T1222.001 - Windows File and Directory Permissions Modification

Description from ATT&CK

> Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.(Citation: Hybrid Analysis Icacls1 June 2018)(Citation: Hybrid Analysis Icacls2 May 2018) File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.).
> Windows implements file and directory ACLs as Discretionary Access Control Lists (DACLs).(Citation: Microsoft DACL May 2018) Similar to a standard ACL, DACLs identifies the accounts that are allowed or denied access to a securable object. When an attempt is made to access a securable object, the system checks the access control entries in the DACL in order. If a matching entry is found, access to the object is granted. Otherwise, access is denied.(Citation: Microsoft Access Control Lists May 2018)
>
> Adversaries can interact with the DACLs using built-in Windows commands, such as `icacls`, `cacls`, `takeown`, and `attrib`, which can grant adversaries higher permissions on specific files and folders. Further, PowerShell provides cmdlets that can be used to retrieve or modify file and directory DACLs. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via Accessibility Features, Boot or Logon Initialization Scripts, or tainting/hijacking other instrumental binary/configuration files via Hijack Execution Flow.

- Atomic Test #2 - cacls - Grant permission to specified user or group recursively
- Atomic Test #3 - attrib - Remove read-only attribute
- Atomic Test #4 - attrib - hide file
- Atomic Test #5 - Grant Full Access to folder for Everyone - Ryuk Ransomware Style

Atomic Test #1 - Take ownership using takeown utility

Modifies the filesystem permissions of the specified file or folder to take ownership of the object. Upon execution, "SUCCESS" will be displayed for the folder and each file inside of it.

Supported Platforms: Windows

auto_generated_guid: 98d34bb4-6e75-42ad-9c41-1dae7dc6a001

Inputs:

Name	Description	Type	Default Value
file_folder_to_own	Path of the file or folder for takeown to take ownership.	Path	%temp%\T1222.001_takeown_folder

Attack Commands: Run with `command_prompt`!

```
takeown.exe /f #{file_folder_to_own} /r
```

Dependencies: Run with `command_prompt`!

Description: Test requrires a file to take ownership of to be located at (#{file_folder_to_own})

Check Prereq Commands:

```
IF EXIST #{file_folder_to_own} ( EXIT 0 ) ELSE ( EXIT 1 )
```

Get Prereq Commands:

```
mkdir #{file_folder_to_own}
echo T1222.001_takeown1 >> #{file_folder_to_own}\T1222.001_takeown1.txt
echo T1222.001_takeown2 >> #{file_folder_to_own}\T1222.001_takeown2.txt
```

Atomic Test #2 - cacls - Grant permission to specified user or group recursively

Modifies the filesystem permissions of the specified folder and contents to allow the specified user or group Full Control. If "Access is denied" is displayed it may be because the file or folder doesn't exit. Run the prereq command to create it. Upon successfull execution, "Successfully processed 3 files" will be displayed.

Supported Platforms: Windows

auto_generated_guid: a8206bcc-f282-40a9-a389-05d9c0263485

Inputs:

Name	Description	Type	Default Value
file_or_folder	Path of the file or folder to change permissions.	Path	%temp%\T1222.001_cacls
user_or_group	User or group to allow full control	String	Everyone

Attack Commands: Run with `command_prompt`!

```
icacls.exe #{file_or_folder} /grant #{user_or_group}:F
```

Dependencies: Run with `command_prompt`!

Description: Test requrires a file to modify to be located at (#{file_or_folder})

Check Prereq Commands:

```
IF EXIST #{file_or_folder} ( EXIT 0 ) ELSE ( EXIT 1 )
```

Get Prereq Commands:

```
mkdir #{file_or_folder}
echo T1222.001_cacls1 >> #{file_or_folder}\T1222.001_cacls1.txt
echo T1222.001_cacls2 >> #{file_or_folder}\T1222.001_cacls2.txt
```

Atomic Test #3 - attrib - Remove read-only attribute

Removes the read-only attribute from a file or folder using the attrib.exe command. Upon execution, no output will be displayed. Open the file in File Explorer > Right Click - Prperties and observe that the Read Only checkbox is empty.

Supported Platforms: Windows

auto_generated_guid: bec1e95c-83aa-492e-ab77-60c71bbd21b0

Inputs:

Name	Description	Type	Default Value
file_or_folder	Path of the file or folder remove attribute.	Path	%temp%\T1222.001_attrib

Attack Commands: Run with `command_prompt`!

```
attrib.exe -r #{file_or_folder}\*.* /s
```

Dependencies: Run with `command_prompt`!

Description: Test requrires a file to modify to be located at (#{file_or_folder})

Check Prereq Commands:

```
IF EXIST #{file_or_folder} ( EXIT 0 ) ELSE ( EXIT 1 )
```

Get Prereq Commands:

```
mkdir #{file_or_folder}
echo T1222.001_attrib1 >> #{file_or_folder}\T1222.001_attrib1.txt
echo T1222.001_attrib2 >> #{file_or_folder}\T1222.001_attrib2.txt
attrib.exe +r #{file_or_folder}\T1222.001_attrib1.txt
attrib.exe +r #{file_or_folder}\T1222.001_attrib2.txt
```

Atomic Test #4 - attrib - hide file

Attackers leverage an existing Windows binary, attrib.exe, to mark specific files or folder as hidden by using specific flags so that the victim does not see the file.

Supported Platforms: Windows

auto_generated_guid: 32b979da-7b68-42c9-9a99-0e39900fc36c

Inputs:

Name	Description	Type	Default Value
file_or_folder	Path of the files to hide.	Path	%temp%\T1222.001_attrib_2

Attack Commands: Run with `command_prompt`!

```
mkdir #{file_or_folder} >nul 2>&1
echo T1222.001_attrib1 >> #{file_or_folder}\T1222.001_attrib1.txt
echo T1222.001_attrib2 >> #{file_or_folder}\T1222.001_attrib2.txt
attrib.exe +h #{file_or_folder}\T1222.001_attrib1.txt
attrib.exe +h #{file_or_folder}\T1222.001_attrib2.txt
```

Cleanup Commands:

```
del /A:H #{file_or_folder}\T1222.001_attrib*.txt >nul 2>&1
```

Atomic Test #5 - Grant Full Access to folder for Everyone - Ryuk Ransomware Style

Invokes the command line similar to that used by Ryuk Ransomware to grant full access to the entire C:\ drive for Everyone. **icacls "C:*" /grant Everyone:F /T /C /Q** However, for this atomic we set the permission on C:\Users\Public so it completes faster and doesn't irreversibly affect the host. You can set your own path variable to "C:*" if you prefer.

Supported Platforms: Windows

auto_generated_guid: ac7e6118-473d-41ec-9ac0-ef4f1d1ed2f6

Inputs:

Name	Description	Type	Default Value
path	Path of folder to recursively set permissions on	Path	C:\Users\Public*
file_path	Path of folder permission back	Path	%temp%\T1222.001-folder-perms-backup.txt

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

```
icacls "#{path}" /grant Everyone:F /T /C /Q
```

Cleanup Commands:

```
icacls '#{path}' /restore #{file_path} /q >nul 2>&1
```

Dependencies: Run with `command_prompt`!

Description: Backup of original folder permissions should exist (for use in cleanup commands)

Check Prereq Commands:

```
IF EXIST #{file_path} ( EXIT 0 ) ELSE ( EXIT 1 )
```

Get Prereq Commands:

```
icacls #{path} /save #{file_path} /t /q >nul 2>&1
```