

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

> T1036.003

> T1036.004

> T1036.005

> T1036.006

> T1036

atomic-red-team / atomics / T1222.002 / T1222.002.md

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

Preview

Code

Blame

337 lines (152 loc) · 8.68 KB

Raw

# T1222.002 - Linux and Mac 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.).

Most Linux and Linux-based platforms provide a standard set of permission groups (user, group, and other) and a standard set of permissions (read, write, and execute) that are applied to each group. While nuances of each platform’s permissions implementation may vary, most of the platforms provide two primary commands used to manipulate file and directory ACLs: `chown` (short for change owner), and `chmod` (short for change mode).

Adversarial may use these commands to make themselves the owner of files and directories or change the mode if current permissions allow it. They could subsequently lock others out of the file. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Unix Shell Configuration Modification](#) or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](#).(Citation: 20 macOS Common Tools and Techniques)

## Atomic Tests

- [Atomic Test #1 - chmod - Change file or folder mode \(numeric mode\)](#)
- [Atomic Test #2 - chmod - Change file or folder mode \(symbolic mode\)](#)
- [Atomic Test #3 - chmod - Change file or folder mode \(numeric mode\) recursively](#)
- [Atomic Test #4 - chmod - Change file or folder mode \(symbolic mode\) recursively](#)
- [Atomic Test #5 - chown - Change file or folder ownership and group](#)
- [Atomic Test #6 - chown - Change file or folder ownership and group recursively](#)
- [Atomic Test #7 - chown - Change file or folder mode ownership only](#)
- [Atomic Test #8 - chown - Change file or folder ownership recursively](#)
- [Atomic Test #9 - chattr - Remove immutable file attribute](#)

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

## Atomic Test #1 - chmod - Change file or folder mode (numeric mode)

Changes a file or folder's permissions using chmod and a specified numeric mode.

**Supported Platforms:** macOS, Linux

**auto\_generated\_guid:** 34ca1464-de9d-40c6-8c77-690adf36a135

**Inputs:**

Name	Description	Type	Default Value
numeric_mode	Specified numeric mode value	Integer	755
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002

**Attack Commands:** Run with **bash** !

```
chmod #{numeric_mode} #{file_or_folder}
```

## Atomic Test #2 - chmod - Change file or folder mode (symbolic mode)

Changes a file or folder's permissions using chmod and a specified symbolic mode.

**Supported Platforms:** macOS, Linux

**auto\_generated\_guid:** fc9d6695-d022-4a80-91b1-381f5c35aff3

**Inputs:**

Name	Description	Type	Default Value
symbolic_mode	Specified symbolic mode value	String	a+w
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002

**Attack Commands:** Run with **bash** !

```
chmod #{symbolic_mode} #{file_or_folder}
```

## Atomic Test #3 - chmod - Change file or folder mode (numeric mode) recursively

Changes a file or folder's permissions recursively using chmod and a specified numeric mode.

Supported Platforms: macOS, Linux

auto\_generated\_guid: ea79f937-4a4d-4348-ace6-9916aec453a4

Inputs:

Name	Description	Type	Default Value
numeric_mode	Specified numeric mode value	Integer	755
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002

Attack Commands: Run with **bash** !

```
chmod -R #{numeric_mode} #{file_or_folder}
```

## Atomic Test #4 - chmod - Change file or folder mode (symbolic mode) recursively

Changes a file or folder's permissions recursively using chmod and a specified symbolic mode.

Supported Platforms: macOS, Linux

auto\_generated\_guid: 0451125c-b5f6-488f-993b-5a32b09f7d8f

Inputs:

Name	Description	Type	Default Value
symbolic_mode	Specified symbolic mode value	String	a+w
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002

Attack Commands: Run with **bash** !

```
chmod -R #{symbolic_mode} #{file_or_folder}
```

## Atomic Test #5 - chown - Change file or folder ownership and group

Changes a file or folder's ownership and group information using chown.

Supported Platforms: macOS, Linux

auto\_generated\_guid: d169e71b-85f9-44ec-8343-27093ff3dfc0

Inputs:

Name	Description	Type	Default Value
------	-------------	------	---------------

owner	Username of desired owner	String	root
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002/T1222.002
group	Group name of desired group	String	root

Attack Commands: Run with `bash` !

```
chown #{owner}:#{group} #{file_or_folder}
```

## Atomic Test #6 - chown - Change file or folder ownership and group recursively

Changes a file or folder's ownership and group information recursively using chown.

Supported Platforms: macOS, Linux

auto\_generated\_guid: b78598be-ff39-448f-a463-adbf2a5b7848

Inputs:

Name	Description	Type	Default Value
owner	Username of desired owner	String	root
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002
group	Group name of desired group	String	root

Attack Commands: Run with `bash` !

```
chown -R #{owner}:#{group} #{file_or_folder}
```

## Atomic Test #7 - chown - Change file or folder mode ownership only

Changes a file or folder's ownership only using chown.

Supported Platforms: macOS, Linux

auto\_generated\_guid: 967ba79d-f184-4e0e-8d09-6362b3162e99

Inputs:

Name	Description	Type	Default Value
------	-------------	------	---------------

owner	Username of desired owner	String	root
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002/T1222.002

Attack Commands: Run with `bash` !

```
chown #{owner} #{file_or_folder}
```

## Atomic Test #8 - chown - Change file or folder ownership recursively

Changes a file or folder's ownership only recursively using chown.

Supported Platforms: macOS, Linux

auto\_generated\_guid: 3b015515-b3d8-44e9-b8cd-6fa84faf30b2

Inputs:

Name	Description	Type	Default Value
owner	Username of desired owner	String	root
file_or_folder	Path of the file or folder	Path	/tmp/AtomicRedTeam/atomics/T1222.002

Attack Commands: Run with `bash` !

```
chown -R #{owner} #{file_or_folder}
```

## Atomic Test #9 - chattr - Remove immutable file attribute

Remove's a file's `immutable` attribute using `chattr` . This technique was used by the threat actor Rocke during the compromise of Linux web servers.

Supported Platforms: macOS, Linux

auto\_generated\_guid: e7469fe2-ad41-4382-8965-99b94dd3c13f

Inputs:

Name	Description	Type	Default Value
file_to_modify	Path of the file	Path	/var/spool/cron/root

Attack Commands: Run with `sh` !

```
chattr -i #{file_to_modify}
```

