

MODULO 03 • AGOSTO 2026

Analisi di File System e Artefatti

NTFS, ext4, APFS; registry, log ed evidenze di sistema:
ricostruire l'attività dell'utente dai dati su disco

\$MFT • \$LogFile • USN Journal • prefetch • shellbag

Corso di Laurea Magistrale in Sicurezza Informatica

Al termine del modulo lo studente sarà in grado di

01 • INTERPRETARE

Le strutture di NTFS, ext4 e APFS: metadati, timestamp, aree non allocate e meccanismi di journaling

02 • RICOSTRUIRE

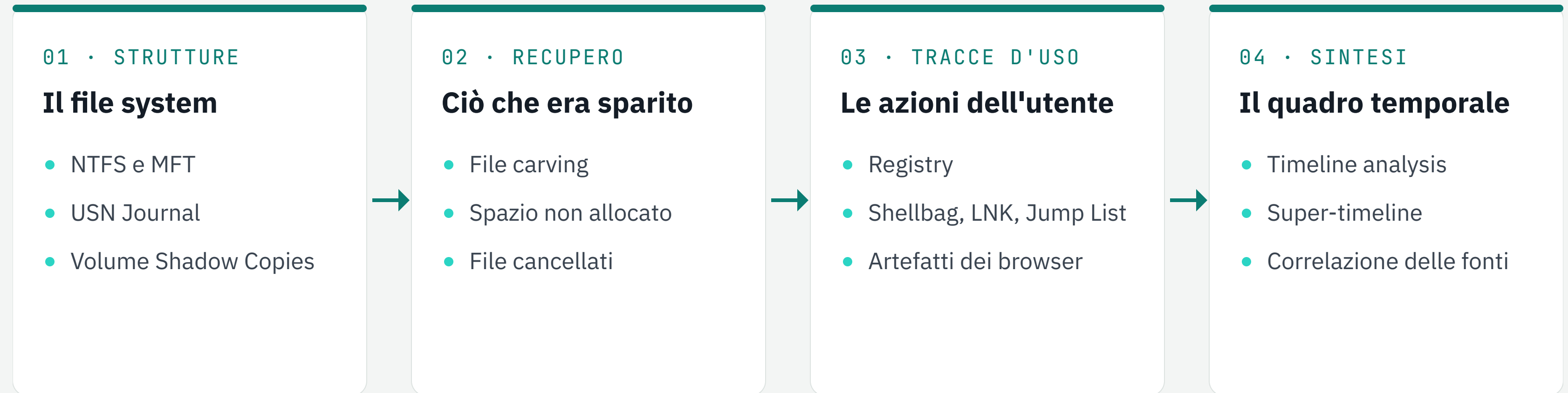
L'attività dell'utente dagli artefatti di Windows, Linux e macOS, con il carving di file cancellati

03 • COSTRUIRE

Una super-timeline degli eventi con Plaso/log2timeline e saperla interrogare criticamente in Timesketch

Avvertenza interpretativa. Ogni artefatto va interpretato: un timestamp ha una semantica che dipende dal file system e dal sistema operativo. L'errore tipico del principiante è trattare i metadati come fatti accertati, senza contestualizzarli.

Dove guardare: le quattro aree del file system



Le quattro aree si susseguono nelle slide del modulo nello stesso ordine dello schema: ciascuna presuppone la precedente.

NTFS, ext4 e APFS a confronto

File system	Strutture chiave	Rilevanza forense
NTFS (Windows)	\$MFT, \$LogFile, USN Journal, ADS	La MFT registra ogni file con 4 timestamp (\$SI e \$FN): confrontarli smaschera il timestomping; il journal traccia le operazioni recenti
ext4 (Linux)	Inode, journal, extent tree	Timestamp alla granularità del nanosecondo; il journal conserva transazioni di metadati utili al recupero
APFS (macOS)	Container, snapshot, copy-on-write	Gli snapshot congelano stati passati del volume; la cifratura nativa condiziona la strategia di acquisizione

Aree non allocate e slack space

Il contenuto dei file cancellati può restare sul disco: il **file carving** lo recupera dalle signature, senza metadati.

Semantica dei timestamp

MACB (Modified, Accessed, Changed, Birth): ogni file system li aggiorna secondo regole proprie.

\$SI contro \$FN: smascherare il timestomping

In NTFS ogni file ha un record nella MFT con **due set di timestamp**: l'attributo **\$STANDARD_INFORMATION** (\$SI), quello che Windows mostra e che le API permettono di modificare, e l'attributo **\$FILE_NAME** (\$FN), aggiornato dal kernel e non alterabile con gli strumenti ordinari.

Gli strumenti di **timestomping** riscrivono \$SI per retrodatare o confondere; quasi mai riescono a toccare \$FN. La **divergenza tra i due set**, in particolare un \$SI anteriore al \$FN, è il segno classico della manipolazione.

ESEMPIO

contratto.docx

\$SI created: 12/01/2023

\$FN created: 05/09/2025

Il file «risulta» del 2023, ma il kernel lo ha visto nascere nel 2025: coerente con un documento retrodatato per simulare un accordo preesistente. La divergenza va spiegata, anche se da sola non prova il dolo.

Strumenti di elezione: **MFTECmd** per il parsing della MFT, confronto sistematico \$SI/\$FN su tutto il volume; granularità dei timestamp (precisione «tonda» al secondo) come ulteriore indizio di scrittura artificiale.

\$LogFile e USN Journal

\$LogFile

Journal transazionale di basso livello: registra le modifiche ai metadati per garantirne la coerenza dopo un crash. Copre poche ore con il massimo dettaglio, incluse operazioni che altrove non lasciano traccia.

USN Journal (\$UsnJrnl)

Registro ad alto livello delle variazioni: creazione, rinomina, cancellazione, modifica, con nome file, momento e ragione. Finestra più ampia (giorni o settimane): è la fonte di elezione per ricostruire «che cosa è successo di recente» a un file.

ESEMPIO • Il file «bilancio_reale.xlsx» non risulta più allocato nella MFT. L'USN Journal conserva però, in sequenza, un **RENAME** in «temp.xlsx» e poi un **DELETE**, tre minuti prima dell'ora in cui l'utente dichiara di aver lasciato la postazione. Il journal documenta la **condotta** e il suo orario; il contenuto si recupera, se possibile, con il carving.

Cautela: i journal ruotano e vanno acquisiti presto; l'assenza di un record indica solo il limite della finestra.

Le versioni precedenti nelle shadow copy

Il servizio **VSS** (Volume Shadow Copy Service) di Windows fotografa lo stato del volume a intervalli regolari: prima degli aggiornamenti, dei punti di ripristino, di alcune installazioni. Ogni snapshot è **differenziale**: conserva soltanto i blocchi sovrascritti da allora.

Per l'analista significa poter leggere **versioni precedenti** di file modificati o cancellati, chiavi di registro incluse: una timeline «verticale» che affianca quella dei metadati. Gli snapshot si montano dalla copia forense (vshadowmount, librerie libvshadow) senza toccare il reperto.

Attenzione: lo spazio dedicato è limitato e gli snapshot più vecchi vengono eliminati a rotazione; la finestra temporale coperta varia da giorni a mesi.

ESEMPIO • IL DOCUMENTO «RIPULITO»

L'indagato consegna un contratto in formato DOCX apparentemente regolare. Nella shadow copy di tre settimane prima, lo stesso file contiene una clausola poi rimossa.

Il confronto fra le due versioni, entrambe con hash e provenienza documentati, diventa l'elemento centrale della consulenza, perché mostra *come* il file è cambiato nel tempo.

Recuperare senza metadati: il carving e i suoi limiti

Quando il riferimento nella MFT è perso, il contenuto può sopravvivere nelle aree non allocate. Il carving lo cerca per **signature**: sequenze di byte che aprono e chiudono i formati noti.

```
JPEG → FF D8 FF ... FF D9  
PDF → 25 50 44 46 (%PDF) ... %%EOF  
ZIP/DOCX → 50 4B 03 04 (PK..)
```

Il file recuperato è **orfano**, privo di nome, percorso e timestamp. La collocazione nel tempo e nel contesto va ricostruita da altre fonti (journal, contenuto stesso, EXIF interni).

Limiti da dichiarare

- **Frammentazione**: file spezzati producono recuperi parziali o corrotti
- **SSD con TRIM**: i blocchi liberati vengono azzerati: il carving spesso non trova nulla
- **Falsi positivi**: signature trovate in dati casuali o annidati
- **Sovrascrittura**: ciò che è stato riallocato è perso

Strumenti: PhotoRec, scalpel, il carver integrato di Autopsy.

Il sistema operativo come testimone involontario

Windows

- **Registry:** dispositivi USB, reti, programmi eseguiti
- **Prefetch:** esecuzioni e loro frequenza
- **Shellbag:** cartelle visitate, anche su supporti rimossi
- **File LNK e Jump List:** documenti aperti
- **Event Log:** accessi, servizi, errori

Linux

- **journald / syslog:** eventi di sistema
- **bash history:** comandi dell'utente
- **auth.log / wtmp:** accessi e sessioni
- **cron e systemd timer:** persistenza
- File di configurazione e pacchetti installati

macOS

- **Unified Log:** telemetria granulare di sistema
- **plist:** preferenze e stato delle applicazioni
- **FSEvents:** modifiche al file system
- **KnowledgeC:** pattern d'uso delle applicazioni
- **Quarantine:** provenienza dei file scaricati

Metodo. Un fatto si considera ricostruito quando più artefatti indipendenti concordano tra loro.

Tracce di una chiavetta USB nel registry

Il registry è il database di configurazione di Windows (hive SYSTEM, SOFTWARE, NTUSER.DAT per utente): registra dispositivi, reti, programmi, preferenze. Il caso d'uso canonico è provare che una specifica chiavetta è stata collegata:

1. **SYSTEM\...\USBSTOR**: marca, modello e seriale del dispositivo, con timestamp di prima e ultima connessione
2. **SYSTEM\MountedDevices**: lettera di unità assegnata a quel seriale
3. **NTUSER.DAT (MountPoints2)**: *quale utente* ha montato quell'unità
4. **Shellbag e LNK**: cartelle visitate e file aperti sull'unità, anche dopo la rimozione

ESEMPIO • LA CATENA COMPLETA

Esfiltrazione sospetta: USBSTOR mostra una chiavetta [seriale] collegata alle 18:42; MountPoints2 la lega all'account dell'indagato; un LNK documenta l'apertura di «listino_clienti.xlsx» da quell'unità alle 18:44; il USN Journal registra la copia del file due minuti prima.

Le quattro fonti, indipendenti tra loro, descrivono una sequenza coerente.

Strumenti: **Registry Explorer / RECcmd** (Zimmerman); i file .LOG degli hive contengono modifiche non ancora consolidate.

Shellbag, file LNK e Jump List

Shellbag

Il registro conserva vista e posizione di ogni cartella **aperta in Explorer**, anche se poi cancellata o su supporti rimovibili e di rete scollegati. Attestano la *navigazione*.

File LNK

Ogni apertura di documento genera un collegamento con **percorso originale, timestamp, volume e numero di serie** del supporto. Un LNK può sopravvivere anni al file cui puntava.

Jump List

Gli elenchi «recenti» per applicazione (AutomaticDestinations) registrano **quale programma ha aperto quale file e quando**: l'anello che unisce utente, applicazione e documento.

ESEMPIO • Un file riservato non è più sul disco. Uno shellbag mostra la cartella «E:\riservato» aperta su un volume USB con seriale noto; un LNK registra l'apertura del file con il percorso completo; la Jump List di Excel la colloca alle 14:32 di un giorno preciso. Il valore probatorio deriva dalla coerenza fra i tre artefatti indipendenti.

Gli artefatti del browser: la vita online dell'utente

Chromium e Firefox conservano la quasi totalità dei dati in **database SQLite**: cronologia (History/places.sqlite), download con percorso di destinazione e origine, cookie, dati dei form, motori di ricerca interrogati. I timestamp usano epoche diverse (WebKit, Unix, PRTIME): la **conversione corretta va verbalizzata**.

La cronologia registra anche i **termini di ricerca** e il numero di visite per URL: elementi che in giudizio pesano quanto i file trovati, perché documentano l'*intenzione*. La navigazione «in incognito» non scrive questi database, ma lascia tracce in RAM, DNS cache e, talvolta, nei file di sessione.

La sincronizzazione cloud dei profili estende l'analisi oltre il singolo dispositivo, con le cautele giuridiche del caso (modulo 11).

ESEMPIO • LA RICERCA CHE PRECEDE IL FATTO

In un caso di frode, la cronologia mostra ricerche su «come eliminare metadati PDF» tre giorni prima della consegna di un documento contestato.

La ricerca da sola non prova la manipolazione; unita alle anomalie nei metadati del PDF, colloca nel tempo una condotta e ne suggerisce la deliberatezza. L'analista riferisce i fatti accertati, la cui qualificazione giuridica spetta al giudice.

La super-timeline: un'unica sequenza di eventi

Plaso/log2timeline estrae eventi datati da centinaia di formati (file system, registry, log, browser, applicazioni) e li normalizza in un'unica sequenza cronologica, esplorabile in **Timesketch**.

La super-timeline trasforma la domanda «cosa contiene il disco?» nella domanda investigativamente corretta: **«cosa è accaduto, e in quale ordine?»**

```
log2timeline.py --storage-file caso.plaso disco.E01  
psort.py -o dynamic -w timeline.csv caso.plaso
```

Cautele interpretative

- Fusi orari e clock skew: normalizzare sempre in UTC
- Timestomping: incrociare \$SI con \$FN
- Il volume degli eventi non equivale alla loro rilevanza
- La timeline suggerisce ipotesi da verificare sull'artefatto originale

Dall'immagine ai fatti ricostruiti

Laboratorio

- Immagine **CFReDS** in Autopsy
- Registry, prefetch e shellbag
- Carving di file cancellati
- Super-timeline con Plaso

DELIVERABLE: cronologia
ricostruita e motivata

Letture obbligatorie

- B. Carrier, *File System Forensic Analysis*, capp. su NTFS
- SANS, *Windows Forensic Analysis* poster
- Documentazione Plaso e Timesketch

Verifica

Ogni affermazione della cronologia va ancorata ad artefatti identificati e corroborata da fonti indipendenti.

Prossimo modulo: fondamenti di machine learning per il forenser (M7, traccia core).

L'analisi procede dagli artefatti ai fatti: ogni ipotesi viene dichiarata e verificata sulle fonti.



Grazie per l'attenzione

Vincenzo Calabrò

Domande, materiali del corso e contatti
sul sito del docente



vincenzocalabro.it