MITRE ATT&CK T1082 System Information Discovery

By Huseyin Can YUCEEL

Published: 2022-06-09 · Archived: 2026-04-05 18:38:00 UTC

System Information Discovery involves collecting details about compromised systems or networks, such as hardware specifications, software inventories, and network configurations—often using built-in, native OS tools.

In the Red Report 2025, this technique ranks seventh among the top ten most prevalent adversarial tactics. Its prominence highlights the frequent use of living-off-the-land binaries (LOLBins) and native tools [1], which enable attackers to perform stealthy reconnaissance while mimicking legitimate activities.

Adversary Use of System Information Discovery

Adversaries leverage system information discovery techniques to collect details about a compromised system. For example, they may investigate the operating system version, architecture, and configuration to identify potential vulnerabilities or optimize their attack strategies. This information is not only valuable for exploit development but also for selecting and employing tools specifically designed for the targeted environment.

The methods used for system information discovery can be categorized into two broad approaches:

- System Commands for Information Collection: Adversaries utilize built-in system commands to extract details such as the operating system type, version, hardware specifications, and network configurations.
- API Calls for Information in Cloud and Virtual Environments: In cloud or virtualized environments, adversaries may exploit available APIs to gather information about system configurations, infrastructure settings, and deployed services.

Understanding these techniques helps illuminate the ways adversaries operate across various platforms and highlights the importance of monitoring for such activities to safeguard systems and infrastructure.

OS Commands Used to Collect System Information

As stressed earlier, adversaries often use built-in OS commands to gather system details during reconnaissance. Here are some, but not all, OS-native tools commonly used in malware campaigns:

- On Windows, tools like Systeminfo provide comprehensive information about the OS and hardware.
- In macOS, commands such as Systemsetup and system_profiler offer insights into system configurations, while uname reveals kernel details.
- On Linux, commands like uname, sysinfo and lsb_release are commonly employed to identify the OS and version.

These platform-specific utilities enable adversaries to efficiently collect information while remaining stealthy.

Let us explain the information gathered by these tools and highlight identified malware samples that leverage them.

systeminfo (Windows)

Systeminfo is a built-in command-line tool that is included with Windows operating systems. This tool can display detailed information about a system's hardware and software components, including the operating system version, the installed hotfixes and service packs, and the system architecture.

The table below shows what information a user can get using the systeminfo tool on Windows machines.

Operating System Configuration	OS name/version/manufacturer/configuration/, OS build type, registered owner, registered organization, original install date, system locale, input locale, product ID, time zone, logon server

Security Information	Hotfix(es)
Hardware Properties	RAM, disk space, network cards, processors, total physical memory, available physical memory, virtual memory
Other System Information	system boot time, system manufacturer, system model, system type, BIOS version, windows directory, system directory, boot device

Below, you will find an example output of the systeminfo tool.

Host Name: MYCOMPUTER

OS Name: Microsoft Windows 10 Pro

OS Version: 10.0.19044 N/A Build 19044

OS Manufacturer: Microsoft Corporation

OS Configuration: Standalone Workstation

OS Build Type: Multiprocessor Free

Registered Owner: John Doe

Registered Organization: N/A

Product ID: 00330-80000-00000-AA825

Original Install Date: 6/15/2021, 3:45:10 PM

System Boot Time: 12/23/2024, 8:20:30 AM

System Manufacturer: Dell Inc.

System Model: XPS 15 7590

System Type: x64-based PC

Processor(s): 1 Processor(s) Installed.

[01]: Intel64 Family 6 Model 158 Stepping 13 GenuineIntel ~2600 Mhz

BIOS Version: Dell Inc. 1.10.1, 6/15/2021

Windows Directory: C:\Windows

System Directory: C:\Windows\system32

Boot Device: \Device\HarddiskVolume1

System Locale: en-us;English (United States)

Input Locale: en-us;English (United States)

Time Zone: (UTC-05:00) Eastern Time (US & Canada)

Total Physical Memory: 16,297 MB

Available Physical Memory: 8,547 MB

Virtual Memory: Max Size: 32,594 MB

Virtual Memory: Available: 22,324 MB

Virtual Memory: In Use: 10,270 MB

Page File Location(s): C:\pagefile.sys

Domain: WORKGROUP

Logon Server: \\MYCOMPUTER

Hotfix(es): 10 Hotfix(es) Installed.

[01]: KB5003173

...

[10]: KB5006670

Network Card(s): 1 NIC(s) Installed.

[01]: Intel(R) Wi-Fi 6 AX201 160MHz

Connection Name: Wi-Fi

DHCP Enabled: Yes

DHCP Server: 192.168.1.1

IP address(es)

[01]: 192.168.1.100

[02]: fe80::1d1f:3a55:dc77:b800

Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

Adversaries commonly use the systeminfo command in the wild.

For instance, in November 2024, it was reported that the **Interlock** ransomware attack leveraged a Remote Access Tool (RAT) to execute the "**systeminfo**" command [2]. This command, run via "**cmd.exe /c systeminfo**," was used to collect system details from the victim's machine and transmit the gathered information to the attackers' command-and-control server.

In another example highlighted in October 2024, **SingleCamper** is a key implant used by the **UAT-5647** threat group [3]. It is loaded by ShadyHammock after being read and decoded from the Windows registry. SingleCamper can execute the following preliminary reconnaissance commands sent by the C2 and respond with the results, such as:

nltest /domain_trusts
systeminfo
ipconfig /all
dir C:\"program Files" C:\"Program Files (x86)" C:\Users

Finally, in one case reported by Microsoft in May 2024, Moonstone Sleet has been observed distributing malware, such as the TrojanDropper:Win64/YouieLoad* (a.k.a data.tmp), via malicious applications like the game DeTankWar [4]. Once executed, this malware can collect system information and relay it back to the attackers.

SHA-256*: 9863173e0a45318f776e36b1a8529380362af8f3e73a2b4875e30d31ad7bd3c1

system_profiler (macOS)

The system_profiler is a command-line utility on macOS that provides detailed information about the hardware and software configuration of a mac device. An adversary who has gained access to a mac host could use this tool to gather information about the system, such as the version of the operating system, the model and make of the computer, the type and amount of memory installed, and so on.

Here is an example command demonstrating how adversaries can leverage the system_profiler utility [5].

system_profiler SPHardwareDataType SPSoftwareDataType

By combining these two data types in a single command, an adversary can efficiently collect a comprehensive profile of both the hardware and software aspects of the system, which can be critical for planning further malicious activities like targeted malware attacks, system exploitation, or data exfiltration.

In fact, in 2024, there is documented evidence of adversaries using the system_profiler utility on macOS to gather system information during their attacks. For instance, the **Cuckoo malware**, reported in May 2024, employs the system_profiler command to extract hardware details from infected macOS systems [6]:

```
10001248c __builtin_strcpy(dest: &systemProfilerCMD, src: "system_profiler SPHardwareDataTy\t,")
100012498 XOR_func(&systemProfilerCMD, 0x23)
1000124a4 char* x0_14 = popenCMD(&systemProfilerCMD, 1)
```

Additionally, the **Rust-based macOS backdoor** analyzed in February 2024 executes the following commands to collect comprehensive system information [7], aiding attackers in profiling the compromised machine:

```
system_profiler SPSoftwareDataType SPHardwareDataType
```

These instances demonstrate that adversaries actively leverage system_profiler to perform system information discovery, facilitating further malicious activities such as data exfiltration or system exploitation.

systemsetup (macOS)

The systemsetup command-line utility in macOS is designed for configuring system settings, such as setting the computer name, adjusting time zones, and managing network configurations. Threat actors, however, often exploit legitimate utilities like this to achieve their objectives—a tactic known as "Living off the Land."

Although systemsetup requires root or administrator privileges to execute certain commands, its options and capabilities can vary depending on the macOS version in use. Commonly, this tool is used for system information discovery or configuration changes that could be misused in malicious activities. Examples include:

-gettimezone: It displays the current time zone of the system.

```
user@macos:~$ sudo systemsetup -gettimezone
Time Zone: Europe/Istanbul
```

Adversaries may leverage this option to determine if the system is configured to use the correct time zone. If not, the target system may be more susceptible to certain types of attacks, such as time-based attacks that rely on the system's clock being out of sync with other systems.

For instance, in a hypothetical scenario, if an attacker discovers a system clock discrepancy, they could schedule a cron job to exploit it, potentially aligning the execution of a malicious script with a specific event or trigger. The cron job might look something like this:

```
0 2 * * * /path/to/malicious/script.sh
```

This line in a crontab file would theoretically schedule the script.sh to run at 2:00 AM system time every day. If the system's clock is incorrectly set, this could trigger the script at an **unexpected time**, possibly aligning with a time-based security loophole or during low monitoring periods.

-getcomputername: It displays the current hostname of the system.

```
user@macos:~$ sudo systemsetup -getcomputername
Computer Name: John's MacBook Pro
```

This option can be used to learn the hostname to determine if the system is configured to use a fully qualified domain name (FQDN) or a simple hostname. It can also be used to identify potential vulnerabilities in the system's name resolution configuration, such as misconfigured DNS records or a lack of domain name validation.

-getremotelogin: It displays the current status of remote login, which allows users to access the system remotely over the network

```
user@macos:~$ sudo systemsetup -getremotelogin
Remote Login: On
```

This option is often leveraged to determine if remote login is enabled on the system, and if this is the case, they may want to learn which remote login protocols are supported. Later, adversaries can use this information to gain unauthorized access to the system by exploiting vulnerabilities in the remote login protocols.

networksetup (macOS)

Systemsetup is not the only built-in tool that adversaries can leverage.

The networksetup tool in macOS can be used by adversaries for reconnaissance purposes. By using the listallnetworkservices option, an adversary can list all network services configured on the system. This information can be crucial for understanding the network environment of the target system and identifying potential avenues for network-based attacks or further exploitation.

user@macos:~$ sudo networksetup -listallnetworkservices

An asterisk (*) denotes that a network service is disabled.

Wi-Fi

Thunderbolt Bridge

*Hotspot Shield VPN

In this example, the command lists available network services like Wi-Fi and Thunderbolt Bridge, and indicates that "Hotspot Shield VPN" is disabled. This knowledge can help an attacker understand the network setup and potentially identify less secure or disabled network services that can be exploited.

On the other hand, the networksetup -getinfo command is another powerful tool in macOS that can be used by adversaries to gather detailed network configuration information. When used with a specific network service like Wi-Fi, it can reveal various settings and parameters.

user@macos:~$ sudo networksetup -getinfo Wi-Fi
DHCP Configuration
IP address: 192.168.1.100
Subnet mask: 255.255.255.0
Router: 192.168.1.1
Client ID:
Wi-Fi ID: 00:1e:65:3b:42:fb

In this output, the command provides critical network information such as the IP address, subnet mask, router address, and the Wi-Fi interface's MAC address. This data can be valuable for an adversary in understanding the network layout, identifying potential internal network targets, and planning further network-based attacks or intrusions.

A notable example involves a backdoor reported in February 2024. Written in Rust language, it targets macOS users, exploiting the networksetup utility to gather detailed information about the victim's machine and its network connections [7]. This malware executed specific commands to enumerate network services and hardware ports, enabling comprehensive system reconnaissance:

networksetup -listallnetworkservices
networksetup -listallhardwareports

The command networksetup -listallnetworkservices was used to list all network services configured on the target system, such as Wi-Fi, Ethernet, or VPN connections. This provided the adversary with an overview of the available network interfaces and their configurations. Additionally, the command networksetup -listallhardwareports revealed details about hardware ports, including device names and MAC addresses, offering insights into the physical and logical network infrastructure.

Built-in Linux Functions

On compromised Linux hosts, adversaries can run built-in commands or create tools that leverage these command-line utilities to gain system-related information.

Function Name	What It Gathers
uname	Name and information about the Linux kernel
sysinfo	Memory statistics and swap space usage
statvfs	Statistics for the filesystem, including the current working directory
if_nameindex	Network interface names
lsb_release	Distribution and version of the operating system

For instance, in December 2024, an analysis of Linux malware revealed that adversaries are exploiting built-in Linux functions to gather system information [8]. Specifically, the malware utilizes the "uname" system call to query kernel version information, aiding in tailoring attacks to the compromised system's environment.

SHA-256*: b0add768c79a7e9f396792dc4b1878fcba9dbe5e9e6e3ee4da05c9ef5ff000fa

This finding underscores the importance of monitoring the use of built-in Linux functions, as they can be exploited by threat actors to facilitate malicious activities on compromised hosts.

API Calls Used to Collect System Information for IaaS

Infrastructure-as-a-Service (IaaS) providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), offer APIs that allow users to retrieve information about the instances in their cloud infrastructure.

Describe-instance-information (AWS)

The DescribeInstanceInformation action is part of the Amazon EC2 Systems Manager API in AWS. It allows you to retrieve information about your Amazon EC2 instances and on-premises servers that are registered with Systems Manager. To call the DescribeInstanceInformation action, adversaries can use the AWS Command Line Interface (CLI) or the Systems Manager API. Here is an example of how adversaries call the action using the AWS CLI:

aws ssm describe-instance-information --instance-information-filter-list key=InstanceIds,valueSet=i-12345678

This command will retrieve information about the instance with the ID i-12345678. You can also specify multiple instances by providing a list of instance IDs in the valueSet parameter.

Here is an example of the JSON response that the DescribeInstanceInformation action might return:


```
{
  "InstanceInformationList": [
    {
      "InstanceId":"i-12345678",
      "PingStatus":"Online",
      "LastPingDateTime":1608299022.927,
      "AgentVersion":"2.3.1234.0",
      "IsLatestVersion":true,
      "PlatformName":"Windows",
      "PlatformType":"Windows",
      "PlatformVersion":"2012",
      "ActivationId":"1234abcd-12ab-12ab-12ab-123456abcdef",
      "IamRole":"ssm-role",
      "RegistrationDate":1608298822.927,
      "ResourceType":"Instance",
```

```
        "Name":"my-instance",
        "IPAddress":"1.2.3.4"
      }
  ]
}
```


Virtual Machine - Get (Azure)

Adversaries can use the Get request to retrieve information about a VM in Microsoft Azure. The Get request can be made using the Azure REST API, Azure PowerShell cmdlets, or Azure CLI. Using the Get request, attackers can retrieve a wide range of information about the VM, including its resource group, location, size, status, and more.

Adversaries can send an HTTP GET request to the Azure Management REST API. The request should be made to the following URL:

https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMach
api-version={apiVersion}

Where:

- subscriptionId is the ID of the subscription that the VM belongs to.
- resourceGroupName is the name of the resource group that the VM belongs to.
- vmName is the name of the VM you want to retrieve information about.
- apiVersion is the version of the Azure Management REST API you want to use.

The request should include an Authorization header with a Bearer token that authenticates the request. Here is a minimized example of the JSON response that the Azure Management REST API might return when you send a GET request to retrieve information about a VM:

{"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}","name
{vmName}","type":"Microsoft.Compute/virtualMachines","location":"EastUS","properties":{"vmId":"
{vmId}","hardwareProfile":{"vmSize":"Standard_D1_v2"},"storageProfile":{"imageReference":
{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest"},"osDisk":{"name":"{vmName}-
osdisk","caching":"ReadWrite","createOption":"FromImage","diskSizeGB":30,"managedDisk":
{"storageAccountType":"Standard_LRS"}}},"osProfile":{"computerName":"
{vmName}","adminUsername":"azureuser","linuxConfiguration":{"disablePasswordAuthentication":true,"ssh":
{"publicKeys":[{"path":"/home/azureuser/.ssh/authorized_keys","keyData":"{ssh-public-key}"}]}}},"networkProfile":
{"networkInterfaces":
[{"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Network/networkInterfaces/{vmName}-
nic","properties":{"primary":true}}]},"provisioningState":"Succeeded"}}

instances.get (GCP)

The instances.get method in Google Cloud Platform (GCP) is used to retrieve information about a specific Compute Engine virtual machine instance. It is a part of the Compute Engine API, which allows you to create and manage virtual machine instances on Google's infrastructure.

To use the instances.get method; you need to provide the name of the instance that you want to retrieve information about, as well as the project and zone in which it is located. You can also specify additional parameters to customize the request.

Here is an example of how to use the instances.get method in the Google Cloud Platform API:

```
gcloud compute instances get [INSTANCE_NAME] \
  --project=[PROJECT_ID] \
  --zone=[ZONE]
```

Here is an example of the minimized JSON response that the instances.get method might return:

{"id":"1234567890","creationTimestamp":"2023-01-01T12:34:56.789Z","name":"my-instance","zone":"projects/my-
project/zones/us-central1-a","machineType":"projects/my-project/machineTypes/n1-standard-
1","status":"RUNNING","disks":[{"deviceName":"my-
instance","index":0,"type":"PERSISTENT","mode":"READ_WRITE","boot":true,"autoDelete":true,"initializeParams":
{"sourceImage":"projects/debian-cloud/global/images/family/debian-9","diskSizeGb":"10","diskType":"projects/my-

project/zones/us-central1-a/diskTypes/pd-standard"},"diskSizeGb":"10","licenses":["projects/my-project/global/licenses/windows-server"],"interface":"SCSI","source":"projects/my-project/zones/us-central1-a/disks/my-instance","guestOsFeatures":[{"type":"VIRTIO_SCSI_MULTIQUEUE"}]}],"canIpForward":false,"networkInterfaces":[{"network":"global/networks/default","subnetwork":"projects/my-project/regions/us-central1/subnetworks/default","accessConfigs":[{"name":"External NAT","type":"ONE_TO_ONE_NAT","natIP":"1.2.3.4"}],"aliasIpRanges":[],"networkIP":"10.128.0.2"}],"description":"My instance","labels":{"env":"prod"},"scheduling":{"preemptible":false,"onHostMaintenance":"MIGRATE","automaticRestart":true},"deletionProtection":false,"reservationAffinity":{"consumeReservationType":"ANY_RESERVATION"}}

Ready to Simulate Real-World Threats From Red Report 2025?

References

[1] “LOLBAS.” Available: https://lolbas-project.github.io. [Accessed: Feb. 17, 2025]

[2] E. Biasiotto, “Unwrapping the emerging Interlock ransomware attack,” Cisco Talos Blog, Nov. 07, 2024. Available: https://blog.talosintelligence.com/emerging-interlock-ransomware/. [Accessed: Nov. 27, 2024]

[3] D. Korzhevin, “UAT-5647 targets Ukrainian and Polish entities with RomCom malware variants,” Cisco Talos Blog, Oct. 17, 2024. Available: https://blog.talosintelligence.com/uat-5647-romcom/. [Accessed: Nov. 27, 2024]

[4] M. T. Intelligence, “Moonstone Sleet emerges as new North Korean threat actor with new bag of tricks,” Microsoft Security Blog, May 28, 2024. Available: https://www.microsoft.com/en-us/security/blog/2024/05/28/moonstone-sleet-emerges-as-new-north-korean-threat-actor-with-new-bag-of-tricks/. [Accessed: Dec. 24, 2024]

[5] “Find your Mac model name and serial number,” Apple Support. Available: https://support.apple.com/en-by/102767. [Accessed: Jan. 03, 2024]

[6] Dhivya, “New Cuckoo Malware Attacking macOS Users to Steal Sensitive Data,” Cyber Security News, May 06, 2024. Available: https://cybersecuritynews.com/malware-attacking-macos/. [Accessed: Dec. 24, 2024]

[7] A. Lapusneanu, “New macOS Backdoor Written in Rust Shows Possible Link with Windows Ransomware Group,” Bitdefender Labs. Available: https://www.bitdefender.com/en-us/blog/labs/new-macos-backdoor-written-in-rust-shows-possible-link-with-windows-ransomware-group. [Accessed: Dec. 24, 2024]

[8] “VirusTotal.” Available: https://www.virustotal.com/gui/file/b0add768c79a7e9f396792dc4b1878fcba9dbe5e9e6e3ee4da05c9ef5ff000fa. [Accessed: Jan. 14, 2025]
