RSAConference2022
San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **OST-R06**

Atomic Red Team: Where Adversary Emulation and EDR Testing Meet

Adam Mashinchi

Director, Open Source Programs
Red Canary

Twitter: @Adam_Mashinchi

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

Presentation content created and owned by Red Canary.

Presentation Goals & Terms

The key takeaways in ~50 mins

Overview

- Specific takeaways:
  - gain an entry-level understanding of the project
  - use cases for red/blue/purple teams
  - differentiate techniques from tests
  - understand execution frameworks
  - execute a test
  - know how to learn more!

Overview: General Terminology

- MITRE: a not-for-profit organization (government) R&D
- ATT&CK®: a grid of adversary tactics and techniques
- adversary: the “who” performing malicious actions
- technique: how adversaries achieve tactical goals
- test: what command, script, or application is executed

Overview: The ATT&CK matrix

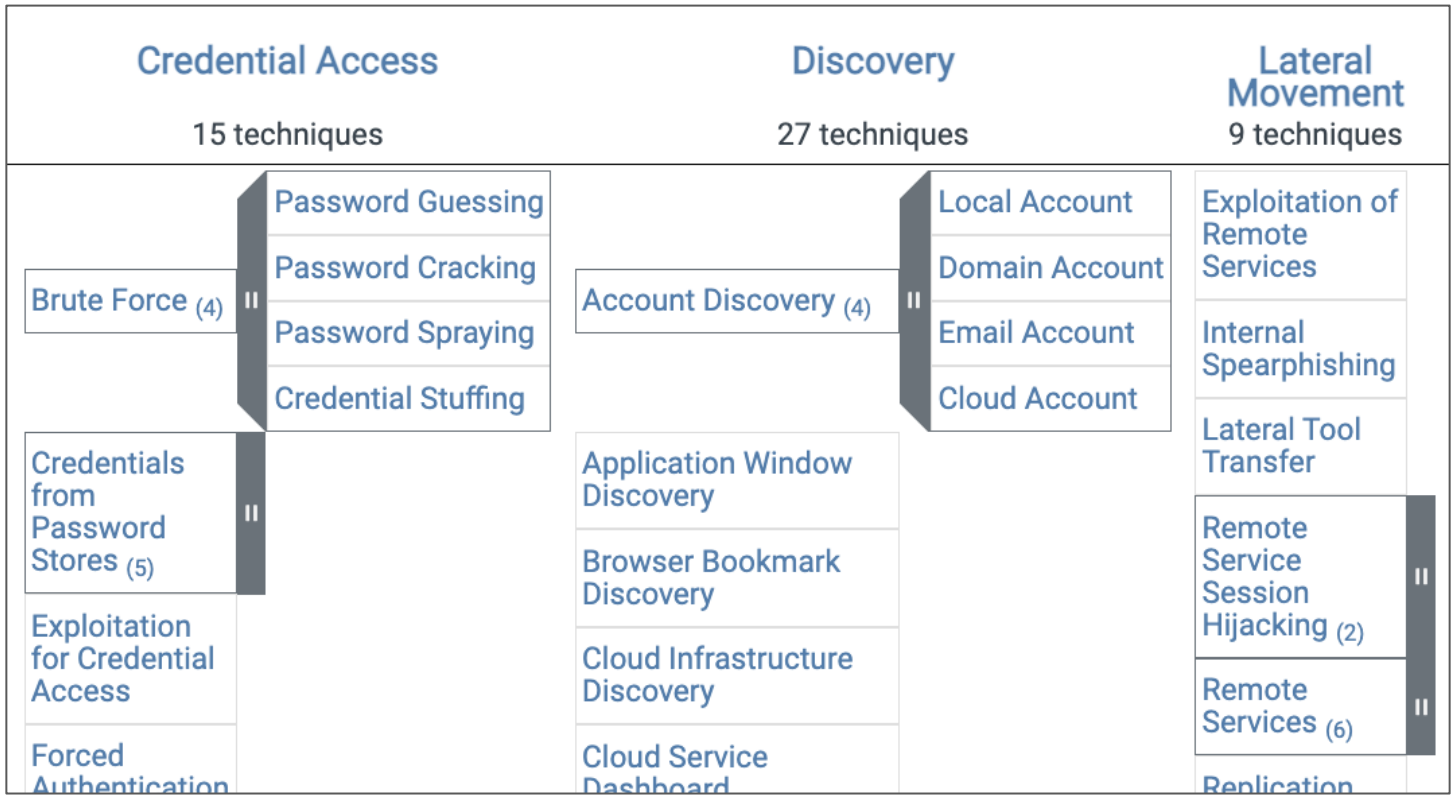
Reconnaissance (10 techniques)	Resource Development (7 techniques)	Initial Access (9 techniques)	Execution (12 techniques)	Persistence (19 techniques)	Privilege Escalation (13 techniques)	Defense Evasion (39 techniques)	Credential Access (15 techniques)	Discovery (27 techniques)	Lateral Movement (9 techniques)	Collection (17 techniques)	Command and Control (16 techniques)	Exfiltration (9 techniques)	Impact (13 techniques)
Active Scanning (2)	Acquire Infrastructure (6)	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (5)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Infrastructure (6)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Clipboard Data	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (3)	Inter-Process Communication (2)	Browser Extensions	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Dashboard	Remote Services (6)	Data from Cloud Storage Object	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification (2)	Deploy Container	Input Capture (4)	Cloud Service Discovery	Replication Through Removable Media	Data from Configuration Repository (2)	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Stage Capabilities (5)	Supply Chain Compromise (3)	Scheduled Task/Job (7)	Create Account (3)	Escape to Host	Direct Volume Access	Man-in-the-Middle (2)	Container and Resource Discovery	Software Deployment Tools	Data from Information Repositories (2)	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)		Trusted Relationship	Shared Modules	Create or Modify System Process (4)	Event Triggered Execution (15)	Domain Policy Modification (2)	Modify Authentication Process (4)	Domain Trust Discovery	Taint Shared Content	Data from Local System	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (2)		Valid Accounts (4)	Software Deployment Tools	Event Triggered Execution (15)	Exploitation for Privilege Escalation	Execution Guardrails (1)	Network Sniffing	File and Directory Discovery	Use Alternate Authentication Material (4)	Data from Network Shared Drive	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			System Services (2)	External Remote Services	Hijack Execution Flow (11)	Exploitation for Defense Evasion	OS Credential Dumping (8)	Network Service Scanning		Data from Removable Media	Non-Application Layer Protocol		Network Denial of Service (2)
			User Execution (3)	Hijack Execution Flow (11)	Process Injection (11)	File and Directory Permissions Modification (2)	Steal Application Access Token	Network Share Discovery		Data Staged (2)	Non-Standard Port		Resource Hijacking
			Windows Management Instrumentation	Implant Internal Image	Scheduled Task/Job (7)	Hide Artifacts (7)	Steal or Forge Kerberos Tickets (4)	Network Sniffing		Email Collection (3)	Protocol Tunneling		Service Stop
				Modify Authentication Process (4)	Valid Accounts (4)	Hijack Execution Flow (11)	Steal Web Session Cookie	Password Policy Discovery		Input Capture (4)	Proxy (4)		System Shutdown/Reboot
				Office Application Startup (6)		Impair Defenses (7)	Two-Factor Authentication Interception	Peripheral Device Discovery		Man in the Browser	Remote Access Software		
				Pre-OS Boot (5)		Indicator Removal on Host (6)	Unsecured Credentials (7)	Permission Groups Discovery (3)		Man-in-the-Middle (2)	Traffic Signaling (1)		
				Scheduled Task/Job (7)		Indirect Command Execution		Process Discovery		Screen Capture	Web Service (3)		
				Server Software Component (3)		Masquerading (6)		Query Registry		Video Capture			
				Traffic Signaling (1)		Modify Authentication Process (4)		Remote System Discovery					
				Valid Accounts (4)		Modify Cloud Compute Infrastructure (4)		Software Discovery (1)					
						Modify Registry		System Information Discovery					
						Modify System Image (2)		System Location Discovery					
						Network Boundary Bridging (1)		System Network Configuration Discovery (1)					

New Industry Standard & Common Language

Columns are “Tactics”

ATOMIC RED TEAM

Overview: ATT&CK sub-techniques



The "Cells" in ATT&CK are "Techniques"

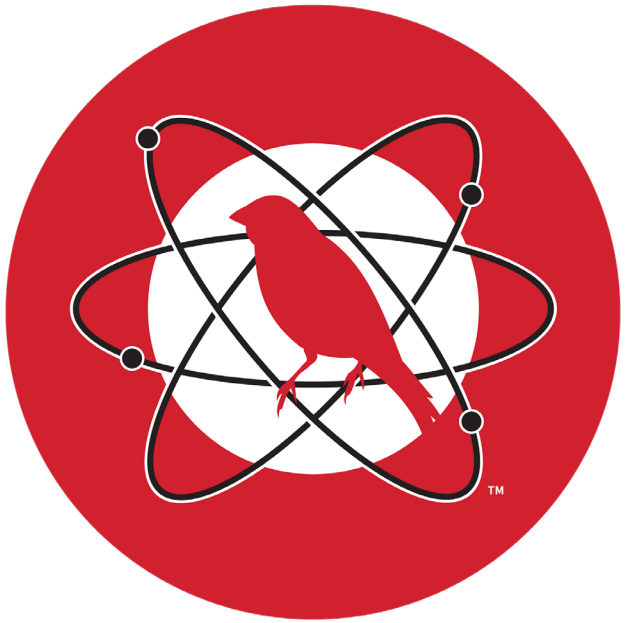
Parent/Child Relationship → Sub-Techniques

ATOMIC RED TEAM

Overview: Atomic Red Team

Atomic Red Team™ is a **library of simple tests** that any security team can execute to test their defenses.

Tests are **focused**, have few dependencies, and are defined in a structured format that can be **used by automation** frameworks.



ATOMIC RED TEAM

Who works on the (free) stuff?

Project Maintainers & Contributors

- These are *amazing* people!!!
  - *"Thanks for everything you do!"*
- Great at small/iterative changes
- Atomic tests are community developed & maintained

Shameless Plug:

- These are *amazing* people!!!
- If you contribute to Atomic Red Team, you get a t-shirt!
- Learn more at atomicredteam.io!

ATOMIC RED TEAM

RSA Conference 2022

A brief history of Atomic Red Team

When, where, and why we created the library

History: DerbyCon 2017

Presentation

"Blue Team Keeping Tempo with Offense"

- endpoint > network telemetry
- introduced "Atomic Testing"
  - small unit-tests for ATT&CK

ATOMIC RED TEAM

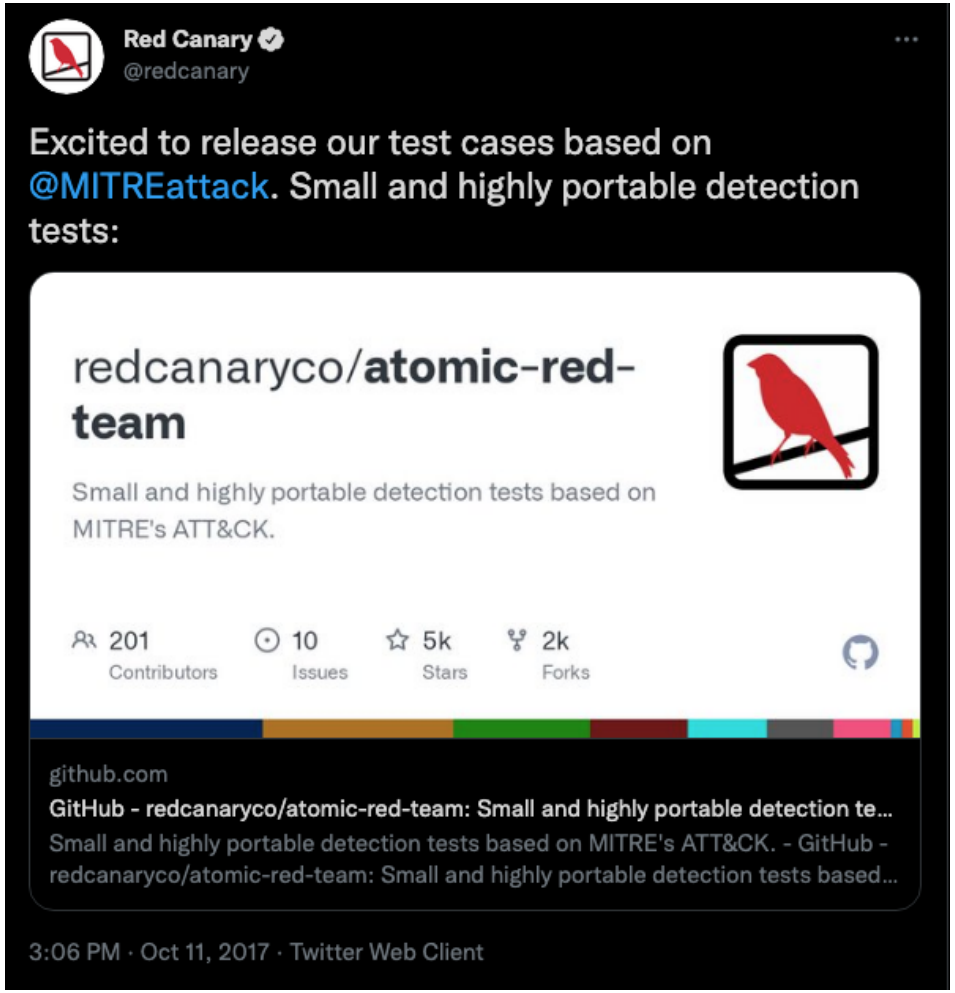
History: A repository is born

GitHub Repository is created

- open source project
- MIT licensed
  - TL;DR: “Anyone can use this for anything”

Original repository structure:

- operating system →
  - list of ATT&CK tactics →
    - tests



ATOMIC RED TEAM

History: Migration to YAML

April 2018

- scripts converted to YAML
  - … YAML is a common data serialization language
- focus on techniques

29 lines (29 sloc) | 865 Bytes

```yaml
attack_technique: T1124
display_name: System Time Discovery
atomic_tests:
- name: System Time Discovery
  auto_generated_guid: 20aba24b-e61f-4b26-b4ce-4784f763ca20
  description: |
    Identify the system time. Upon execution, the local computer system time and timezone will be displayed.
  supported_platforms:
  - windows
  input_arguments:
    computer_name:
      description: computer name to query
      type: String
      default: localhost
  executor:
    command: |
      net time \\#{computer_name}
      w32tm /tz
    name: command_prompt
- name: System Time Discovery - PowerShell
  auto_generated_guid: 1d5711d6-655c-4a47-ae9c-6503c74fa877
  description: |
    Identify the system time via PowerShell. Upon execution, the system time will be displayed.
  supported_platforms:
  - windows
  executor:
    command: |
      Get-Date
    name: powershell
```

Interlude

Businesses creating open source

Obligations & implications

Red Canary

- created Atomic Red Team
- manages where the code lives
- provides support
  - staffing
  - goods
  - financial

The Community

- writes the tests
- “maintains” the code base

Other Vendors

- can use freely
- add to their software

Open Source Community

The maintainers team

- Community members
  - not all Red Canary staff!
- Oversee the project
- Review/approve all additions
- Assisted by Red Canary's Open Source Programs Team
  - act as product/project managers
  - facilitates communications, etc.

Bhavin Patel

Slack: Bhavin Patel
GitHub: patel-bhavin

Carl Petty

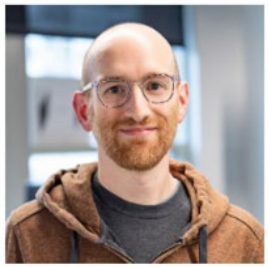
Slack: Carl Petty
GitHub: rc-grey

Carrie Roberts

Slack: OrOneEqualsOne
GitHub: clr2of8

Jose Hernandez

Slack: Jose Hernandez
GitHub: d1vious



Matt Graeber

Slack: mattifestation
GitHub: mattifestation

Mike Haag

Slack: Mike Haag
GitHub: MHaggis

Breaking down ...

… the (atomic) tests

Use cases for atomics (in InfoSec terms)

Red team (offense)

- simulate a variety of diverse threat behaviors

Blue team (defense)

- tests for detections, signatures, and behaviors

Purple team (joint exercises)

- regression/unit test framework

T1124: System Time Discovery

System Time Discovery

"An adversary may gather the system time and/or time zone from a local or remote system."

The atomic test

```
> powershell Get-Date
```

The How to execute this

Windows → cmd.exe →

```
powershell Get-Date
```

Windows → powershell.exe →

```
Get-Date
```

Linux/macOS → bash/zsh →

```
date
```

Defining ATT&CK "coverage"

A (naive) approach to "breadth" coverage…

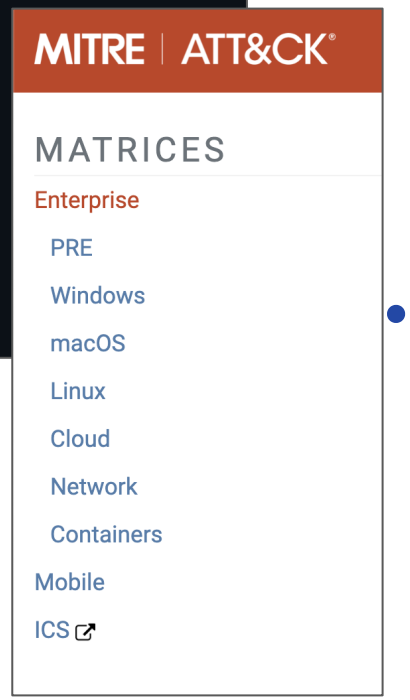
- Atomic test for technique ID? → "Done!"
- Breakdown by "all" and each platform

But what about "depth"?

- How *well* does a group of tests cover a technique?
- How difficult is a test to execute?
- What about "sub-platforms"?
  - (i.e., IaaS vs. IaaS:AWS)

Coverage is non-trivial to map

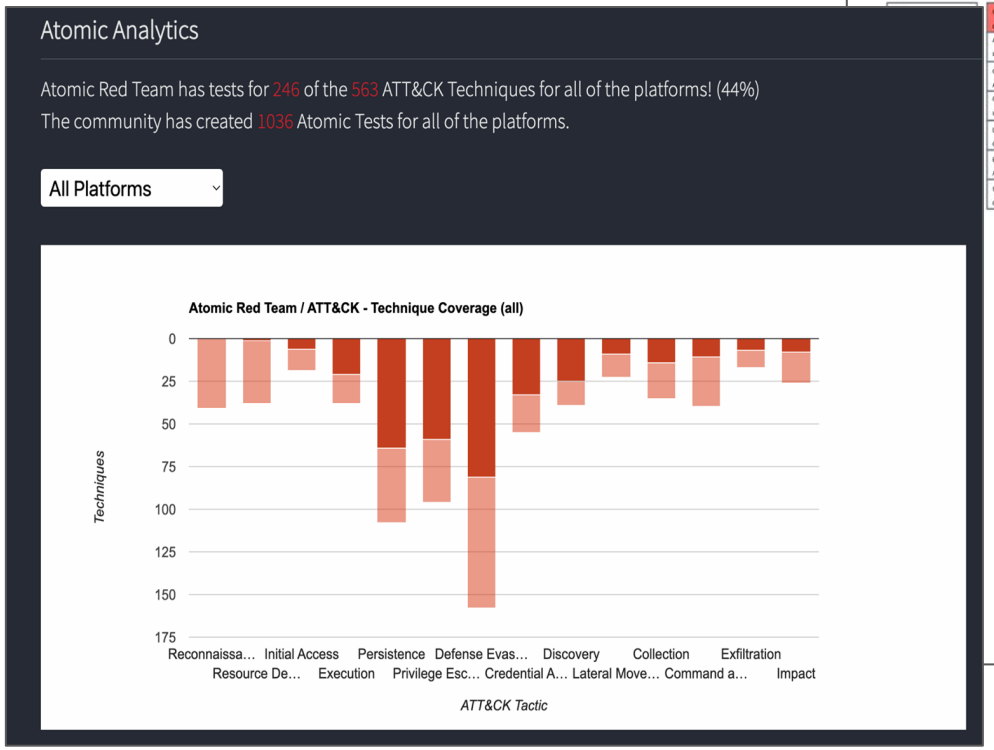
```
# supported platforms is an array of the OS/platforms this atomic test can be run upon. Values include:
# - windows
# - macos
# - linux
# - office-365
# - azure-ad
# - google-workspace
# - saas
# - iaas
# - containers
# - iaas:gcp
# - iaas:azure
# - iaas:aws
```

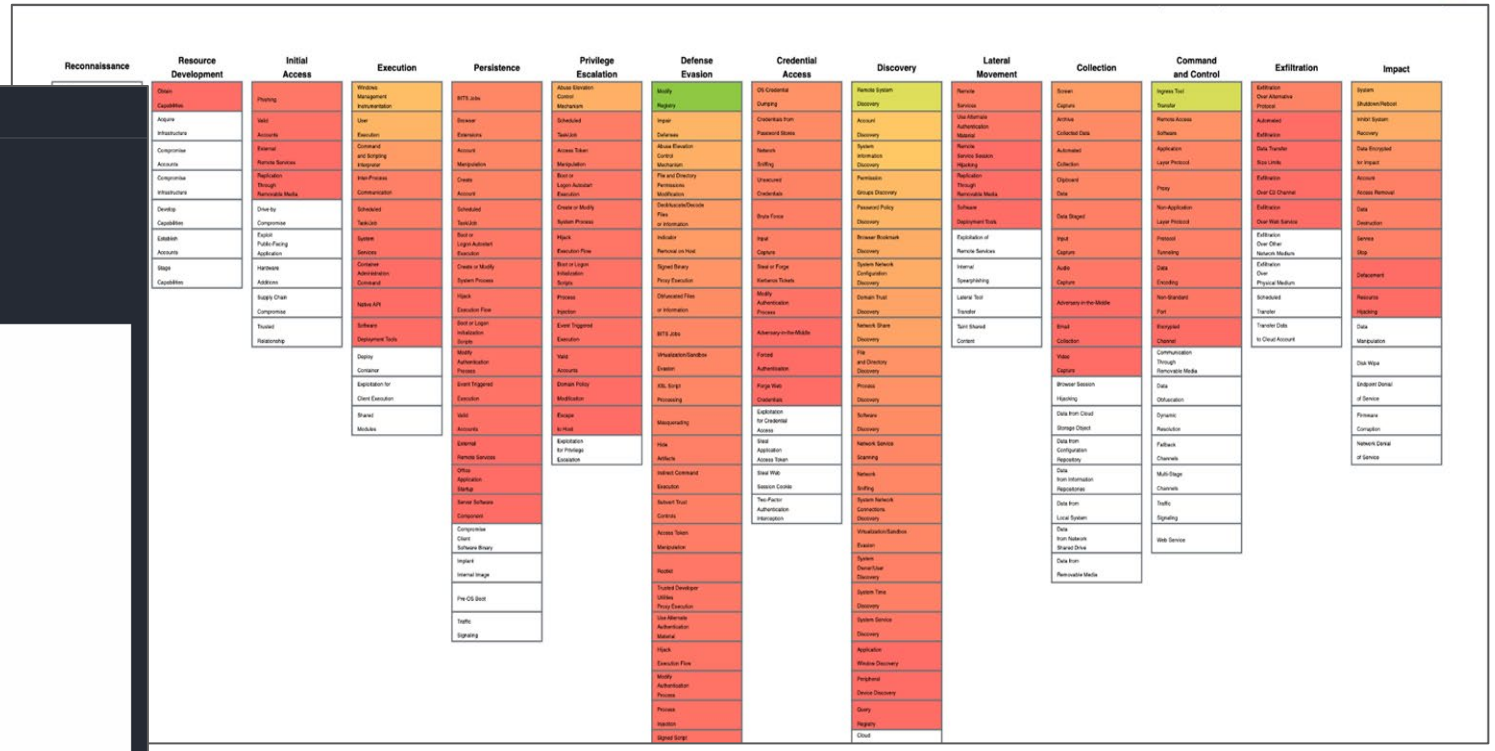


- Even “counting” can be tricky to master
- Defining “platform”
  - easy to confuse
  - where test is running?
  - target of the test?
  - … vs. Executor

ATOMIC RED TEAM

Visualization is critical



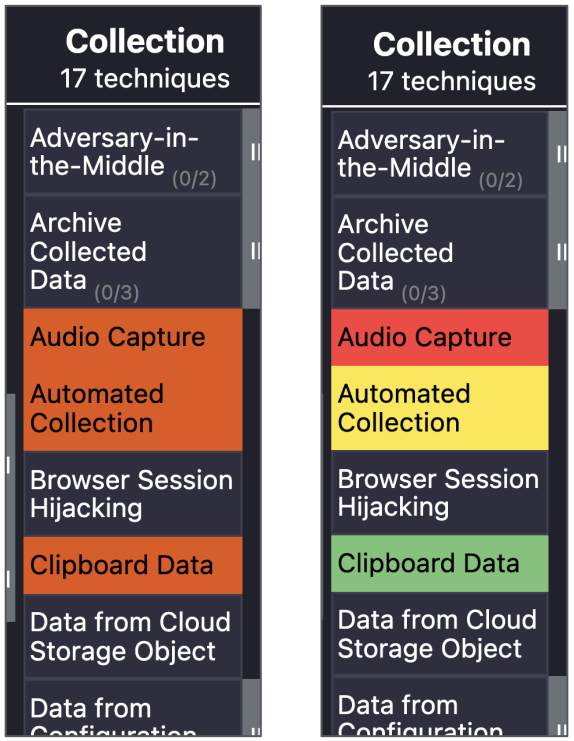


- Heatmaps create instantly-understandable coverage documents

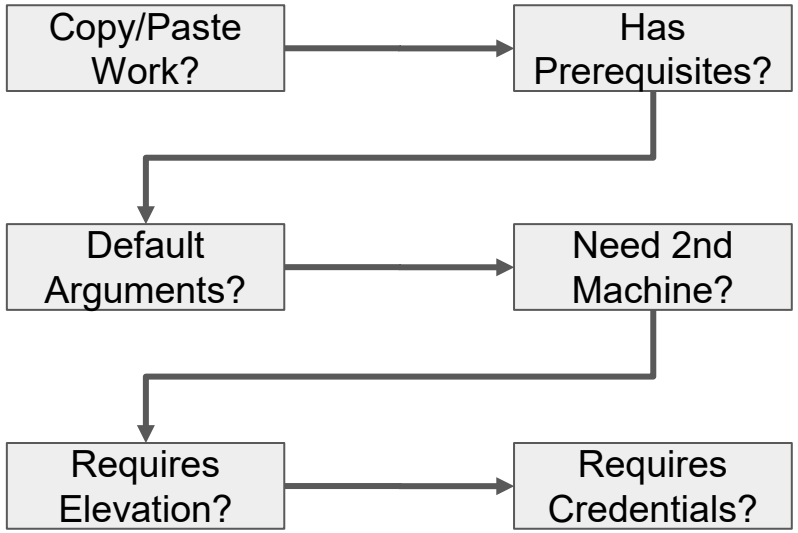
ATOMIC RED TEAM

Coverage: Depth Analysis

- How “well coverage” is a given TID?
- We know if…
  - we have test(s), how many, by platform
- But what if…
  - T9100 has 14 tests, but only 2 for Linux?
  - T9200 has 14 tests, but all are really tricky?
- And to further complicate things…
  - “System Time Discovery” ← Easy?
  - “Network Sniffing” ← Less-Easy?
  - “Cloud Storage Object Discovery” ← ???
    - Prerequisites & Paid Providers

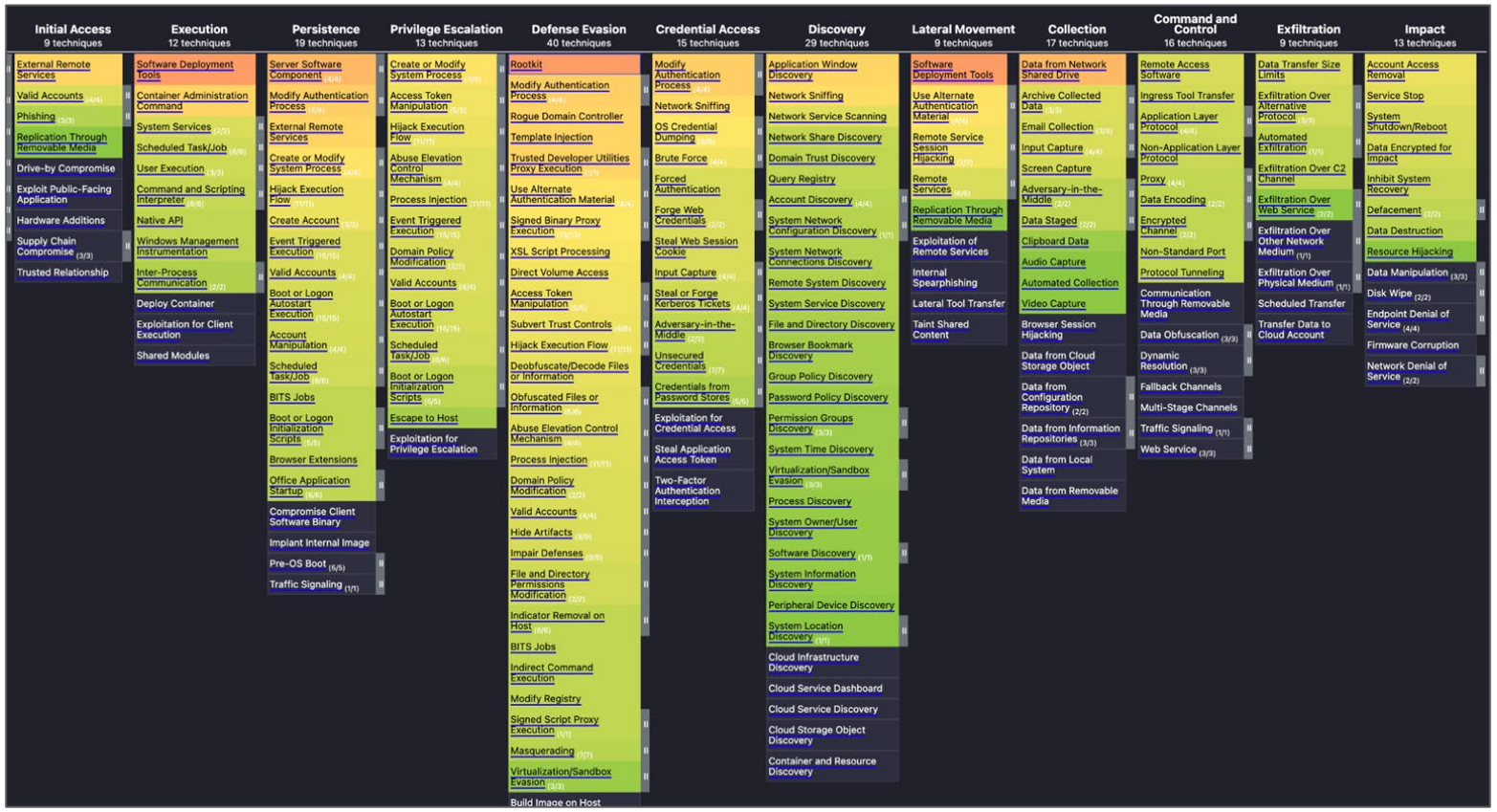


Can Define a "Scale" of Difficulty



- Define a scale of "difficulty"
- MUST BE:
  - Strictly defined
  - Human actionable
- Bonus Round:
  - Machine parsable
  - Test execution (not here yet)

Heatmaps for (Difficulty) Coverage



Indicator of where tests need to be added!

ATOMIC RED TEAM

RSAConference2022

Execution frameworks

Actioning tests with “ease”

What & why

The problem

- a large amount of tests
- a need to execute them in bulk/often

The solution

- execution frameworks
  - A program to assist with the execution of the atomic tests in a consistent manner
- automation

The Atomic Family

Invoke-AtomicRedTeam

- a PowerShell-based framework for developing and executing atomic tests

AtomicTestHarnesses

- a PowerShell module for executing variations of an attack technique

Chain-Reactor

- a tool for test detection and response coverage on Linux

Third-party frameworks

Vendors

- breach and attack simulation (BAAS)
- endpoint detection and response (EDR)

Open source community

- General use (i.e., “atomic-operator”)
- Specific use (i.e., “atomic confetti bomb”)

A demonstration

See an atomic test in action

Choosing a technique

Common question: "*Where should I start?*"

- many options of places to start
- let's look at adversaries are doing

The Threat Detection Report

- Technique: **Powershell**
- Usage: Encoding/**Obfuscation**

TECHNIQUE T1059.001

PowerShell

PowerShell was the most common technique we observed in 2020, affecting nearly half of our customers. It remains among the most versatile of built-in utilities for adversaries, defenders, and system administrators alike.

#1
PARENT TECHNIQUE RANK

48.7%
ORGANIZATIONS AFFECTED

2,366
CONFIRMED THREATS

ATOMIC RED TEAM

DEMONSTRATION

- Pick a technique
- Find or create an atomic test
- Copy & paste command
  - … or use an Execution Framework
- See results on an endpoint
- (Hopefully) see an alert

18 lines (18 sloc) | 964 Bytes

```yaml
attack_technique: T1059.001
display_name: 'Command and Scripting Interpreter: PowerShell'
atomic_tests:
- name: PowerShell
  auto_generated_guid: a538de64-1c74-46ed-aa60-b995ed302598
  description: |
    PowerShell was the most common technique we observed in 2020, affecting nearly half of our customers.
  supported_platforms:
  - windows
  input_arguments:
    obfuscated_code:
      description: 'Defaults to: Invoke-Expression with a "Write-Host" line.'
      type: string
      default: JgAgACgAZwBjAG0AIAAoACcAaQBlAHsAMAB9ACcAIAAtAGYAIAAnAHgAJwApACkAIAAoACIAVwByACIAKwAiAGkAdA/
  executor:
    command: |
      powershell.exe -e  #{obfuscated_code}
    name: command_prompt
```

ATOMIC RED TEAM

Apply what you have learned today!

- Next week you should:
  - See if you are doing any defense validation now
  - Evaluate your cyber security maturity
- In the next three months you should:
  - Try out validating your defenses & defenders
  - Select some threat-aligned techniques to test
  - Attempt some (manual) tests using Atomic Red Team
- Within six months you should:
  - Evaluate the use of automation
  - Review some breach & attack simulation/emulation options

ATOMIC RED TEAM

Extra Credit: There's a (known) problem…

- How can you execute Atomics in-bulk if you…
  - Have no concept if the Atomic "succeeded" or "failed"
  - Cannot confirm you're generating the correct telemetry
- Example: System Time Discovery (T1124)
  - What does "success" look like for: `> date`
  - An exit code? A string? What about not-US formats?
- The Atomic Red Team Maintainers are open to suggestions!
  - Specifications can grow and be modified
  - Projects need to adapt over time

ATOMIC RED TEAM

Atomic Red Team is community driven

- Want to **learn** more?
  - Web: atomicredteam.io
- Care to **contribute**?
  - Github: github.com/redcanaryco/atomic-red-team
- Have **Feedback**?
  - Email: opensource@redcanary.com

ATOMIC RED TEAM

Use AtomicRedTeam.io to explore Atomics!

Explore Atomic Red Team

Learn More Atomics Tags Newsletter Join the Slack

Explore Atomic Red Team

This site is designed to help you explore and navigate the Atomic Red Team™ library of tests, as they are mapped to the MITRE ATT&CK® framework and the platforms they support.

Learn More | Start Exploring

Explore Atomic Red Team

Get started

Start navigating atomic tests by ATT&CK Technique, platform/os, or via search at the top of the window.

You can also see some ATT&CK coverage statistics here!

Learn more

The Atomic Red Team documentation is available as a wiki.

For information and descriptions of the Atomic Red Team family of projects visit the Learn More page.

ATOMIC RED TEAM

Use AtomicRedTeam.io to explore coverage!

Explore Atomic Red Team

Learn More Atomics Tags Newsletter Join the Slack

Explore Atomic Red Team

This site is designed to help you explore and navigate the Atomic Red Team™ library of tests, as they are mapped to the MITRE ATT&CK® framework and the platforms they support.

Learn More | Start Exploring

Explore Atomic Red Team

Get started

Start navigating atomic tests by ATT&CK Technique, platform/os, or via search at the top of the window.

You can also see some ATT&CK coverage statistics here!

Learn more

The Atomic Red Team documentation is available as a wiki.

For information and descriptions of the Atomic Red Team family of projects visit the Learn More page.

ATOMIC RED TEAM

Don’t just take my word for it…

- Anyone can be a contributor
  - New tests, fixes, typos, are welcome
  - You get free stuff!
  - New & Top Contributors listed in monthly Atomic Newsletter
- Become part of the Slack community!

https://atomicredteam.io

ATOMIC RED TEAM

RSA Conference2022

Thank you!

(Time for Q&A?)