Microsoft 365 Defender simulation

Atomic Red Team 2021 Threat Detection Report

November 2021

Summary

Red Canary's 2021 Threat Detection Report enumerates the most prevalent adversary techniques and offers custom Atomic Red Team™ tests to simulate those techniques. This simulation is a compilation of those tests, which comprise adversary behaviors that have been converted into individual atomic commands.

This simulation generates suspicious process starts, command-line execution events, arbitrary file downloads, and more.

Credential Access	LSASS memory dump
Defense Evasion	Rename system utilities
Execution	Scheduled task
Privilege Escalation	Process injection
Execution	PowerShell
Execution	Windows command shell
Command and Control	Ingress tool transfer
Defense Evasion	Rundll32
Persistence	Windows service

Running the simulation

> Note: We recommend running this simulation in a testing environment such as the Evaluation lab. You can also run the simulation on any other device of your choice.

Use the following steps to run the simulation in the evaluation lab:

1. Navigate to **Evaluation & tutorials** > **Evaluation lab**.
2. Go to the **Devices** tab and select **Add Device**. Follow the wizard to create a Windows device. Save the provided credentials.
3. Once the device is ready, click the Actions button (⋮), then click **Connect**, and connect to the device.
4. Download the simulation script, unzip it, copy it to the device and run it.

Simulation steps

LSASS memory dump MITRE T1003.001

In this test, “rundll32.exe” will write a full “lsass.exe” process dump to “%windir%\Temp\lsass.dmp” using MiniDump.

> Investigation tip: search for alerts named “**Process memory dump**” and “**’RundllLolBin’ malware was prevented**”.

Rename system utilities MITRE T1036.003

This test executes an encoded PowerShell command while masquerading as “notepad.exe” which is a copy of PowerShell placed in the “ %windir%\Temp\” directory.

> Investigation tip: Search for a Microsoft Defender for Endpoint alert named “**System executable renamed and launched**”. In addition, you may find an alert from Microsoft Defender Antivirus, named “**Suspicious ‘MasqSuspiciousProcessLaunched’ behavior was blocked”**.

Scheduled task MITRE T1053.005

Running this test will create a scheduled task named “CMDTestTask” that runs “cmd.exe” every three minutes. This scheduled task is then deleted.

> Investigation tip: search for an alert named “**Suspicious Task Scheduler activity**”.

Process injection MITRE T1055

In this test, “mavinject.exe” will inject “vbscript.dll” into the running “lsass.exe” process using Dynamic-link Library Injection.

> Investigation tip: search for an alert named “**A process was injected with potentially malicious code**”.

PowerShell MITRE T1059.001

Running this test should print “Hello, from PowerShell!” to the terminal via an obfuscated command.

> Investigation tip: Using PowerShell to display a message onto a screen is considered, by itself, to be benign, and may not trigger an alert. However, when performed along with other potentially malicious commands, Microsoft Defender for Endpoint considers the command’s context and may trigger a behavioral alert for a “**Suspicious PowerShell command line**”.

Windows command shell MITRE T1059.003

This test obfuscates “cmd.exe”, writes “Hello, from CMD!” to “hello.txt”, and then displays it.

> Investigation tip: Using cmd.exe to display a message to screen is considered benign, and therefore will not trigger an alert. However, you may use **Advanced Hunting** to hunt for it. Try it now, using the following query:

```
DeviceProcessEvents
| where DeviceName =~ "<your device name>"
| where ActionType == "ProcessCreated"
| where FileName == "cmd.exe"
| where InitiatingProcessAccountName != "system"
| project Timestamp, InitiatingProcessId, InitiatingProcessFileName, ProcessId,
ProcessCommandLine
```

Ingress tool transfer MITRE T1105

The above test will use PowerShell to download a file from the public internet to “LICENSE.txt” and display it with “notepad.exe”.

> Investigation tip: look for a Microsoft Defender for Endpoint alert named “**Suspicious Powershell command line**”, as well as a Microsoft Defender Antivirus alert named “**Suspicious ‘AmsiProcessDetect’ behavior was blocked**”.

Rundll32 MITRE T1218.011

In this test, “notepad.exe” will spawn as a child process of “rundll32.exe”.

> Investigation tip: process creation events caused by rundll32.exe can be queried in **Advanced Hunting**, using the following query:

```
DeviceProcessEvents
| where DeviceName =~ "<your device name>"
| where ActionType == "ProcessCreated"
| where InitiatingProcessFileName == "rundll32.exe"
| project Timestamp, InitiatingProcessId, InitiatingProcessFileName,
InitiatingProcessCommandLine
```

Windows service MITRE T1543.003

The test consists of commands that will install a service named “CMDTestService”. That service executes “cmd.exe”, which, in turn, writes the current date to “%windir%\Temp\current_date.txt”. The service is then deleted.

> Investigation tip: search for alerts named “**Suspicious service registration**” and “**Suspicious system service discovery”**. Alternatively, you may find this activity in the **Device Timeline**, by searching for “CMDTestService”.

About Atomic Red Team

Created by Red Canary, Atomic Red Team™ is an open-source library of tests that security teams can use to simulate adversarial activity in their environments.

Atomic tests are simple. Each test is mapped to exactly one MITRE ATT&CK® technique or sub-technique, so it's comparatively easy to develop a focused testing strategy. Most of the time, you don't have to install any software to run an atomic test, and many tests come with easy-to-use configuration and cleanup commands.

We originally created Atomic Red Team to test our own detection coverage. We found the usual testing method—executing real malware samples—to be cumbersome, potentially risky, and not representative of real-world adversary behaviors. We began work on a suite of tests mapped to MITRE ATT&CK. Atomic Red Team was released to the public in 2017.

To learn more, dive into the GitHub repo, and join our open Slack channel to connect with the maintainers and other contributors.