



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 / T1070.006 / T1070.006.md



CircleCI Atomic Red Team doc... Generate docs from job=generate_and_commit_gui... 36d49de · 3 years ago

387 lines (211 loc) · 10.1 KB

Preview

Code

Blame

Raw



T1070.006 - Timestomp

Description from ATT&CK

Adversaries may modify file time attributes to hide new or changes to existing files. Timestomping is a technique that modifies the timestamps of a file (the modify, access, create, and change times), often to mimic files that are in the same folder. This is done, for example, on files that have been modified or created by the adversary so that they do not appear conspicuous to forensic investigators or file analysis tools.

Timestomping may be used along with file name [Masquerading](#) to hide malware and tools. (Citation: WindowsIR Anti-Forensic Techniques)

Atomic Tests

- [Atomic Test #1 - Set a file's access timestamp](#)
- [Atomic Test #2 - Set a file's modification timestamp](#)
- [Atomic Test #3 - Set a file's creation timestamp](#)

- [Atomic Test #4 - Modify file timestamps using reference file](#)
- [Atomic Test #5 - Windows - Modify file creation timestamp with PowerShell](#)
- [Atomic Test #6 - Windows - Modify file last modified timestamp with PowerShell](#)
- [Atomic Test #7 - Windows - Modify file last access timestamp with PowerShell](#)
- [Atomic Test #8 - Windows - Timestamp a File](#)

Atomic Test #1 - Set a file's access timestamp

Stomps on the access timestamp of a file

Supported Platforms: Linux, macOS

auto_generated_guid: 5f9113d5-ed75-47ed-ba23-ea3573d05810

Inputs:

| Name | Description | Type | Default Value |
|-----------------|---|------|---------------|
| target_filename | Path of file that we are going to stomp on last access time | Path | /opt/filename |

Attack Commands: Run with `sh` !

```
touch -a -t 197001010000.00 #{target_filename}
```

Atomic Test #2 - Set a file's modification timestamp

Stomps on the modification timestamp of a file

Supported Platforms: Linux, macOS

auto_generated_guid: 20ef1523-8758-4898-b5a2-d026cc3d2c52

Inputs:

| Name | Description | Type | Default Value |
|-----------------|---|------|---------------|
| target_filename | Path of file that we are going to stomp on last access time | Path | /opt/filename |

Attack Commands: Run with `sh` !

```
touch -m -t 197001010000.00 #{target_filename}
```

Atomic Test #3 - Set a file's creation timestamp

Stomps on the create timestamp of a file

Setting the creation timestamp requires changing the system clock and reverting. Sudo or root privileges are required to change date. Use with caution.

Supported Platforms: Linux, macOS

auto_generated_guid: 8164a4a6-f99c-4661-ac4f-80f5e4e78d2b

Inputs:

| Name | Description | Type | Default Value |
|-----------------|---|------|---------------|
| target_filename | Path of file that we are going to stomp on last access time | Path | /opt/filename |

Attack Commands: Run with `sh` !

```
NOW=$(date)
date -s "1970-01-01 00:00:00"
touch #{target_filename}
date -s "$NOW"
stat #{target_filename}
```



Atomic Test #4 - Modify file timestamps using reference file

Modifies the `modify` and `access` timestamps using the timestamps of a specified reference file.

This technique was used by the threat actor Rocke during the compromise of Linux web servers.

Supported Platforms: Linux, macOS

auto_generated_guid: 631ea661-d661-44b0-abdb-7a7f3fc08e50

Inputs:

| Name | Description | Type | Default Value |
|---------------------|--|------|---------------|
| target_file_path | Path of file to modify timestamps of | Path | /opt/filename |
| reference_file_path | Path of reference file to read timestamps from | Path | /bin/sh |

Attack Commands: Run with `sh` !

```
touch -acmr #{reference_file_path} #{target_file_path}
```



Atomic Test #5 - Windows - Modify file creation timestamp with PowerShell

Modifies the file creation timestamp of a specified file. This technique was seen in use by the Stitch RAT. To verify execution, use File Explorer to view the Properties of the file and observe that the Created time is the year 1970.

Supported Platforms: Windows

auto_generated_guid: b3b2c408-2ff0-4a33-b89b-1cb46a9e6a9c

Inputs:

| Name | Description | Type | Default Value |
|------------------|---|--------|------------------------------------|
| target_date_time | Date/time to replace original timestamps with | String | 01/01/1970 00:00:00 |
| file_path | Path of file to change creation timestamp | Path | \$env:TEMP\T1551.006_timestomp.txt |

Attack Commands: Run with **powershell** !

```
Get-ChildItem #{file_path} | % { $_.CreationTime = "#{target_date_time}" }
```

Cleanup Commands:

```
Remove-Item #{file_path} -Force -ErrorAction Ignore
```

Dependencies: Run with **powershell** !

Description: A file must exist at the path (#{file_path}) to change the creation time on

Check Prereq Commands:

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

Get Prereq Commands:

```
New-Item -Path #{file_path} -Force | Out-Null
```

```
Set-Content #{file_path} -Value "T1551.006 Timestamp" -Force | Out-Null
```

Atomic Test #6 - Windows - Modify file last modified timestamp with PowerShell

Modifies the file last modified timestamp of a specified file. This technique was seen in use by the Stitch RAT. To verify execution, use File Explorer to view the Properties of the file and observe that the Modified time is the year 1970.

Supported Platforms: Windows

auto_generated_guid: f8f6634d-93e1-4238-8510-f8a90a20dcf2

Inputs:

| Name | Description | Type | Default Value |
|------------------|---|--------|------------------------------------|
| target_date_time | Date/time to replace original timestamps with | String | 01/01/1970 00:00:00 |
| file_path | Path of file to change modified timestamp | Path | \$env:TEMP\T1551.006_timestamp.txt |

Attack Commands: Run with powershell !

```
Get-ChildItem #{file_path} | % { $_.LastWriteTime = "#{target_date_time}" }
```

Cleanup Commands:

```
Remove-Item #{file_path} -Force -ErrorAction Ignore
```

Dependencies: Run with powershell !

Description: A file must exist at the path (#{file_path}) to change the modified time on

Check Prereq Commands:

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

Get Prereq Commands:

```
New-Item -Path #{file_path} -Force | Out-Null
Set-Content #{file_path} -Value "T1551.006 Timestamp" -Force | Out-Null
```

Atomic Test #7 - Windows - Modify file last access timestamp with PowerShell

Modifies the last access timestamp of a specified file. This technique was seen in use by the Stitch RAT. To verify execution, use File Explorer to view the Properties of the file and observe that the Accessed time is the year 1970.

Supported Platforms: Windows

auto_generated_guid: da627f63-b9bd-4431-b6f8-c5b44d061a62

Inputs:

| Name | Description | Type | Default Value |
|------------------|---|--------|------------------------------------|
| target_date_time | Date/time to replace original timestamps with | String | 01/01/1970 00:00:00 |
| file_path | Path of file to change last access timestamp | Path | \$env:TEMP\T1551.006_timestamp.txt |

Attack Commands: Run with powershell !

```
Get-ChildItem #{file_path} | % { $_.LastAccessTime = "#{target_date_time}" }
```

Cleanup Commands:

```
Remove-Item #{file_path} -Force -ErrorAction Ignore
```



Dependencies: Run with `powershell` !

Description: A file must exist at the path (`#{file_path}`) to change the last access time on

Check Prereq Commands:

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



Get Prereq Commands:

```
New-Item -Path #{file_path} -Force | Out-Null
Set-Content #{file_path} -Value "T1551.006 Timestamp" -Force | Out-Null
```



Atomic Test #8 - Windows - Timestamp a File

Timestamp kxwn.lock.

Successful execution will include the placement of kxwn.lock in `#{file_path}` and execution of timestamp.ps1 to modify the time of the .lock file.

[Mitre ATT&CK Evals](#)

Supported Platforms: Windows

auto_generated_guid: d7512c33-3a75-4806-9893-69abc3ccdd43

Inputs:

| Name | Description | Type | Default Value |
|-----------|---------------------------------|--------|-------------------------|
| file_path | File path for timestamp payload | String | \$env:appdata\Microsoft |

Attack Commands: Run with `powershell` !


```
import-module #{file_path}\timestamp.ps1  
timestamp -dest "#{file_path}\kxwn.lock"
```



Cleanup Commands:

```
Remove-Item #{file_path}\timestamp.ps1 -ErrorAction Ignore  
Remove-Item #{file_path}\kxwn.lock -ErrorAction Ignore
```



Dependencies: Run with `powershell!`

Description: timestamp.ps1 must be present in #{file_path}.

Check Prereq Commands:

```
if (Test-Path #{file_path}\timestamp.ps1) {exit 0} else {exit 1}
```



Get Prereq Commands:

```
Invoke-WebRequest "https://raw.githubusercontent.com/mitre-attack/attack-arsenal/b
```



Description: kxwn.lock must be present in #{file_path}.

Check Prereq Commands:

```
if (Test-Path -path "#{file_path}\kxwn.lock") {exit 0} else {exit 1}
```



Get Prereq Commands:

```
New-Item -Path #{file_path}\kxwn.lock -ItemType File
```

