# Windows & Linux Timestamps

Presenter: Fusen(Dobby) Guo

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

# Understanding Windows & Linux Timestamps

Presented by Fusen(Dobby) Guo

SWINBURNE UNIVERSITY OF TECHNOLOGY

# Acknowledgement of Country

On behalf of those present I acknowledge the Wurundjeri people of the Kulin Nation who are the traditional custodians of the land on which we now meet. I pay my respect to their Elders: past, present and emerging.

I also pay my respect to all Aboriginal and Torres Strait Islander people of Australia and hope that the path towards reconciliation continues to be shared and embraced.

# Topics Covered

- Windows Timestamp
- How does the Timestamp change in windows
- Linux Timestamp(inode)
- How does the Timestamp change in Linux
- Timestamp Manipulation
- How Attackers Manipulates Timestamps

# Windows Timestamp

**Definition**: MACB timestamps refer to the four types of time-related attributes that are tracked for each file or directory.

**M - Modified Time:**

1: It records the last time the file's content was modified.
2: It only changes whenever the actual contents of the file are altered.

**Example:** Editing a text document will update this timestamp.

**A - Accessed Time:**

1: It reflects the last time the file or directory was accessed or read.

**Example:** Simply opening a file (without modifying it) can update this timestamp
**Note:** Some Windows systems, this timestamp might be disabled (hided) by default to improve performance, but it can be enabled via registry settings.

**C - Changed Time:**

1: It is updated when the file's metadata (such as name, permissions, or location) is modified.
2: If only the content is changed without altering metadata, this time remains unchanged.

**Example:** Renaming a file or Changing its security settings

**B - Birth Time:**

1: It indicates the creation time of the file or directory.
2: Moving a file within the same file system will not change it

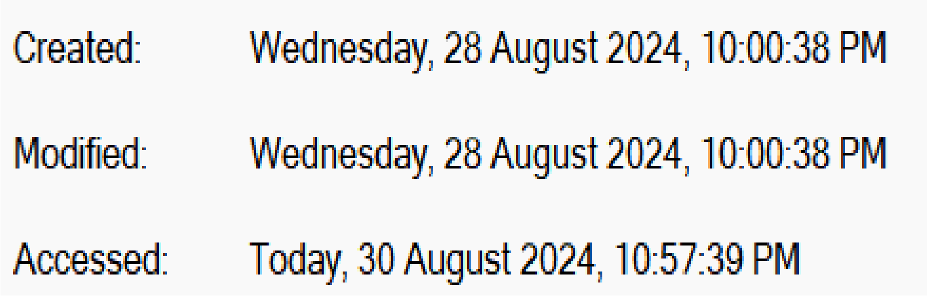
**Example:** Creating a new document or copying a file to a new directory
**Note:** In the Linux system, we only need to emphasis on access time, modified time and changed time. Because birth time is designed to be immutable and does not change after a file is created.

# How does the Timestamp change in windows

## How to see the timestamp at Windows?

**Method 1: Viewing Timestamps by the properties**

Locate the file you want to check → Right click on the file → Select “Properties” → You will see:



Created: Wednesday, 28 August 2024, 10:00:38 PM

Modified: Wednesday, 28 August 2024, 10:00:38 PM

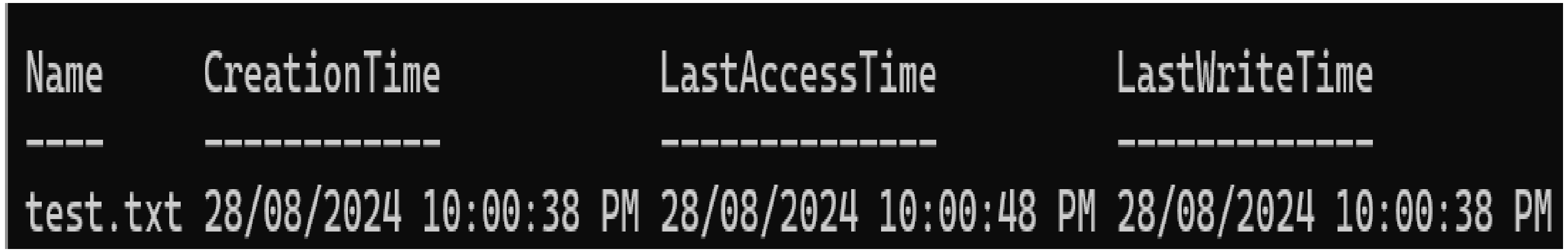
Accessed: Today, 30 August 2024, 10:57:39 PM

**Method 2: Using PowerShell to View Timestamps**

1: Open the Powershell at windows

2: Navigate to the directory containing your files using the cd command, like: cd C:\Users\YourUsername\Documents

3: Use the following command: Get-Item "C:\Path\To\Your\File.txt" | Select-Object Name, CreationTime, LastAccessTime, LastWriteTime



```
Name     CreationTime           LastAccessTime         LastWriteTime
----     ------------           --------------         -------------
test.txt 28/08/2024 10:00:38 PM 28/08/2024 10:00:48 PM 28/08/2024 10:00:38 PM
```

# How does the Timestamp change in windows

## General PowerShell Usage

### When Renaming a File

**Command**: Use the rename command

```
Rename-Item "C:\path\to\your\file.txt" "newfile.txt"
```

**Observation:** Only the Changed Time will update. The Modified Time and Birth Time remain unchanged.

### When Moving a File Between Folders

**Command**: Use the move command

```
Move-Item "C:\path\to\your\file.txt" "C:\newfolder\"
```

**Observation:** Only the Accessed Time will update.

### When Moving a File Between Disks

**Command**: Use move command with paths on different drives

```
Move-Item "C:\path\to\your\file.txt" "D:\newfolder\"
```

**Observation:** Both Accessed Time and Birth Time will update.

### When Copying a File

**Command:** Use copy command

```
Copy-Item "C:\path\to\your\file.txt" "C:\newfolder\"
```

**Observation:** The Accessed Time and Birth Time will update to the current time.

### When Writing a New File

**Command:** Use New-Item-Path command

```
New-Item -Path "C:\Path\To\Your\NewFile.txt" -ItemType "File"
```

**Observation:** All timestamps (Modified, Accessed, Birth) will be set to the current time of file creation.

### When Modifying an Existing File

**Command:** Use Add-Content command

```
Add-Content "C:\path\to\your\file.txt" "Appending some text"
```

**Observation:** The Modified Time and Accessed Time will update.

### When Accessing an Existing File

**Command:** Use Get-Content command

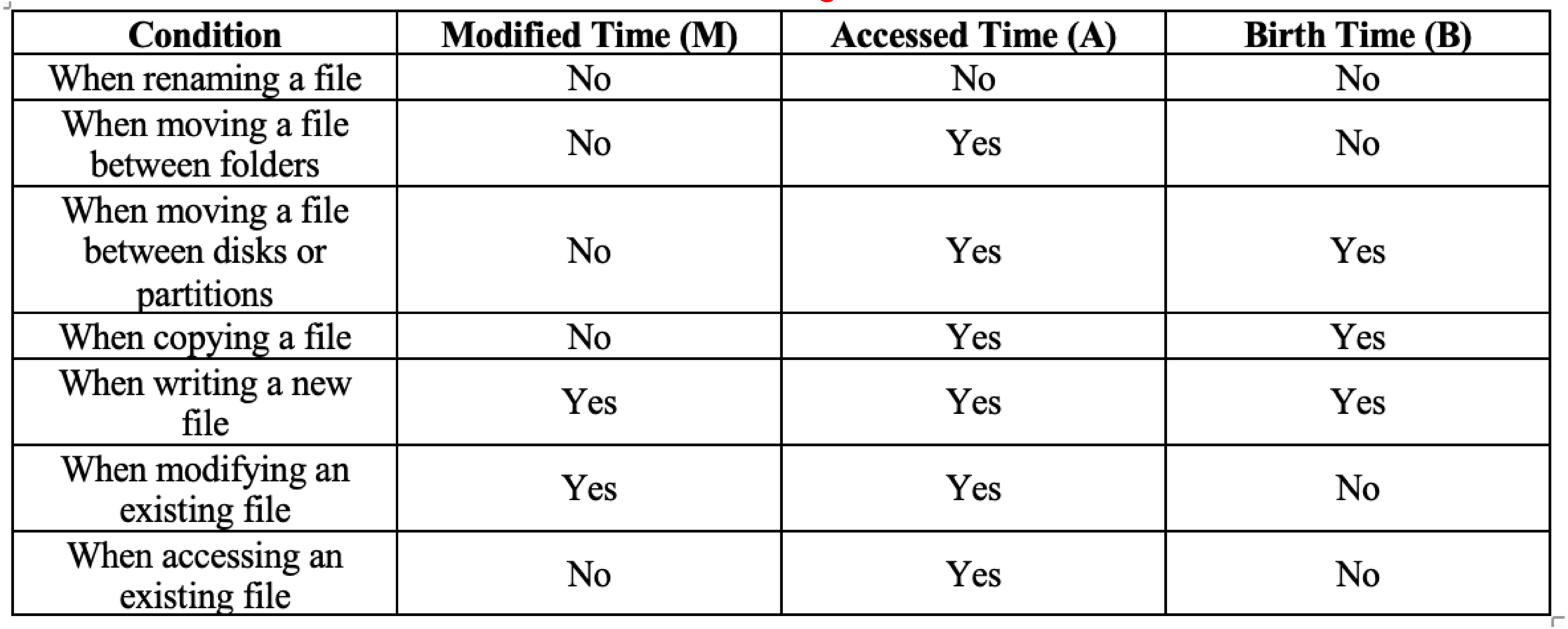
```
Get-Content "C:\path\to\your\file.txt”
```

**Observation:** Only the Accessed Time will update.

**Let's give it a try.**

# How does the Timestamp change in windows

## Summary



| Condition | Modified Time (M) | Accessed Time (A) | Birth Time (B) |
|---|---|---|---|
| When renaming a file | No | No | No |
| When moving a file between folders | No | Yes | No |
| When moving a file between disks or partitions | No | Yes | Yes |
| When copying a file | No | Yes | Yes |
| When writing a new file | Yes | Yes | Yes |
| When modifying an existing file | Yes | Yes | No |
| When accessing an existing file | No | Yes | No |

# Linux Timestamp

## What is inode in Linux System?

1: It is a data structure used to store information about a file or directory, excluding its name and actual data content.

2: Each file and directory in a file system has a unique inode, which contains metadata about the file, including its size, permissions, owner, and timestamps.

## What information does an inode contain?

**File Type**: Whether it’s a regular file, directory, symbolic link, etc.

**Permissions**: Read, write, and execute permissions for the owner, group, and others.

**Owner and Group IDs**: User and group identifiers that own the file.

**Size**: The size of the file in bytes.

**Number of Links**: The number of directory entries that refer to this inode (i.e., hard links).

**Timestamps**: Metadata about the times related to the file’s state, including access, modification, and status change times.

**Data Block Pointers**: Pointers to the blocks on disk that store the actual file data.

## How inodes and timestamps work together?

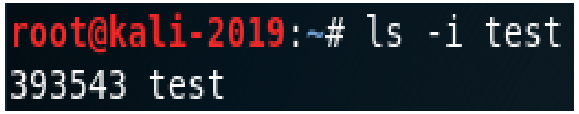
**Storage of Timestamps**: Timestamps are stored as part of the inode structure. When a file is created, accessed, modified, or has its metadata changed, the corresponding timestamp in the inode is updated.

**Updates and Synchronization**: When specific actions are performed on a file (like reading, writing, or changing permissions), the timestamp in the inode is automatically updated by the file system. This ensures that the inode always reflects the latest status and history of the file.

# How does the Timestamp change in Linux

## How to find the inode number of a file?

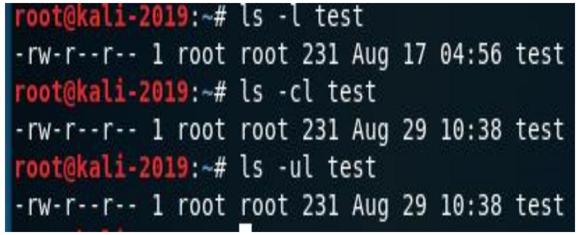
Command: ls -i “file name”



```
root@kali-2019:~# ls -i test
393543 test
```

## How to display the MAC time?

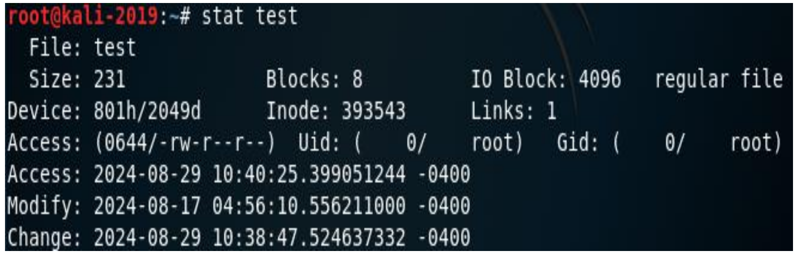
Command for Modified time: ls -l “file name”
Commonad for Changed time: ls -cl “file name”
Commonad for Access time: ls -ul “file name”



```
root@kali-2019:~# ls -l test
-rw-r--r-- 1 root root 231 Aug 17 04:56 test
root@kali-2019:~# ls -cl test
-rw-r--r-- 1 root root 231 Aug 29 10:38 test
root@kali-2019:~# ls -ul test
-rw-r--r-- 1 root root 231 Aug 29 10:38 test
```

## How to get detailed file information?

Command: Stat “file name”



```
root@kali-2019:~# stat test
  File: test
  Size: 231             Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 393543      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2024-08-29 10:40:25.399051244 -0400
Modify: 2024-08-17 04:56:10.556211000 -0400
Change: 2024-08-29 10:38:47.524637332 -0400
```

## Summary

| Condition | Modified Time (M) | Accessed Time (A) | Changed Time(C) |
|---|---|---|---|
| When renaming a file | No | No | Yes |
| When moving a file between folders | No | No | Yes |
| When moving a file between disks | No | No | Yes |
| When copying a file | No | No | Yes |
| When writing a new file | Yes | Yes | Yes |
| When modifying an existing file | Yes | Yes | Yes |
| When accessing an existing file | No | Yes | No |

**Let's give it a try ourselves**

# What is Timestamp Manipulation?

**Definition:** Timestamp manipulation involves altering the time-related metadata associated with files and directories on a system.

**By changing these timestamps**, attackers can:

**1: Make malicious files appear older or newer than they actually are**, blending them in with legitimate files and making detection harder.

**2: Erase evidence of access** by changing Accessed Times, making it seem as though no one viewed or accessed sensitive files.

**3: Confuse investigators** by setting misleading timestamps that disrupt the chronological sequence of events, making it difficult to piece together what actually happened.

# How Attackers Manipulate Timestamps

In Linux, commands like touch can change the Accessed and Modified timestamps of a file:

```
touch -a -m -t 202308271200.00 /path/to/file.txt
```

This command changes the Accessed (-a) and Modified (-m) timestamps of file.txt to August 27, 2023, 12:00 PM.

In Windows, PowerShell commands or even simple file attribute editing utilities can be used to modify timestamps:

```
(Get-Item $file).CreationTime = $date
(Get-Item $file).LastWriteTime = $date
(Get-Item $file).LastAccessTime = $date
```

In Windows, Using Tools like Timestomp allows attackers to alter all timestamps (ctime, mtime, atime) of a file on NTFS file systems:

```
timestomp.exe file name.file type -m "01/01/2023 12:00:00" -a "01/01/2023 12:00:00" -c "01/01/2023 12:00:00"
```

# Thank you

SWINBURNE UNIVERSITY OF TECHNOLOGY