Discovery'), ('T1106', 'Execution')]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension with a for loop and the dir command recursively"
    preconditions = [('rat', OPRat), ('host', OPHost(OPVar("rat.host")))]
    postconditions = [('file_g', OPFile({'use_case': 'collect', 'host': OPVar("host")}))]
    significant_parameters = ['host']

@staticmethod
def description(rat, host):
    return "Using cmd to recursively look for .pem files to collect on {}".format(host.hostname)

@staticmethod
async def action(operation, rat, host, file_g):
    return True

Now we'll actually add some logic to the action function. The first thing we need to identify is the user account that the Rat is running under. This will determine the files that we will be able to access. The rat's user context is stored in the username field of the rat. This is a string that is formatted in the typical Windows way, as <windows domain>\<username>, for example, caldera\administrator or nt authority\system. If we are operating as the system user, we will be able to access all files, so we can search the entire C:\Users\ folder. Otherwise we will parse out the username portion and use that as the sub-directory within the Users’s folder.

Note: In some cases, the user’s home folder will not be C:\Users\<username>. To be more robust we should actually be using the USERPROFILE environment variable. However the CALDERA RAT does not yet support environment variable replacement so we construct the home folder using the username instead.

We have one last thing to do. To ensure that the rat.username property exists, we have to define it as a preproperty, otherwise the rat object that is provided may not have the username field defined.

Here is the Step with all of these changes:

class PEMCollection(Step):
    attack_mapping = [('T1005', 'Collection'), ('T1083', 'Discovery'), ('T1106', 'Execution')]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension in the user's home directory"
    preconditions = [('rat', OPRat), ('host', OPHost(OPVar("rat.host")))]
    postconditions = [('file_g', OPFile({'use_case': 'collect', 'host': OPVar("host")}))]
    significant_parameters = ['host']
Next we will execute a Windows command to recursively search through the directory to discover files that match our path. This can be done with the Windows built-in command dir. CALDERA has some utilities that generate command lines for Windows commands, including dir. These are located in caldera.app.commands and are documented in Commands API. The function we want to call is caldera.app.commands.cmd.dir_list(). The documentation shows that the command takes in a bunch of arguments and returns two values, an instance of caldera.app.commands.command.CommandLine and a function that can parse the output of the dir command. This parser function will take the output of the dir command and returns to us a well formatted list of the files that were found. We will pass both the CommandLine and the parser into the operation.execute_shell_command() function. operation is an instance of caldera.app.operation.operation_obj.OperationWrapper. The execute_shell_command() function takes a rat, a CommandLine and a parser function as arguments. It will execute the CommandLine on the Rat and pass the output of the command to the parser, returning the result of the parser which in this case will be a list of files ending with the .pem extension. The parser can also detect error outputs from dir. We will catch the FileNotFoundError.

```python
from ..commands import cmd
class PEMCollection(Step):
    attack_mapping = ["T1005", "Collection"], ["T1083", "Discovery"], ["T1106", "Execution"]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension in the user's home directory"
    preconditions = ["rat", OPRat],
                    ["host", OPHost(OPVar("rat.host"))]
    postconditions = ["file_g", OPFile({"use_case": 'collect',
                                         "host": OPVar("host")})]
    significant_parameters = ["host"]
    preproperties = ["rat.username"]

    @staticmethod
    def description(rat, host):
        return "Using cmd to recursively look for .pem files to collect on {}".
        --format(host.hostname)
    @staticmethod
```

(continues on next page)
```python
async def action(operation, rat, host, file_g):
    if "system" in rat.username:
        path = "C:\\Users\\*.pem"
    else:
        path = "C:\\Users\\{rat.username.split(\\)[1]}\*.pem".format(rat.username.split(\\)[1])

    try:
        files = await operation.execute_shell_command(rat, *cmd.dir_
                           list(search=path, b=True, s=True, a="-d"))
    except FileNotFoundError:
        # the path was invalid, the file wasn't found, or access denied, so move on
        pass

    return True
```

For each file that we discover with `dir` we want to print out the contents. One way of doing this is with the Windows command `type`. If we look at the *Commands API*, we can see that CALDERA does not have a generator for the `type` command yet, so we’ll make one.

The format is fairly straightforward:

```python
# add this to the caldera.app.commands.cmd module
def type(path: str) -> Tuple[CommandLine, Callable[[str], None]]:
    """
    type is the command to show the contents of the file
    Args:
    path: the path of the file contents to be shown
    """
    args = ['cmd /c type', """" + path + """]

    return CommandLine(args), parsers.cmd.type
```

Note that we are placing this function within the `caldera.app.commands.cmd` module because `type` is actually a built-in command to the Windows terminal program, `cmd.exe`. By convention, individual programs are stored in their own module and sub-commands are stored within that module. However, you’re free to follow whatever standard you would like.

We’ll also need a parser to parse the output of `type`. In this case we don’t actually have to do any special, so our parser will rather simply just return everything.

```python
# add this to the cmd class in caldera.app.commands.parsers
@staticmethod
def type(text: str) -> None:
    return text
```

Note: Parsers can be significantly more complex than the one that we have created here. One of the things they can do is raise exceptions when they encounter an error. Many different kinds of exceptions already exist and are documented in `caldera.app.commands.errors`. If you write a step that calls a command, be sure to check for any exceptions that may be generated. Feel free to add your own exceptions as well.

We can now call this from within the `action` function. Here’s the completed code:
class PEMCollection(Step):
    attack_mapping = [('T1005', 'Collection'), ('T1083', 'Discovery'), ('T1106', 'Execution')]
    display_name = "Get PEM"
    summary = "Get the contents of files with a .pem extension in the user's home directory"

    preconditions = [('rat', OPRat), ('host', OPHost(OPVar("rat.host")))]

    postconditions = [('file_g', OPFile({'use_case': 'collect', 'host': OPVar("host"))}]]

    significant_parameters = ['host']
    preproperties = ['rat.username']

    @staticmethod
    def description(rat, host):
        return "Using cmd to recursively look for .pem files to collect on {}".
        format(host.hostname)

    @staticmethod
    async def action(operation, rat, host, file_g):
        if 'system' in rat.username:
            path = "C:\Users\*.pem"
        else:
            path = "C:\Users\{}\*.pem".format(rat.username.split("\")[1])

        try:
            files = await operation.execute_shell_command(rat, *cmd.dir
            list(search=path,
            b=True,
            _s=True, a="-d"))
            for file in files:
                contents = await operation.execute_shell_command(rat, *cmd.
                type(file))
                print(contents)
        except FileNotFoundError:
            # the path was invalid, the file wasn't found, or access denied, so move
            on
            pass

        return True
CALDERA uses a simple encoding scheme to disguise some of the external scripts and tools used by the project. This can be useful in preventing AV software from interfering with the operation of the CALDERA server.

15.1 Using the script editor

Small changes to external scripts can be made via the CALDERA web application via the built-in Script Editor.

15.2 Manually with encode.py

For larger changes and encoding binary files scripts/encode.py can be used. This script will read in a file specified with the \(-i\) option and output an encoded file to a path specified with the \(-o\) option.

15.2.1 Example

The following series of commands are an example of downloading and encoding a new version of powerview using the encode.py script.

```
cd scripts/
# Download a version of powerview from Empire’s dev branch
curl -L -o powerview.psl https://github.com/EmpireProject/Empire/raw/dev/
          data/module_source/situational_awareness/network/powerview.psl

# Encode the powershell script
python encode.py -i powerview.psl  -o powerview-psl

mv powervew-psl ../caldera/files
```

(continues on next page)
# remove the downloaded file
rm powerview.psl
### Legal Information

This software was produced for the U. S. Government under Basic Contract No. W15P7T-13-C-A802, and is subject to the Rights in Noncommercial Computer Software and Noncommercial Computer Software Documentation Clause 252.227-7014 (FEB 2012)

3. 2017 The MITRE Corporation

This project contains content developed by The MITRE Corporation. If this code is used in a deployment or embedded within another project, it is requested that you send an email to opensource@mitre.org in order to let us know where this software is being used.

CALDERA Copyright 2017 The MITRE Corporation

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ATT&CK Copyright 2017 The MITRE Corporation The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use Adversarial Tactics, Techniques and Common Knowledge (ATT&CK™) for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE’s copyright designation and this license in any such copy.

Many libraries are used within this software and their respective copyrights and licenses apply. See the LICENSE-3RD-PARTY file within CALDERA’s source code for an exhaustive list of licenses.

====================================================================

Mimikatz
====================================================================

Creative Commons Attribution 4.0 International Public License

(continues on next page)
By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 - Definitions.

Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to the Licensed Material and that the Licensor has authority to license.

Licensor means the individual(s) or entity(ies) granting rights under this Public License.

Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 - Scope.

License grant.

Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

(continues on next page)
reproduce and Share the Licensed Material, in whole or in part; and produce, reproduce, and Share Adapted Material.

Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

Term. The term of this Public License is specified in Section 6(a).

Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

Downstream recipients.

Offer from the Licensor - Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

Other rights.

Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

Patent and trademark rights are not licensed under this Public License. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 - License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

Attribution.

If You Share the Licensed Material (including in modified form), You must:

retain the following if it is supplied by the Licensor with the Licensed Material:
- identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
- a copyright notice;
- a notice that refers to this Public License;
- a notice that refers to the disclaimer of warranties;
- a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
indicate if You modified the Licensed Material and retain an indication of any __previous modifications; and __
indicate the Licensed Material is licensed under this Public License, and include the __
- text of, or the URI or hyperlink to, this Public License. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on __
- the medium, means, and context in which You Share the Licensed Material. For __
- example, it may be reasonable to satisfy the conditions by providing a URI or __
- hyperlink to a resource that includes the required information. If requested by the Licensor, You must remove any of the information required by __
- Section 3(a)(1)(A) to the extent reasonably practicable. If You Share Adapted Material You produce, the Adapter's License You apply must not __
- prevent recipients of the Adapted Material from complying with this Public License. Section 4 - Sui Generis Database Rights. Where the Licensed Rights include Sui Generis Database Rights that apply to Your use __
- of the Licensed Material:

for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, __
- reproduce, and Share all or a substantial portion of the contents of the database; if You include all or a substantial portion of the database contents in a database in __
- which You have Sui Generis Database Rights, then the database in which You have Sui __
- Generis Database Rights (but not its individual contents) is Adapted Material; and __
- You must comply with the conditions in Section 3(a) if You Share all or a substantial __
- portion of the contents of the database. For the avoidance of doubt, this Section 4 supplements and does not replace Your __
- obligations under this Public License where the Licensed Rights include other __
- Copyright and Similar Rights. Section 5 - Disclaimer of Warranties and Limitation of Liability. Unless otherwise separately undertaken by the Licensor, to the extent possible, the __
- Licensor offers the Licensed Material as-is and as-available, and makes no __
- representations or warranties of any kind concerning the Licensed Material, whether __
- express, implied, statutory, or other. This includes, without limitation, __
- warranties of title, merchantability, fitness for a particular purpose, non- __
- infringement, absence of latent or other defects, accuracy, or the presence or __
- absence of errors, whether or not known or discoverable. Where disclaimers of __
- warranties are not allowed in full or in part, this disclaimer may not apply to You. To the extent possible, in no event will the Licensor be liable to You on any legal __
- theory (including, without limitation, negligence) or otherwise for any direct, __
- special, indirect, incidental, consequential, punitive, exemplary, or other losses, __
- costs, expenses, or damages arising out of this Public License or use of the __
- Licensed Material, even if the Licensor has been advised of the possibility of such __
- losses, costs, expenses, or damages. Where a limitation of liability is not allowed, __
- in full or in part, this limitation may not apply to You. The disclaimer of warranties and limitation of liability provided above shall be __
- interpreted in a manner that, to the extent possible, most closely approximates an __
- absolute disclaimer and waiver of all liability. Section 6 - Term and Termination. This Public License applies for the term of the Copyright and Similar Rights licensed __
- here. However, if You fail to comply with this Public License, then Your rights __
- under this Public License terminate automatically. Where Your right to use the Licensed Material has terminated under Section 6(a), it __
- reinstates: automatically as of the date the violation is cured, provided it is cured within 30 __
- days of Your discovery of the violation; or
upon express reinstatement by the Licensor. For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License. Sections 1, 5, 6, 7, and 8 survive termination of this Public License. Section 7 - Other Terms and Conditions.

The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License. Section 8 - Interpretation.

For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

=============================================================
MemoryModule
=============================================================

Mozilla Public License Version 2.0
==================================

1. Definitions
-------------

1.1. "Contributor"
 means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
 means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
 means Covered Software of a particular Contributor.

1.4. "Covered Software"
 means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.
1.5. "Incompatible With Secondary Licenses"
   means
   (a) that the initial Contributor has attached the notice described
       in Exhibit B to the Covered Software; or
   (b) that the Covered Software was made available under the terms of
       version 1.1 or earlier of the License, but not also under the
       terms of a Secondary License.

1.6. "Executable Form"
   means any form of the work other than Source Code Form.

1.7. "Larger Work"
   means a work that combines Covered Software with other material, in
   a separate file or files, that is not Covered Software.

1.8. "License"
   means this document.

1.9. "Licensable"
   means having the right to grant, to the maximum extent possible,
   whether at the time of the initial grant or subsequently, any and
   all of the rights conveyed by this License.

1.10. "Modifications"
    means any of the following:
    (a) any file in Source Code Form that results from an addition to,
        deletion from, or modification of the contents of Covered
        Software; or
    (b) any new file in Source Code Form that contains any Covered
        Software.

1.11. "Patent Claims" of a Contributor
    means any patent claim(s), including without limitation, method,
    process, and apparatus claims, in any patent Licensable by such
    Contributor that would be infringed, but for the grant of the
    License, by the making, using, selling, offering for sale, having
    made, import, or transfer of either its Contributions or its
    Contributor Version.

1.12. "Secondary License"
    means either the GNU General Public License, Version 2.0, the GNU
    Lesser General Public License, Version 2.1, the GNU Affero General
    Public License, Version 3.0, or any later versions of those
    licenses.

1.13. "Source Code Form"
    means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
    means an individual or a legal entity exercising rights under this
    License. For legal entities, "You" includes any entity that
    controls, is controlled by, or is under common control with You. For
purposes of this definition, "control" means (a) the power, direct
or indirect, to cause the direction or management of such entity,
whether by contract or otherwise, or (b) ownership of more than
fifty percent (50%) of the outstanding shares or beneficial
ownership of such entity.

2. License Grants and Conditions
--------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)
Licensable by such Contributor to use, reproduce, make available,
modify, display, perform, distribute, and otherwise exploit its
Contributions, either on an unmodified basis, with Modifications, or
as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer
for sale, have made, import, and otherwise transfer either its
Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution
become effective for each Contribution on the date the Contributor first
distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under
this License. No additional rights or licenses will be implied from the
distribution or licensing of Covered Software under this License.
Notwithstanding Section 2.1(b) above, no patent license is granted by a
Contributor:

(a) for any code that a Contributor has removed from Covered Software;

or

(b) for infringements caused by: (i) Your and any other third party's
modifications of Covered Software, or (ii) the combination of its
Contributions with other software (except as part of its Contributor
Version); or

(c) under Patent Claims infringed by Covered Software in the absence of
its Contributions.

This License does not grant any rights in the trademarks, service marks,
or logos of any Contributor (except as may be necessary to comply with
the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this
License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of
the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor notifies You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions,
counter-claims, and cross-claims) alleging that a Contributor Version
directly or indirectly infringes any patent, then the rights granted to
You by any and all Contributors for the Covered Software under Section
2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which
have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

************************************************************************
* *
* 6. Disclaimer of Warranty *
* ------------------------- *
* *
* Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or *
* statutory, including, without limitation, warranties that the *
* Covered Software is free of defects, merchantable, fit for a *
* particular purpose or non-infringing. The entire risk as to the *
* quality and performance of the Covered Software is with You. *
* Should any Covered Software prove defective in any respect, You *
* (not any Contributor) assume the cost of any necessary servicing, *
* repair, or correction. This disclaimer of warranty constitutes an *
* essential part of this License. No use of any Covered Software is *
* authorized under this License except under this disclaimer. *
* *
************************************************************************
************************************************************************
* *
* 7. Limitation of Liability *
* -------------------------- *
* *
* Under no circumstances and under no legal theory, whether tort *
* (including negligence), contract, or otherwise, shall any *
* Contributor, or anyone who distributes Covered Software as *
* permitted above, be liable to You for any direct, indirect, *
* special, incidental, or consequential damages of any character *
* including, without limitation, damages for lost profits, loss of *
* goodwill, work stoppage, computer failure or malfunction, or any *
* and all other commercial damages or losses, even if such party *
* shall have been informed of the possibility of such damages. This *
* limitation of liability shall not apply to liability for death or *
* personal injury resulting from such party's negligence to the *
* extent applicable law prohibits such limitation. Some *
* jurisdictions do not allow the exclusion or limitation of *
* incidental or consequential damages, so this exclusion and *
* limitation may not apply to You. *
* *
************************************************************************
************************************************************************

8. Litigation
-------------

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

PowerSploit

PowerSploit is provided under the 3-clause BSD license below.

Copyright (c) 2012, Matthew Graeber
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PyYAML

Copyright (c) 2006 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Bootstrap

The MIT License (MIT)

Copyright (c) 2011-2016 Twitter, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

normalize.css

Copyright © Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(continues on next page)
Bootstrap-Multiselect

Copyright (c) 2012 - 2015 David Stutz All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of David Stutz nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

js-bson

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Binary form" shall mean the compiled form, including but not limited to bytecode, object code, execution images, binary libraries, and data files.

"Licensee" shall mean the recipient of the software under the terms of this License.

"Contribution" shall mean any work of authorship, including the original version, any modifications and any additions to the original version, or any optionally licensed additions to the original version, in each case including portions either added in source code form, in object code form, in execution image form, or in a form resulting from independent development by another authorized Contributor, and any documentation generated during the creation of the Contributions.

"Contributor" shall mean Licensor or any individual or Legal Entity authorized by Licensor that creates or contributes to such Contributions.

"Contributor License Agreement" shall mean an agreement by an individual or Legal Entity releasing Contributions on behalf of the Contributors, and governing their rights and obligations regarding Contributions to the Software, whether such Contributions are made directly to the Software or by means of a distribution facility provided by a Contributor.  

Chapter 16. Legal Information
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
Accepting Warranty or Additional Liability. While redistributing the Work or
Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance,
of support, warranty, indemnity, or other liability obligations and/or rights
consistent with this License. However, in accepting such obligations, You may act
only on Your own behalf and on Your sole responsibility, not on behalf of any other
Contributor, and only if You agree to indemnify, defend, and hold each Contributor
harmless for any liability incurred by, or claims asserted against, such
Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice,
with the fields enclosed by brackets "[]" replaced with your own identifying
information. (Don't include the brackets!) The text should be enclosed in the
appropriate comment syntax for the file format. We also recommend that a file or
class name and description of purpose be included on the same "printed page" as the
copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under
the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
ANY KIND, either express or implied. See the License for the specific language
governing permissions and limitations under the License.

=================================================================================================
crossroads.js
=================================================================================================

The MIT License (MIT)
Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this,
software and associated documentation files (the "Software"), to deal in the,
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the,
following conditions:

The above copyright notice and this permission notice shall be included in all copies,
or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION,
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE,
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(continues on next page)
d3.js

Copyright 2010-2016 Mike Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of the author nor the names of contributors may be used to
  endorse or promote products derived from this software without specific prior
  written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----------------------------
Hasher.js

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this,
-software and associated documentation files (the "Software"), to deal in the,
-Software without restriction, including without limitation the rights to use, copy,
-modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
-and to permit persons to whom the Software is furnished to do so, subject to the,
-following conditions:

The above copyright notice and this permission notice shall be included in all copies,
or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A,
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION,
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE,
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------
jQuery

(continues on next page)
Copyright jQuery Foundation and other contributors, https://jquery.org/

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

Knockout.js

The MIT License (MIT) - http://www.opensource.org/licenses/mit-license.php

Copyright (c) Steven Sanderson, the Knockout.js team, and other contributors http://knockoutjs.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(continues on next page)
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

============================================
knockout-sortable
============================================

The MIT License (MIT)

Copyright (c) 2015 Ryan Niemeyer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

============================================
lodash
============================================

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own
licenses; we recommend you read them, as their terms may differ from the
terms above.

=============================================================

Js-Signals

The MIT License (MIT)
Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this,
software and associated documentation files (the "Software"), to deal in the,
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the,
following conditions:

The above copyright notice and this permission notice shall be included in all copies,
or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A,
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT,
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION,
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE,
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=============================================================

Python

(continues on next page)
A. HISTORY OF THE SOFTWARE  
==========================

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.
B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

(continues on next page)
BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
-----------------------------------------------

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be
bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-------------------------------------------------------------------
Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Stichting Mathematisch
Centrum or CWI not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-------------------------------------------------------------------
Ace
-------------------------------------------------------------------

Copyright (c) 2010, Ajax.org B.V.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright
    notice, this list of conditions and the following disclaimer in the
    documentation and/or other materials provided with the distribution.
  * Neither the name of Ajax.org B.V. nor the
    names of its contributors may be used to endorse or promote products
    derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL AJAX.ORG B.V. BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
jsTree

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Empire

Copyright (c) 2017, Will Schroeder, Justin Warner, Matt Nelson, Steve Borosh, Alex—Rymdeko-harvey, Chris Ross
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Empire nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
CHAPTER 17

API Information

This is information for developers, including API documentation and internal functioning.

17.1 Commands API

The Commands API contains helper functions for constructing `caldera.app.commands.command.CommandLine` objects and parser functions to parse the output of those commands when run. It is typically used when creating custom Steps.

17.1.1 CommandLine

17.1.2 CommandLine generators

17.1.3 Parser Exceptions

```python
exception caldera.app.commands.errors.AVBlockError
    Represents an error caused because AntiVirus software blocked the action

exception caldera.app.commands.errors.AccessDeniedError
    Represents an error caused because access was denied

exception caldera.app.commands.errors.AccountDisabledError
    Represents an error caused because an account is disabled

exception caldera.app.commands.errors.AcquireLSAError
    Represents an error caused because the process could not acquire access to LSA

exception caldera.app.commands.errors.CantControlServiceError
    Represents an error caused because a service cannot be controlled

exception caldera.app.commands.errors.DomainIssueError
    Represents an error caused because there is a domain issue
```
exception caldera.app.commands.errors.FileInUseError
    Represents an error caused because a file is currently in use

exception caldera.app.commands.errors.IncorrectParameterError
    Represents an error caused because a parameter was incorrect

exception caldera.app.commands.errors.NoFileError
    Represents an error caused by a file not being found

exception caldera.app.commands.errors.NoNetworkPathError
    Represents an error caused because a network path does not exist

exception caldera.app.commands.errors.NoProcessError
    Represents an error caused because a process does not exist

exception caldera.app.commands.errors.NoRegKeyError
    Represents an error caused because a registry key was missing

exception caldera.app.commands.errors.NoServiceError
    Represents an error caused because a service does not exist

exception caldera.app.commands.errors.NoShareError
    Represents an error caused by a share not existing

exception caldera.app.commands.errors.ParseException
    Represents a generic parsing error

exception caldera.app.commands.errors.ParserNotImplementedError
    Represents an error caused because the parser does not support it

exception caldera.app.commands.errors.PathSyntaxError
    Represents an error caused because the syntax of a path is incorrect

exception caldera.app.commands.errors.ServiceAlreadyRunningError
    Represents an error caused because a service is already running

exception caldera.app.commands.errors.ServiceNotStartedError
    Represents an error caused because a service is not started

exception caldera.app.commands.errors.UnresponsiveServiceError
    Represents an error caused because a service is unresponsive

17.2 Planner API

17.3 Operation API
The following are build instructions for the CALDERA agent and Crater. They should not be necessary for normal users.

### 18.1 Cagent Build Instructions

1. **Install Python 3.5**  
   Install Python 3.5 can be acquired from the Operating System’s package manager or from https://www.python.org/. The standard installation is straightforward.

2. **Upgrade to setuptools 24.0 or later**  
   The python package setuptools, version 24.0 or later must be installed. Setuptools comes with Python 3.5, but some versions may not be up to date. To update it, run:
   ```bash
   pip install --upgrade setuptools
   ```

3. **Install Visual C++ 2015 Build Tools**  
   Install the Visual C++ 2015 Build Tools. During install check Windows 8.1 SDK and Windows 10 SDK options.

   **Note:** If Visual Studio 2015 is already installed, **Visual C++ 2015 Build Tools** should not need to be installed.

4. **Install PyWin32 v.220**  
   PyWin32 v.220 must be installed.

5. **Install py2exe**  
   A modified version of py2exe must be installed. It should be provided with CALDERA and installed by executing `easy_install py2exe-0.9.2.2-py3.5.egg`

6. **Install additional Python Dependences**  
   Additional Python dependencies are downloaded and installed automatically by running the following command within the `caldera/dep/caldera-agent/` directory:
   ```bash
   pip install -r requirements.txt
   ```

7. **Compile Caldera Agent**  
   Compile the Caldera agent by running the following command twice within the `caldera/dep/caldera-agent/caldera_agent/` directory:
python setup.py

After running the above command twice the caldera agent executable will be built and located at caldera/dep/caldera-agent/caldera_agent/dist/cagent.exe

18.2 Crater Build Instructions

Crater must be compiled on a Windows system. If the Caldera server is installed on a Linux system, Crater must be built on Windows and then copied to its build location: caldera/dep/crater/crater/CraterMain.exe

1. Install Mono  Download and install Mono.

2. Run build.bat  Navigate to the caldera/dep/crater/crater and execute build.bat
CHAPTER 19

Indices and tables

• genindex
• modindex
caldera.app.commands.errors, 75
Index

A
AccessDeniedError, 75
AccountDisabledError, 75
AcquireLSAError, 75
AVBlockError, 75

C
caldera.app.commands.errors (module), 75
CantControlServiceError, 75

D
DomainIssueError, 75

F
FileInUseError, 75

I
IncorrectParameterError, 76

N
NoFileError, 76
NoNetworkPathError, 76
NoProcessError, 76
NoRegKeyError, 76
NoServiceError, 76
NoShareError, 76

P
ParseError, 76
ParserNotImplementedError, 76
PathSyntaxError, 76

S
ServiceAlreadyRunningError, 76
ServiceNotStartedError, 76

U
UnresponsiveServiceError, 76