Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

redcanaryco / atomic-red-team Public

Notifications Fork 2.8k Star 9.7k

Code Issues 6 Pull requests 5 Actions Wiki Security Insights

Files

f339e7d

Go to file

- .github
- atomic_red_team
- atomics
  - Indexes
  - T1003.001
  - T1003.002
  - T1003.003
  - T1003.004
  - T1003.005
  - T1003.006
  - T1003.007
  - T1003.008
  - T1003
  - T1006
  - T1007
  - T1010
  - T1012
  - T1014
  - T1016
  - T1018
  - T1020
  - T1021.001
  - T1021.002
  - T1021.003
  - T1021.006
  - T1027.001
  - T1027.002
  - T1027.004
  - T1027
  - T1030
  - T1033
  - T1036.003
  - T1036.004
  - T1036.005
  - T1036.006
  - T1036

atomic-red-team / atomics / T1497.001 / T1497.001.md

CircleCI Atomic Red Team doc... Generate docs from job=genera... 7091fa8 · 2 years ago History

Preview Code Blame 144 lines (61 loc) · 5.94 KB Raw

T1497.001 - System Checks

Description from ATT&CK

> Adversaries may employ various system checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) during automated discovery to shape follow-on behaviors.(Citation: Deloitte Environment Awareness)
> Specific checks will vary based on the target and/or adversary, but may involve behaviors such as Windows Management Instrumentation, PowerShell, System Information Discovery, and Query Registry to obtain system information and search for VME artifacts. Adversaries may search for VME artifacts in memory, processes, file system, hardware, and/or the Registry. Adversaries may use scripting to automate these checks into one script and then have the program exit if it determines the system to be a virtual environment.
>
> Checks could include generic system properties such as host/domain name and samples of network traffic. Adversaries may also check the network adapters addresses, CPU core count, and available memory/drive size.
>
> Other common checks may enumerate services running that are unique to these applications, installed programs on the system, manufacturer/product fields for strings relating to virtual machine applications, and VME-specific hardware/processor instructions.(Citation: McAfee Virtual Jan 2017) In applications like VMWare, adversaries can also use a special I/O port to send commands and receive output.
>
> Hardware checks, such as the presence of the fan, temperature, and audio devices, could also be used to gather evidence that can be indicative a virtual environment. Adversaries may also query for specific readings from these devices.(Citation: Unit 42 OilRig Sept 2018)

Atomic Tests

- Atomic Test #1 - Detect Virtualization Environment (Linux)
- Atomic Test #2 - Detect Virtualization Environment (Windows)
- Atomic Test #3 - Detect Virtualization Environment (MacOS)
- Atomic Test #4 - Detect Virtualization Environment via WMI Manufacturer/Model Listing (Windows)

- T1037.001
- T1037.002
- T1037.004
- T1037.005
- T1039
- T1040

Atomic Test #1 - Detect Virtualization Environment (Linux)

systemd-detect-virt detects execution in a virtualized environment. At boot, dmesg stores a log if a hypervisor is detected.

Supported Platforms: Linux

auto_generated_guid: dfbd1a21-540d-4574-9731-e852bd6fe840

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
if (systemd-detect-virt || sudo dmidecode | egrep -i 'manufacturer|produ
```

Atomic Test #2 - Detect Virtualization Environment (Windows)

Windows Management Instrumentation(WMI) objects contains system information which helps to detect virtualization. This command will specifically attempt to get the CurrentTemperature value from this object and will check to see if the attempt results in an error that contains the word supported. This is meant to find the result of Not supported, which is the result if run in a virtual machine

Supported Platforms: Windows

auto_generated_guid: 502a7dc4-9d6f-4d28-abf2-f0e84692562d

Attack Commands: Run with `powershell`!

```
$error.clear()
Get-WmiObject -Query "SELECT * FROM MSAcpi_ThermalZoneTemperature" -Erro
if($error) {echo "Virtualization Environment detected"}
```

Cleanup Commands:

```
$error.clear()
```

Atomic Test #3 - Detect Virtualization Environment (MacOS)

ioreg contains registry entries for all the device drivers in the system. If it's a virtual machine, one of the device manufacturer will be a Virtualization Software.

Supported Platforms: macOS

auto_generated_guid: a960185f-aef6-4547-8350-d1ce16680d09

Attack Commands: Run with `sh`!

```
if (ioreg -l | grep -e Manufacturer -e 'Vendor Name' | grep -iE 'Oracle|'
```

Atomic Test #4 - Detect Virtualization Environment via WMI Manufacturer/Model Listing (Windows)

Windows Management Instrumentation(WMI) objects contain system information which helps to detect virtualization. This test will get the model and manufacturer of the machine to determine if it is a virtual machine, such as through VMware or VirtualBox.

Supported Platforms: Windows

auto_generated_guid: 4a41089a-48e0-47aa-82cb-5b81a463bc78

Attack Commands: Run with `powershell`!

```
$Manufacturer = Get-WmiObject -Class Win32_ComputerSystem | select-objec
$Model = Get-WmiObject -Class Win32_ComputerSystem | select-object -expa
if((($Manufacturer.ToLower() -eq "microsoft corporation") -and ($Model.T
```