CHAPTER 11

Disk Forensics

Chapter Overview

01

Forensic Tools & Timestamps

Autopsy, FTK, MFT, and MACE timestamp analysis

02

Execution & Access Artifacts

Prefetch, LNK files, jump lists, and SRUM

03

Registry & Browser Artifacts

Persistence, user activity, and browsing history

04

Advanced Artifacts

USN journal, shadow copies, email, and automated triage

Forensic Tools & Timestamps

The foundation of disk forensics

Forensic Tools

Open-Source Tools

Autopsy / Sleuth Kit: full-featured disk forensics suite

KAPE (Kroll Artifact Parser): targeted collection + processing

Plaso / log2timeline: timeline generation from multiple artifacts

RegRipper: automated registry analysis

Eric Zimmerman Tools: fast, purpose-built parsers (PECmd, LECmd, MFTECmd)

MACE Timestamps

M: Modified — file content last changed

A: Accessed — file last opened

C: Changed (MFT) — MFT entry last modified

E: Created — file creation time

Timestomping: attackers modify timestamps to confuse timelines

$STANDARD_INFORMATION vs. $FILE_NAME: compare both to detect stomping

MFT — Master File Table Analysis

MFT: every file and directory on an NTFS volume has an MFT entry (1 KB each)

- **Each entry contains**: file name, timestamps (MACE), size, parent directory, data attribute
- **MFT entry survives deletion**: marked as 'unallocated' but remains until overwritten

MFTECmd (Eric Zimmerman): parses $MFT to CSV — fast, comprehensive

MFT analysis reveals: files created during attack, files deleted after, renamed tools

$FILE_NAME attribute: timestamps set at creation, harder to tamper than $STANDARD_INFORMATION

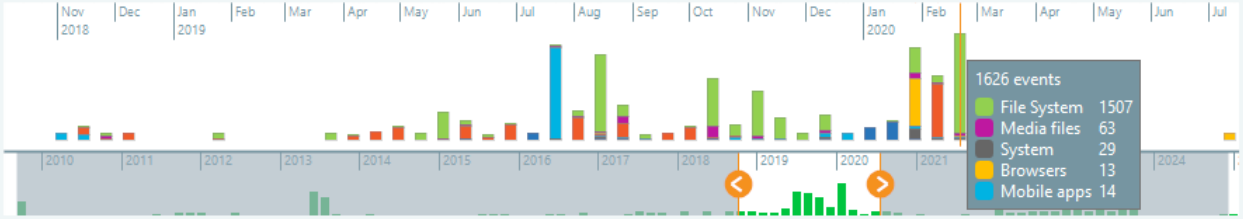
Timestomping detection: $STANDARD_INFORMATION timestamps earlier than $FILE_NAME timestamps

Timeline from MFT: all file system activity for entire volume in chronological order

Belkasoft Evidence Center X | v.2.7.19171 NOT FOR RESALE | Timelines

Dashboard Artifacts Timeline

Items: 7719 of 18962 Filtered by: Date range



Category	Item type	Time (UTC)	Info	Event type
File System		2018-11-21 13:53:37	decryption option.txt	File modified
Media files		2018-11-21 13:53:37	decryption option.txt	File modified
Chats		2018-11-22 10:59:02	hi!	Message sent
File System		2018-11-23 10:21:52	Gun.jpg	File modified
Media files		2018-11-23 10:21:52	Gun.jpg	File modified
File System		2018-11-23 10:27:34	Huntingclub.jpg	File modified
Media files		2018-11-23 10:27:34	Huntingclub.jpg	File modified

Hex Picture preview

Properties

File

File name	Gun.jpg
Path	image:\1\vol_0\Pictures\GunDetection\Gun.jpg
Offset (bytes)	190488576
File size (bytes)	47367
Created (UTC)	2023-10-17 09:16:41
Created (local)	2023-10-17 12:16:41
Modified (UTC)	2018-11-23 10:21:52
Modified (local)	2018-11-23 13:21:52
Access time (UTC)	2023-10-17 09:16:41
Access time (local)	2023-10-17 12:16:41
Saved to the database	No

General

Width (px)	736
Height (px)	552
Is deleted	No

Execution & Access Artifacts

Evidence that programs ran and files were opened

Prefetch & LNK Files / Jump Lists

Prefetch: C:\Windows\Prefetch*.pf — evidence a program was executed

Stores: executable name, run count, last run time, files/dirs accessed

Enabled on workstations by default, disabled on servers

LNK (Link) files: auto-created when files are opened via Explorer

LNK files contain: original path, volume serial, MAC address, timestamps

Jump Lists: recently/frequently accessed files per application

Tools: PECmd (Prefetch), LECmd (LNK), JLECmd (Jump Lists) by Eric Zimmerman

SRUM & Registry Analysis

SRUM (System Resource Usage Monitor): tracks app network/CPU/energy use for 30+ days

SRUM database: C:\Windows\System32\sru\SRUDB.dat (SQLite)

Records: bytes sent/received per process — even after process is deleted

Registry hive locations: SYSTEM, SAM, SOFTWARE, NTUSER.DAT, UsrClass.dat

Key forensic keys: **UserAssist** (GUI program execution), **ShimCache** (app compatibility)

MRU lists: recently opened files, typed paths, run commands

USBSTOR: devices ever connected to this system

UserAssist, ShimCache & AmCache

UserAssist: NTUSER.DAT key tracking GUI application execution — ROT-13 encoded paths

- **UserAssist fields**: last execution time, run count, focus count, focus duration

ShimCache (AppCompatCache): records every executable OS considers for compatibility shims

- **ShimCache key**: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache
- ShimCache proves file EXISTED on system — not necessarily that it ran

AmCache (Amcache.hve): first execution time, full path, SHA-1 hash of executable

- AmCache is the strongest execution evidence on modern Windows (Vista+)

Tools: AppCompatCacheParser.exe (ShimCache), AmcacheParser.exe (AmCache) by Eric Zimmerman

Advanced Artifacts

Deep-dive forensic evidence sources

USN Journal, Shadow Copies & Triage

USN Journal & Browser Activity

USN Journal: NTFS change journal — records file creates, deletes, renames

Survives file deletion — deleted malware may still appear in USN

Browser artifacts: history, cache, downloads, cookies, passwords

Firefox: places.sqlite | Chrome/Edge: History (SQLite)

Web cache: actual file content from visited URLs

Volume Shadow Copies & Triage

VSS: point-in-time snapshots of NTFS volumes

Attackers delete shadow copies (vssadmin delete shadows /all)

If available: recover deleted files and earlier file versions

Automated triage: KAPE, Velociraptor, Autopsy Ingest modules

Linux/UNIX: bash_history, /var/log/auth.log, /var/log/syslog, cron tabs

Browser Forensics

Chrome / Edge (Chromium): History, Cache, Cookies, Login Data — all SQLite databases

Chrome profile path: C:\Users\<user>\AppData\Local\Google\Chrome\User Data\Default\

History DB: SELECT url, title, visit_count, last_visit_time FROM urls

Cache: actual file content from web pages — may include downloaded payloads

Firefox: places.sqlite (history + bookmarks), cookies.sqlite, formhistory.sqlite

Browser history reveals: what attacker researched, C2 panel accessed, what was downloaded

BrowsingHistoryView (NirSoft): parses all browsers in one GUI

Hindsight: open-source Chrome forensics tool with timeline output

Email Forensics & KAPE Collection

Outlook PST/OST: C:\Users\<user>\AppData\Local\Microsoft\Outlook*.pst

PST: offline archive file — portable, readable by Outlook or forensic tools

Key artifacts: spear-phishing emails with malicious attachments or links

Email headers: Received chain reveals true origin even if From: is spoofed

KAPE targets: pre-defined collection recipes — over 200 built-in targets

!BasicCollection: Event Logs, Prefetch, Registry Hives, SRUM, LNK files, Jump Lists

KAPE command: kape.exe --tsource C:\ --tdest D:\evidence --target !BasicCollection --module !EZParser

KAPE vs full image: KAPE in 10 minutes vs. 2+ hours for full disk image

Super Timeline & Anti-Forensics

Super timeline: merge ALL artifact timestamps into one chronological sequence

log2timeline / plaso: ingests dozens of artifact types, outputs .plaso binary store

psort: filter and output plaso store to CSV, JSON, or Timesketch

Timesketch: collaborative web-based timeline analysis platform

Timeline Explorer (Eric Zimmerman): fast Windows GUI for timeline CSV analysis

Timestomping: $STANDARD_INFORMATION timestamps modified by user-mode tools — $FILE_NAME is harder

Log clearing: wevtutil cl Security — clears Security.evtx; leaves Event 1102 as evidence

VSS deletion: vssadmin delete shadows /all — Event ID 524 + Sysmon captures vssadmin.exe

Windows Event Log Artifacts on Disk

EVTX location: C:\Windows\System32\winevt\Logs\

Key log files: Security.evtx, System.evtx, Application.evtx, PowerShell/Operational.evtx

EvtxECmd: batch parse all EVTX files in a directory to CSV for timeline analysis

EVTX after clearing: Event 1102 will appear in new Security.evtx even after clear

Shadow copy recovery: VSS may contain older EVTX with events before clearing

Rollback attack: attacker replaces current EVTX with older version — check MFT timestamps

SIEM forwarded events: local logs cleared does not mean all evidence is gone

Plaso supports EVTX: ingests event logs alongside filesystem and registry into super timeline

File Deletion and Recovery

Deleted file: MFT entry marked unallocated; data clusters marked free but not overwritten

Recovery: use Autopsy, PhotoRec, or FTK to carve deleted files from unallocated space

File carving: search for file headers/footers (magic bytes) in unallocated space — tool: scalpel

NTFS: $Recycle.Bin contains deleted files + $I metadata (original path, deletion time, size)

Volume Shadow Copy: vssadmin list shadows — if present, access previous versions via mklink

Secure deletion tools: sdelete -z C: — overwrite free space; prevents recovery (from SysInternals)

Evidence of secure deletion: uniform entropy in unallocated space instead of varied patterns

MFT+USN journal: even after file deletion, MFT entry and USN record the file existed

Linux and Unix Artifacts

bash_history: ~/.bash_history — records commands run by each user (may be cleared by attacker)

auth.log / secure: /var/log/auth.log (Debian) or /var/log/secure (RHEL) — SSH, sudo, su events

syslog: /var/log/syslog or /var/log/messages — general system events and service starts

cron tabs: /var/spool/cron/, /etc/cron.* — scheduled tasks used for persistence

persistence locations: /etc/rc.local, systemd units (/etc/systemd/system/), init.d scripts

Web server logs: /var/log/apache2/, /var/log/nginx/ — webshell access and exploitation

Inode timestamps: stat filename shows Access/Modify/Change/Birth times

Linux artifacts in Autopsy: supports ext4/xfs/btrfs — same timeline approach as Windows

Conclusion

Disk forensics reconstructs attacker activity even after deletion and evasion

Prefetch, LNK files, and SRUM prove program execution without process list

Registry MRU lists and UserAssist reveal user and attacker activity timeline

USN journal records file system changes — deleted malware leaves a trace

Anti-forensics leaves its own traces — log clearing, timestomping, and VSS deletion are detectable

Automated triage tools (KAPE, Velociraptor) dramatically reduce analysis time

Knowledge Check