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

40b77d6

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 / T1083 / T1083.md

Atomic Red Team doc generat... Generated docs from job=generate-d... e48781e · 2 years ago History

Preview Code Blame 288 lines (159 loc) · 7.99 KB Raw

T1083 - File and Directory Discovery

Description from ATT&CK

> Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery] (https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
> Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the Native API. Adversaries may also leverage a Network Device CLI on network devices to gather file and directory information.(Citation: US-CERT-TA18-106A)

Atomic Tests

- Atomic Test #1 - File and Directory Discovery (cmd.exe)
- Atomic Test #2 - File and Directory Discovery (PowerShell)
- Atomic Test #3 - Nix File and Directory Discovery
- Atomic Test #4 - Nix File and Directory Discovery 2
- Atomic Test #5 - Simulating MAZE Directory Enumeration
- Atomic Test #6 - Launch DirLister Executable

Atomic Test #1 - File and Directory Discovery (cmd.exe)

Find or discover files on the file system. Upon successful execution, this test will output the results of all the data discovery commands to a specified file.

Supported Platforms: Windows

auto_generated_guid: 0e36303b-6762-4500-b003-127743b80ba6

Inputs:

Name	Description	Type	Default Value
output_file	File to output results to	String	%temp%\T1083Test1.txt

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

Attack Commands: Run with `command_prompt`!

```
dir /s c:\ >> #{output_file}
dir /s "c:\Documents and Settings" >> #{output_file}
dir /s "c:\Program Files\" >> #{output_file}
dir "%systemdrive%\Users\*.*" >> #{output_file}
dir "%userprofile%\AppData\Roaming\Microsoft\Windows\Recent\*.*" >> #{ou
dir "%userprofile%\Desktop\*.*" >> #{output_file}
tree /F >> #{output_file}
```

Cleanup Commands:

```
del #{output_file}
```

Atomic Test #2 - File and Directory Discovery (PowerShell)

Find or discover files on the file system. Upon execution, file and folder information will be displayed.

Supported Platforms: Windows

auto_generated_guid: 2158908e-b7ef-4c21-8a83-3ce4dd05a924

Attack Commands: Run with `powershell`!

```
ls -recurse
get-childitem -recurse
gci -recurse
```

Atomic Test #3 - Nix File and Directory Discovery

Find or discover files on the file system

References:

http://osxdaily.com/2013/01/29/list-all-files-subdirectory-contents-recursively/

https://perishablepress.com/list-files-folders-recursively-terminal/

Supported Platforms: macOS, Linux

auto_generated_guid: ffc8b249-372a-4b74-adcd-e4c0430842de

Inputs:

Name	Description	Type	Default Value
output_file	Output file used to store the results.	Path	/tmp/T1083.txt

Attack Commands: Run with `sh`!

```
ls -a >> #{output_file}
if [ -d /Library/Preferences/ ]; then ls -la /Library/Preferences/ > #{o
file */* *>> #{output_file}
cat #{output_file} 2>/dev/null
find . -type f
ls -R | grep ":$" | sed -e 's/:$//' -e 's/[^-][^\/]*\//--/g' -e 's/^/ /'
```

```
locate *
which sh
```

Cleanup Commands:

```
rm #{output_file}
```

Atomic Test #4 - Nix File and Directory Discovery 2

Find or discover files on the file system

Supported Platforms: macOS, Linux

auto_generated_guid: 13c5e1ae-605b-46c4-a79f-db28c77ff24e

Inputs:

Name	Description	Type	Default Value
output_file	Output file used to store the results.	Path	/tmp/T1083.txt

Attack Commands: Run with `sh`!

```
cd $HOME && find . -print | sed -e 's;[^/]*/;|__;g;s;__|; |;g' > #{outpu
if [ -f /etc/mtab ]; then cat /etc/mtab >> #{output_file}; fi;
find . -type f -iname *.pdf >> #{output_file}
cat #{output_file}
find . -type f -name ".*"
```

Cleanup Commands:

```
rm #{output_file}
```

Atomic Test #5 - Simulating MAZE Directory Enumeration

This test emulates MAZE ransomware's ability to enumerate directories using Powershell. Upon successful execution, this test will output the directory enumeration results to a specified file, as well as display them in the active window. See https://www.mandiant.com/resources/tactics-techniques-procedures-associated-with-maze-ransomware-incidents

Supported Platforms: Windows

auto_generated_guid: c6c34f61-1c3e-40fb-8a58-d017d88286d8

Inputs:

Name	Description	Type	Default Value
File_to_output	File to output results to	String	$env:temp\T1083Test5.txt

Attack Commands: Run with `powershell`!

```
$folderarray = @("Desktop", "Downloads", "Documents", "AppData/Local", "
Get-ChildItem -Path $env:homedrive -ErrorAction SilentlyContinue | Out-F
Get-ChildItem -Path $env:programfiles -erroraction silentlycontinue | Ou
Get-ChildItem -Path "${env:ProgramFiles(x86)}" -erroraction silentlycont
```

```
$UsersFolder = "$env:homedrive\Users\"
foreach ($directory in Get-ChildItem -Path $UsersFolder -ErrorAction Sil
{
foreach ($secondarydirectory in $folderarray)
 {Get-ChildItem -Path "$UsersFolder/$directory/$secondarydirectory" -Err
}
cat #{File_to_output}
```

Cleanup Commands:

```
remove-item #{File_to_output} -ErrorAction SilentlyContinue
```

Atomic Test #6 - Launch DirLister Executable

Launches the DirLister executable for a short period of time and then exits.

Recently seen used by BlackCat ransomware to create a list of accessible directories and files.

Supported Platforms: Windows

auto_generated_guid: c5bec457-43c9-4a18-9a24-fe151d8971b7

Inputs:

Name	Description	Type	Default Value
dirlister_path	Path to the DirLister executable	String	PathToAtomicsFolder\T1083\bin\DirLister.exe

Attack Commands: Run with `powershell`!

```
Start-Process #{dirlister_path}
Start-Sleep -Second 4
Stop-Process -Name "DirLister"
```

Dependencies: Run with `powershell`!

Description: DirLister.exe must exist in the specified path #{dirlister_path}

Check Prereq Commands:

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

Get Prereq Commands:

```
$parentpath = Split-Path "#{dirlister_path}"
Invoke-WebRequest https://github.com/SanderSade/DirLister/releases/downl
New-Item -ItemType Directory -Force -Path $parentpath | Out-Null
Expand-Archive -Path $env:TEMP\TDirLister.v2.beta4.zip -DestinationPath
Copy-Item $env:TEMP\TDirLister.v2.beta4\* $parentpath -Recurse
Remove-Item $env:TEMP\TDirLister.v2.beta4.zip,$env:TEMP\TDirLister.v2.be
```