

Atomic Red Team Adversary Simulation Lab — Technical Documentation

By: Ritvik Indupuri

Overview

This project focuses on simulating adversary techniques from the **MITRE ATT&CK framework** using **Atomic Red Team (ART)** in a **controlled Windows lab environment**. The objective is to execute and analyze real-world attack techniques, replicating adversary behaviors in a safe, isolated setup for practical Red Team learning.

Objective

- Simulate MITRE ATT&CK Techniques **T1553.005 (ISO Payload Execution)** and **T1016 (Network Configuration Discovery & Egress Port Enumeration)**.
 - Leverage **Atomic Red Team (ART)** modules to execute adversary behaviors.
 - Automate attack scenarios using **PowerShell scripts**.
 - Document execution flow, attack mappings, and results.
-

Lab Environment Setup

- **Host OS:** Windows 10 (Lab Machine)
 - **Atomic Red Team (ART):** Cloned from official GitHub repository.
 - **Execution Tools:** PowerShell 5.1+
 - **Network:** Isolated virtual network to ensure safe adversary simulation.
-

Attack Simulation Techniques

1. T1553.005 — ISO Payload Execution

Simulated adversary technique where a **malicious payload is executed from a mounted ISO image**.

Execution Steps:

1. Mount ISO image containing payload (**hello.exe**).
2. Execute payload directly from the mounted drive.
3. Observe execution behavior within the controlled lab.

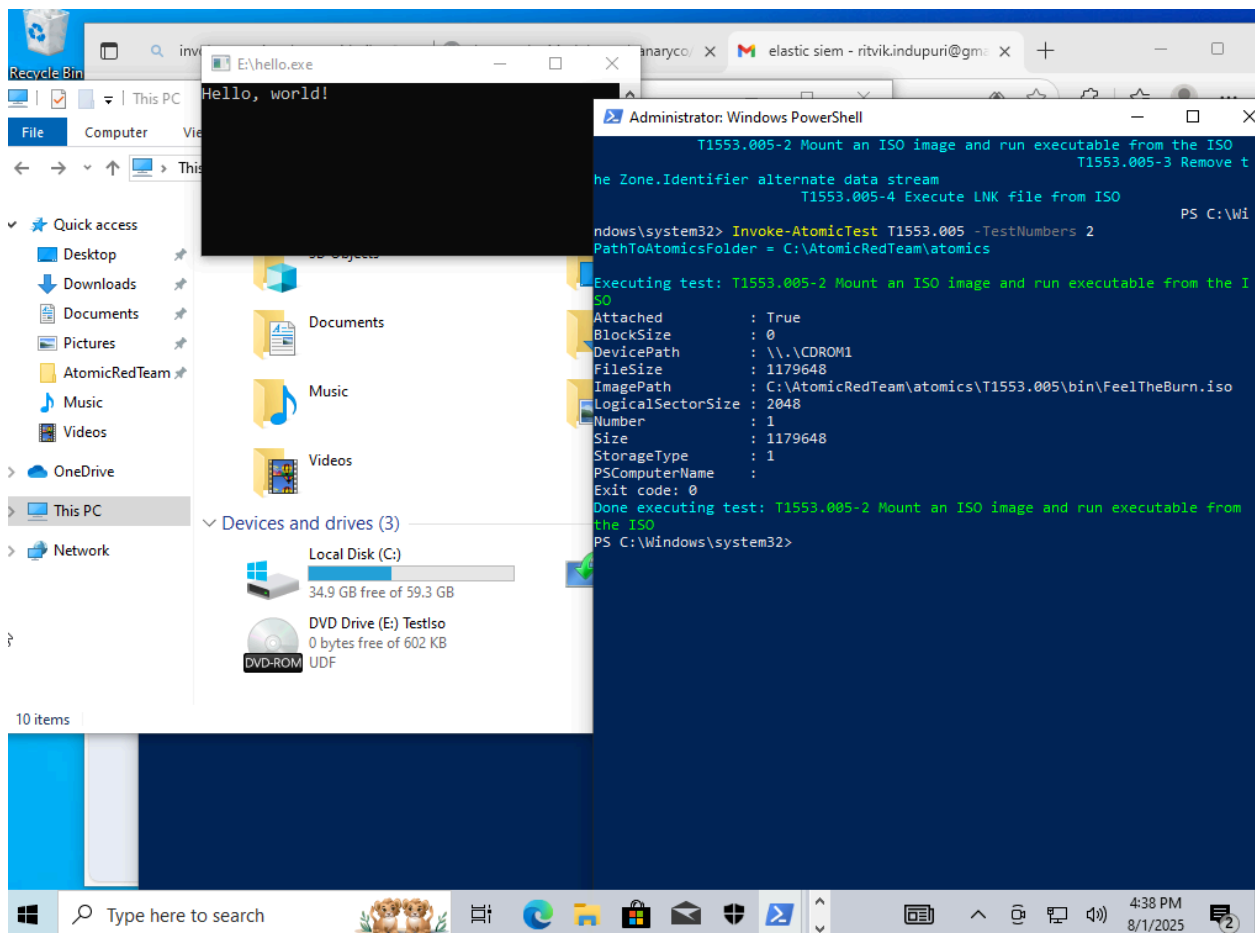


Figure 1 — ISO Payload Execution (T1553.005)

Demonstrates the execution of **MITRE ATT&CK Technique T1553.005** by mounting an ISO image and running a simulated malicious payload (**hello.exe**) using Atomic Red Team in a controlled Windows lab environment.

2. T1016 — Network Configuration Discovery

Simulated adversary technique to enumerate **system network configurations** as part of reconnaissance.

Execution Steps:

1. Run Atomic Red Team T1016 module.
2. Execute network discovery commands (`ipconfig /all`, `net view`, `nltest`).
3. Collect output to observe enumeration success.

```
PS C:\Windows\system32> Invoke-AtomicTest T1016 -TestNumbers 1
PathToAtomicFolder = C:\AtomicRedTeam\atomics

Executing test: T1016-1 System Network Configuration Discovery on Windows
Windows IP Configuration
Host Name . . . . . : DESKTOP-PI2SRBR
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : localdomain

Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : localdomain
Description . . . . . : Intel(R) 82574L Gigabit Network Connection
Physical Address. . . . . : 00-0C-29-F2-9C-08
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::8f50:573b:6e3d:de6f%5(Preferred)
IPv4 Address. . . . . : 192.168.13.135(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Friday, August 1, 2025 3:20:14 PM
Lease Expires . . . . . : Friday, August 1, 2025 5:35:21 PM
Default Gateway . . . . . : 192.168.13.2
DHCP Server . . . . . : 192.168.13.254
DHCPv6 IAID . . . . . : 100666409
DHCPv6 Client DUID. . . . . : 00-01-00-01-30-1E-B1-F8-00-0C-29-F2-9C-08
DNS Servers . . . . . : 192.168.13.2
Primary WINS Server . . . . . : 192.168.13.2
NetBIOS over Tcpip. . . . . : Enabled

-----
Admin State      State          Type           Interface Name
-----
Enabled          Connected      Dedicated      Ethernet0
Interface: 192.168.13.135 --- 0x5
Internet Address  Physical Address  Type
192.168.13.2      00-50-56-e3-6a-ff  dynamic
192.168.13.254    00-50-56-fb-6e-ac  dynamic
192.168.13.255    ff-ff-ff-ff-ff-ff  static
224.0.0.22        01-00-5e-00-00-16  static
224.0.0.251       01-00-5e-00-00-fb  static
224.0.0.252       01-00-5e-00-00-fc  static
239.255.255.250   01-00-5e-7f-ff-fa  static
255.255.255.255   ff-ff-ff-ff-ff-ff  static

Ethernet0:
Node IpAddress: [192.168.13.135] Scope Id: []
NetBIOS Local Name Table
-----
Name          Type          Status
-----
```

Figure 2 — Network Configuration Discovery Output (T1016)

Displays the output of **MITRE ATT&CK Technique T1016** executed via Atomic Red Team, showing detailed network configuration enumeration.

3. T1016 — Egress Port Enumeration & Execution Flow

Detailed simulation of **egress port scanning** and network enumeration using PowerShell scripting logic aligned with ATT&CK T1016.

Execution Breakdown:

- The **left panel** showcases MITRE ATT&CK technique IDs, descriptions, and mapped attack commands.
- The **right panel** details the PowerShell scripting logic used to simulate port enumeration, control flow, and result handling.

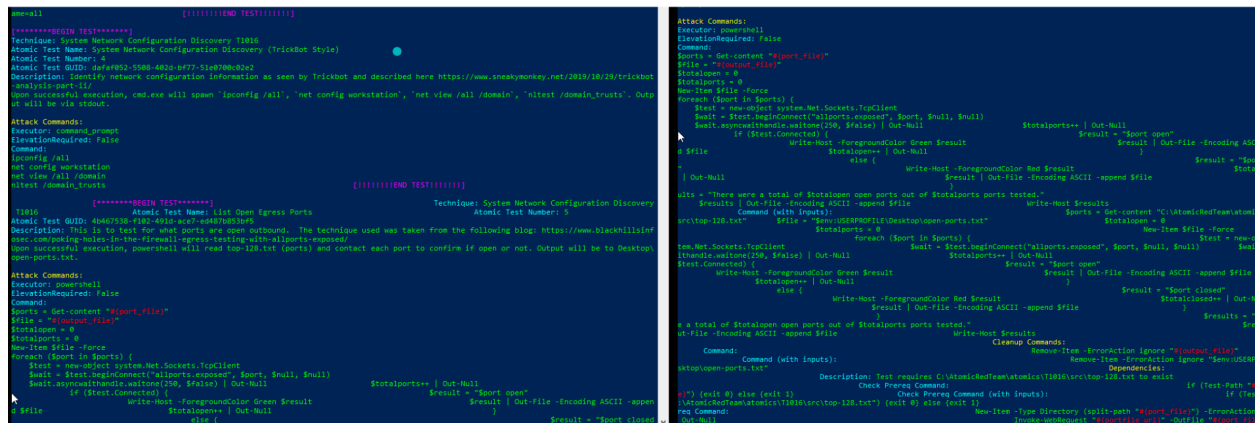


Figure 3 — MITRE T1016 Technique Mapping & PowerShell Execution Flow

Displays the **MITRE ATT&CK T1016 adversary simulation**, with the left panel showing the technique ID, descriptions, and mapped attack commands for **Network Configuration Discovery** and **Egress Port Enumeration**, while the right panel presents the detailed PowerShell script execution logic. This illustrates how ATT&CK techniques are operationalized through scripting in a Red Team lab environment.

Project Outcomes

- Successfully executed MITRE ATT&CK techniques **T1553.005** and **T1016** using Atomic Red Team modules.
- Automated adversary simulations through PowerShell to replicate real-world attack behaviors.
- Designed and maintained a secure, isolated lab environment to safely conduct Red Team adversary simulations.
- Developed a structured documentation approach to align MITRE techniques with execution workflows.

Skills Demonstrated

- MITRE ATT&CK Framework
 - Atomic Red Team (ART) Execution
 - PowerShell Scripting & Automation
 - Red Team Adversary Simulation
 - Lab Environment Setup & Isolation
-

Next Steps (Planned Enhancements)

- Integrate SIEM solution (Elastic Stack) for detection validation.
 - Expand simulations to include lateral movement and privilege escalation techniques.
-

Appendix — Image List

Figure No.	Image Title
Figure 1	ISO Payload Execution — T1553.005 Adversary Simulation
Figure 2	Network Configuration Discovery — T1016 Simulation Output
Figure 3	MITRE ATT&CK T1016 Technique Mapping & PowerShell Execution Flow
