

T1033 - System Owner/User Discovery

Description from ATT&CK

Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping](https://attack.mitre.org/techniques/T1003). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery] (https://attack.mitre.org/techniques/T1033) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Various utilities and commands may acquire this information, including whoami. In macOS and Linux, the currently logged in user can be identified with w and who. On macOS the dscl . list /Users | grep -v '_' command can also be used to enumerate user accounts. Environment variables, such as %USERNAME% and \$USER, may also be used to access this information.

Atomic Tests

- Atomic Test #1 System Owner/User Discovery
- Atomic Test #2 System Owner/User Discovery
- Atomic Test #3 Find computers where user has session Stealth mode (PowerView)
- Atomic Test #4 User Discovery With Env Vars PowerShell Script
- Atomic Test #5 GetCurrent User with PowerShell Script

Atomic Test #1 - System Owner/User Discovery

Identify System owner or users on an endpoint.

Upon successful execution, cmd.exe will spawn multiple commands against a target host to identify usernames. Output will be via stdout. Additionally, two files will be written to disk - computers.txt and usernames.txt.

Supported Platforms: Windows

auto_generated_guid: 4c4959bf-addf-4b4a-be86-8d09cc1857aa

Inputs:

Name	Description	Туре	Default Value
computer_name	Name of remote computer	String	localhost

Attack Commands: Run with command_prompt!

```
cmd.exe /C whoami
wmic useraccount get /ALL
quser /SERVER:"#{computer_name}"
quser
qwinsta.exe /server:#{computer_name}
qwinsta.exe
for /F "tokens=1,2" %i in ('qwinsta /server:#{computer_name} ^| findstr
@FOR /F %n in (computers.txt) DO @FOR /F "tokens=1,2" %i in ('qwinsta /s
```

Atomic Test #2 - System Owner/User Discovery

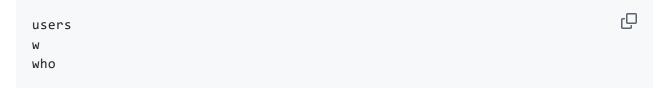
Identify System owner or users on an endpoint

Upon successful execution, sh will stdout list of usernames.

Supported Platforms: Linux, macOS

auto_generated_guid: 2a9b677d-a230-44f4-ad86-782df1ef108c

Attack Commands: Run with sh!



Atomic Test #3 - Find computers where user has session - Stealth mode (PowerView)

Find existing user session on other computers. Upon execution, information about any sessions discovered will be displayed.



atomic-red-team / atomics / T1033 / T1033.md

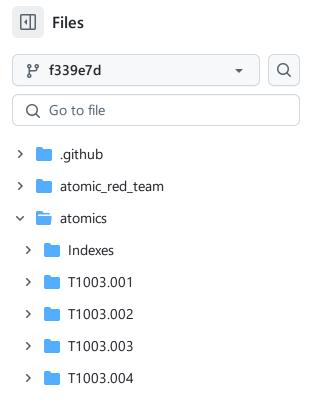


↑ Top

Atomic Test #4 - User Discovery With Env Vars PowerShell Script

Use the PowerShell environment variables to identify the current logged user.

Supported Platforms: Windows



T1003.005



T1037.001

T1037.002

T1037.004

T1037.005

```
auto_generated_guid: dcb6cdee-1fb0-4087-8bf8-88cfd136ba51
Attack Commands: Run with powershell!
  [System.Environment]::UserName | Out-File -FilePath .\CurrentactiveUser. □
  $env:UserName | Out-File -FilePath .\CurrentactiveUser.txt -Append
Cleanup Commands:
                                                                          Q
  Remove-Item -Path .\CurrentactiveUser.txt -Force
Atomic Test #5 - GetCurrent User with PowerShell Script
Use the PowerShell "GetCurrent" method of the WindowsIdentity .NET class to identify the
logged user.
Supported Platforms: Windows
auto_generated_guid: 1392bd0f-5d5a-429e-81d9-eb9d4d4d5b3b
Attack Commands: Run with powershell!
  [System.Security.Principal.WindowsIdentity]::GetCurrent() | Out-File -Fi ☐
Cleanup Commands:
                                                                         Q
  Remove-Item -Path .\CurrentUserObject.txt -Force
```