≡   ○   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 / T1083 / **T1083.md** 📋

Atomic Red Team doc generat...   Generated docs from job=generate-docs branch=master ...   436a980 · 2 years ago   🕐

232 lines (126 loc) · 6.29 KB

Preview   Code   Blame   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 #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

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\*.*" >> #{output_file}
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/ > #{output_file}
file */* *>> #{output_file}
cat #{output_file} 2>/dev/null
find . -type f
ls -R | grep ":$" | sed -e 's/:$//' -e 's/[^-][^\/]*\//--/g' -e 's/^/ /' -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' > #{output_file}
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", "AppData/Ro;
Get-ChildItem -Path $env:homedrive -ErrorAction SilentlyContinue | Out-File -append
Get-ChildItem -Path $env:programfiles -erroraction silentlycontinue | Out-File -ap
Get-ChildItem -Path "${env:ProgramFiles(x86)}" -erroraction silentlycontinue | Out-
$UsersFolder = "$env:homedrive\Users\"
foreach ($directory in Get-ChildItem -Path $UsersFolder -ErrorAction SilentlyConti
{
foreach ($secondarydirectory in $folderarray)
 {Get-ChildItem -Path "$UsersFolder/$directory/$secondarydirectory" -ErrorAction S
}
cat #{File_to_output}
```

Cleanup Commands:

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