MOBIA is a business technology integrator with over thirty years of experience and 500+ employees across Canada and USA. Our talented bench of technical engineers and trusted advisors deliver process improvements and business transformations within our core pillars of Cloud, Infrastructure, Software Development, Cybersecurity and Broadband & Wireless Services.

Our inside-out approach allows us to understand business challenges from deep within the organization, mapping the impact as it ripples outward. This insight enables us to create future-proof solutions that maximize results and repeatedly exceed our client's expectations.

Adversary Simulation Workshop

Lab Guide:
Install Atomic Red Team

Objective:

Install and Configure the Atomic Red Team PowerShell Execution framework to simplify the execution of atomic tests.

Instructions:

An "Execution Framework" is a tool to aid in the execution of atomic tests, so we don't have to copy and paste commands into the specified executors (e.g. PowerShell or cmd.exe).

The main Atomic Red Team GitHub page has a link to the "atomics" folder where the atomic tests are defined but it also has a link to the execution frameworks available. Click on the "execution-frameworks" link as shown below.

Bookmarks→ ART→ Main GitHub

Bookmarks→ ART→ Execution Frameworks

Getting Started

- Getting Started With Atomic Red Team
- Automated Test Execution with the Execution Frameworks
- Peruse the Complete list of Atomic Tests (md, csv) and the ATT&CK Matrix
  - Windows Matrix and tests by tactic (md, csv)
  - MacOS Matrix and tests by tactic (md, csv)
  - Linux Matrix and tests by tactic (md, csv)
- Using ATT&CK Navigator? Check out our coverage layers (All, Windows, MacOS, Linux)
- Fork and Contribute your own modifications
- Have questions? Join the community on Slack at https://atomicredteam.slack.com
  - Need a Slack invitation? Submit an invite request via this Google Form

Execute an Atomic Test with an Execution Framework

There are a variety of Execution Frameworks that automate the execution of the atomic tests defined in this repository. The most actively maintained and feature rich execution framework is the PowerShell Invoke-AtomicRedTeam framework. It works cross-platform for executing atomic tests locally or on remote machines. There are also Python and GoLang versions developed by the community.

The PowerShell execution framework is currently the most actively developed framework and works across all operating systems. The Python executor is helpful in the case that you are executing tests on Linux or macOS and cannot, or do not, want to install PowerShell core.

The executor code for the Python and Ruby executors exists within the same atomic-red-team repository but the PowerShell framework has its own dedicated repository. Click "here" to go to the PowerShell Invoke-AtomicRedTeam repository.

Bookmarks→ ART→ Invoke-AtomicRedTeam

README.md

Invoke-AtomicRedTeam is a PowerShell module to execute tests as defined in the atomics folder of Red Canary's Atomic Red Team project. The "atomics folder" contains a folder for each Technique defined by the MITRE ATT&CK™ Framework. Inside of each of these "T#" folders you'll find a **yaml** file that defines the attack procedures for each atomic test as well as an easier to read markdown (**md**) version of the same data.

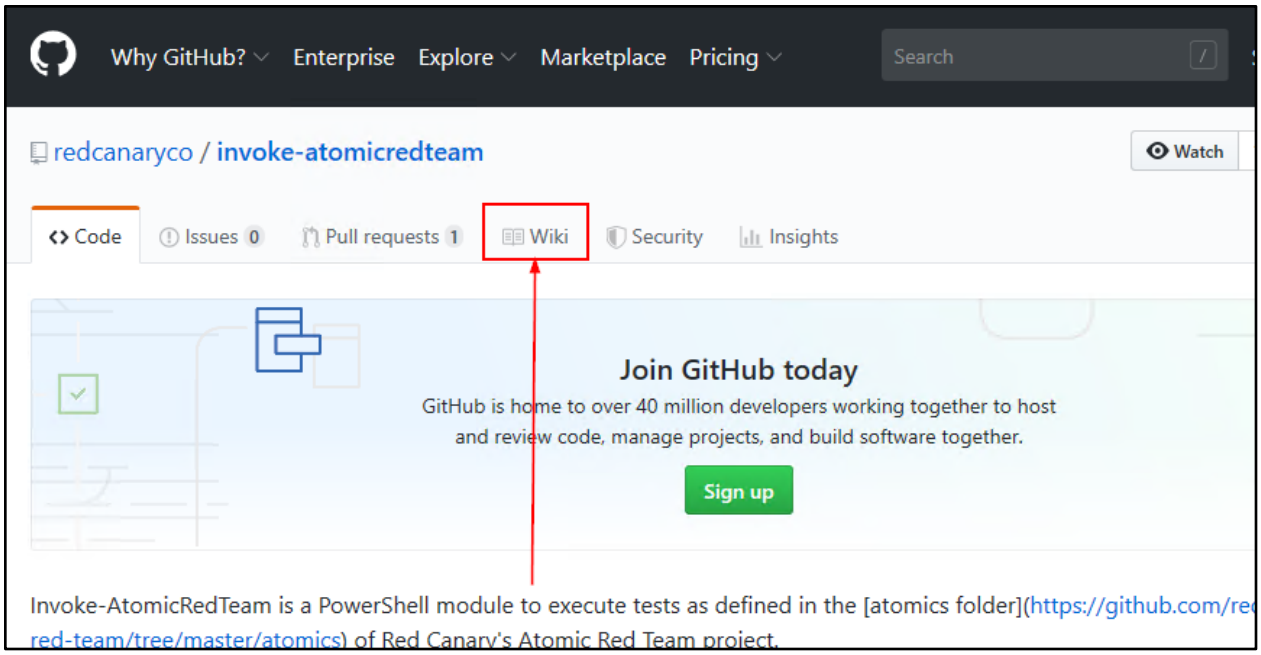
- Executing atomic tests may leave your system in an undesirable state. You are responsible for understanding what a test does before executing.
- Ensure you have permission to test before you begin.
- It is recommended to set up a test machine for atomic test execution that is similar to the build in your environment. Be sure you have your collection/EDR solution in place, and that the endpoint is checking in and active.

See the Wiki for complete Installation and Usage instructions.

Note: This execution frameworks works on Windows, MacOS and Linux. If using on MacOS or Linux you must install PowerShell Core first.

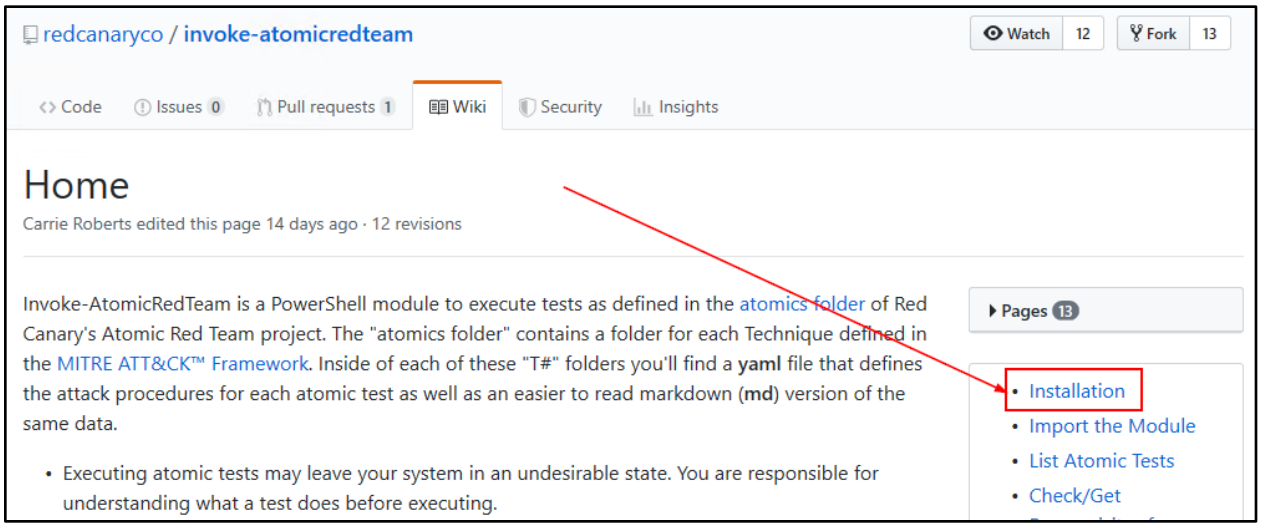
Click on the Wiki link at the top of the page or click here.

Bookmarks→ ART→ Invoke-AtomicRedTeam Wiki



The Wiki is full of helpful information about how to install,configure and use Atomic Red Team and the execution framework.

Click on the “Installation” link on the right.

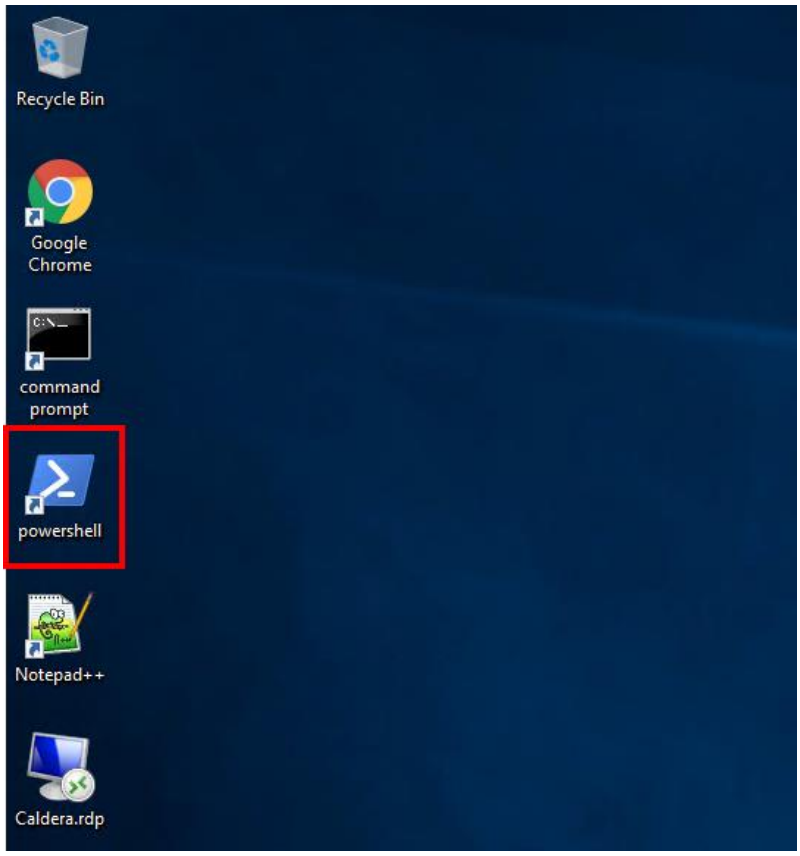


Please read through the installation instructions on the Wiki. You will notice there are three installation options. The first installs only the execution framework, without the atomic test definitions. This is helpful if we are in

an environment where we don't want to download the entire atomics folder full of simulated malware which is likely to set off alarms or be automatically removed. In this scenario, we may want to hand pick only the atomics we are going to run and copy only those over to the system.

The second installation option will install both the Execution Framework and the "atomics" folder full of the atomic test definitions and supporting files. This is the installation option we want to use for the labs in this class.

Follow the instructions in the "Install Execution Framework and Atomics Folder" section. The installation commands should be run from a PowerShell prompt. You can start the PowerShell prompt by clicking on the shortcut on the desktop.



Paste the commands from the Wiki into the PowerShell prompt. You might need to hit “Enter” after pasting the commands.

```
IEX (IWR
'https://raw.githubusercontent.com/redcanaryco/invoke-
atomicredteam/master/install-atomicredteam.ps1' -
UseBasicParsing);
Install-AtomicRedTeam -getAtomics
```

You will also be prompted to import the NuGet provider to which you should answer “Y”. If you have run this command before, you will need to add the “-Force” parameter to force the overwriting of the previous installation.

```
powershell
PS C:\Users\art> IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-a
tomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);
PS C:\Users\art> Install-AtomicRedTeam -getAtomics

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact
with NuGet-based repositories. The NuGet provider must be available in
'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\art\AppData\Local\PackageManagement\ProviderAssemblies'. You can also
install the NuGet provider by running 'Install-PackageProvider -Name NuGet
-MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install and
import the NuGet provider now?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
Installation of AtomicRedTeam Failed.
File C:\AtomicRedTeam\invoke-atomicredteam\Private\AtomicClassSchema.ps1 cannot b
e loaded because running scripts is disabled on this system. For more information
, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
```

Notice that the last red line is telling us that the installation failed? The reason it failed is because there is a PowerShell Execution policy in place preventing the running of scripts. We will need to bypass this safety feature to use the execution framework. For simplicity in these labs, we will just bypass the execution policy completely for the current user. There are options for just temporarily bypassing the execution policy but we don’t cover those here.

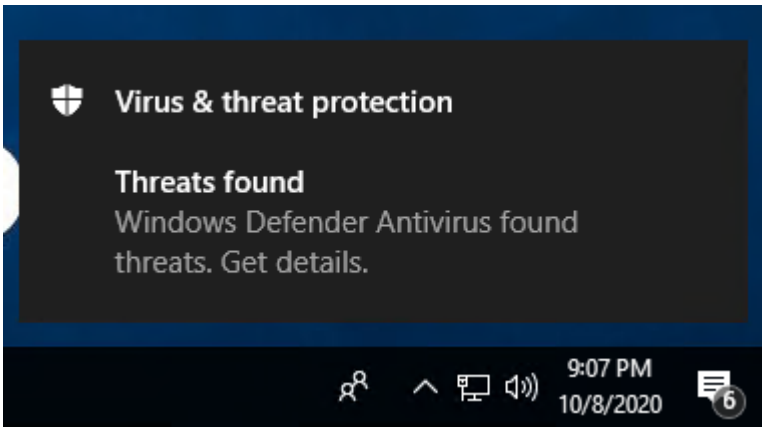
```
Set-ExecutionPolicy Bypass -Scope CurrentUser
```

```
Select powershell
PS C:\Users\art> Set-ExecutionPolicy Bypass -Scope CurrentUser

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust.
Changing the execution policy might expose you to the security risks described
in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the
execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "N"):Y
```

Bypassing the execution policy outside of the lab environment may not be as straightforward in your environment but one of these methods is likely to work. Method 12 is especially promising.

Now that we have installed the execution framework and the "atomics folder" containing test definitions and simulated malware we start seeing Windows Defender showing disapproval.

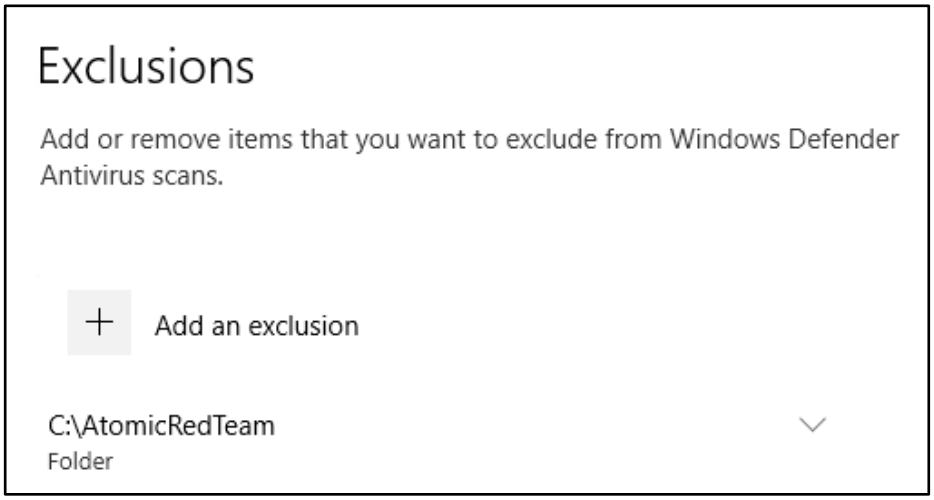


If we review the Virus and Threat Protection settings for Windows Defender we can see some needed files being blocked, quarantined and/or deleted. In our lab environment, we are going to exclude the atomics folder from

being scanned by Windows Defender so that we are able to run all of the atomic tests.

On the start menu search for “Virus and Threat Protection” and launch. Under “Virus & threat protection settings” click “Manage settings”

Add an exclusion for the C:\AtomicRedTeam folder.



An alternative way to quickly add this exclusion from the PowerShell command line is given below for convenience but must be run from an administrative PowerShell prompt.

```
Add-MpPreference -ExclusionPath C:\AtomicRedTeam\
```

Now, run the installation command one more time to re-download all the atomic files that may have been blocked or removed by Windows Defender.

```
IEX (IWR
'https://raw.githubusercontent.com/redcanaryco/invoke-
atomicredteam/master/install-atomicredteam.ps1' -
UseBasicParsing);
Install-AtomicRedTeam -getAtomics -Force
```

```
powershell
PS C:\Users\art> IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/
invoke-atomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);

PS C:\Users\art> Install-AtomicRedTeam -getAtomics -Force
Installation of Invoke-AtomicRedTeam is complete. You can now use the Inv
oke-AtomicTest function
See Wiki at https://github.com/redcanaryco/invoke-atomicredteam/wiki for
complete details
```

Occasionally, Windows Defender decides to block the install at this point. If this happens, use the instructions linked below to completely disable Windows Defender for the installation.

[Disable Windows Defender]

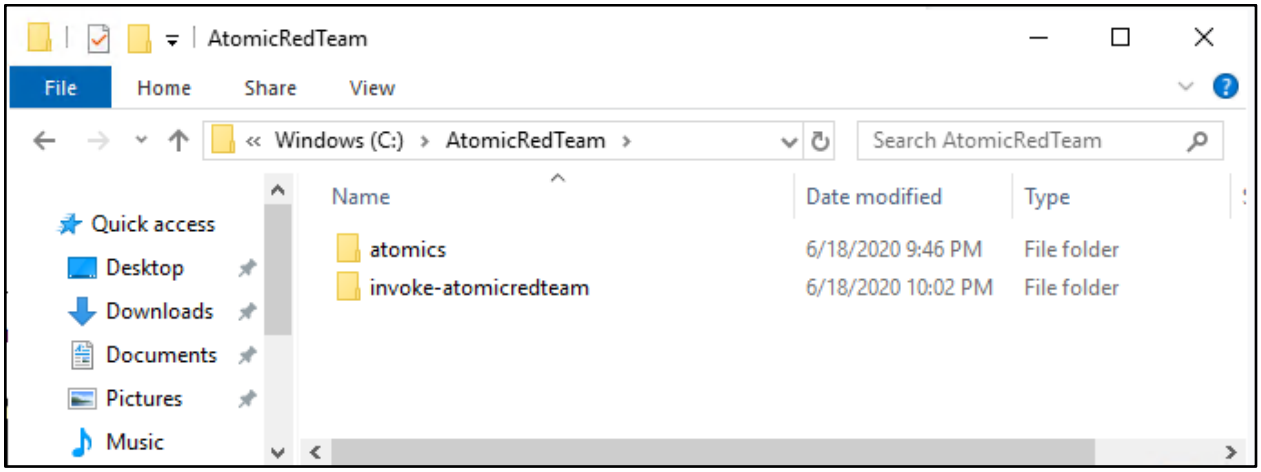
To make sure everything was installed correctly, let's run the "Get-Module" PowerShell command. You should see the two items highlighted in red.

```
Select powershell
PS C:\Users\art> Get-Module

ModuleType Version    Name


Script     0.0        AtomicClassSchema
Script     1.0.0.0    Invoke-AtomicRedTeam
Manifest   1.0.1.0    Microsoft.PowerShell.Archive
Manifest   3.1.0.0    Microsoft.PowerShell.Management
Manifest   3.1.0.0    Microsoft.PowerShell.Utility
Binary     1.0.0.1    PackageManagement
Script     1.0.0.1    PowerShellGet
Script     2.0.0      PSReadline
```

You can also validate that you have the “atomics” and “invoke-atomicredteam” folders in the default installation folder of C:\AtomicRedTeam.



We purposefully gave instructions to let you run into the common hurdles encountered during the installation. We could expedite the installation in the future by ensuring we bypass the execution policy and by adding the C:\AtomicRedTeam folder to the Windows Defender exceptions list before the installation attempt.

Atomic Red Team and the Execution Framework are now installed and ready for use. You can leave the PowerShell prompt as is until the next lab.

End of Lab