

PER ALTRI APPUNTI CONSULTARE IL SITO:

https://luigi-v.github.io/Appunti_Universita/

1. DIGITAL FORENSICS

La **digital forensics** (investigazione digitale forense) consiste nell'uso di metodi scientificamente provati, per le attività di:

conservazione – raccolta – convalida – identificazione - analisi – interpretazione – documentazione - presentazione

di **dati digitali**, derivati da **dispositivi informatici**, con lo scopo di:

- semplificare la **ricostruzione** di eventi criminali o azioni illegali;

- contribuire ad **anticipare** le azioni **non autorizzate** con l'obiettivo di definire operazioni pianificate per evitare tali azioni.

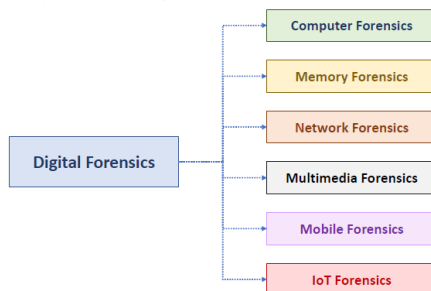
Al centro di ogni investigazione digitale forense, vi sono sicuramente le **digital evidence** (*prove digitali* o *evidenze digitali*).

Una **prova digitale** è **qualsiasi dato digitale**, contenente informazioni utilizzabili per **supportare** o **confutare** ipotesi di un crimine.

In generale, le prove digitali (digital evidences) individuate all'interno di un dispositivo informatico, possono essere utilizzate (in base alle legislazioni dello Stato) per supportare o confutare le seguenti azioni illegali:

Omicidio e atti di violenza, Spionaggio industriale, Frodi, riciclaggio di denaro e furto, Estorsione, Coinvolgimento con i narcotici, Pedofilia e cyberstalking, Terrorismo, Violenza familiare, ecc...

Branche della Digital Forensics:



ORIGINI DELL'INVESTIGAZIONE FORENSE:

Alphonse Bertillon introdusse un **processo di documentazione** di **elementi** (armi, oggetti, ecc.) in una **scena del crimine** mediante **fotografie** e affermava che la **scienza** e la **logica** dovrebbero essere utilizzate per **investigare** e **risolvere** il crimine.

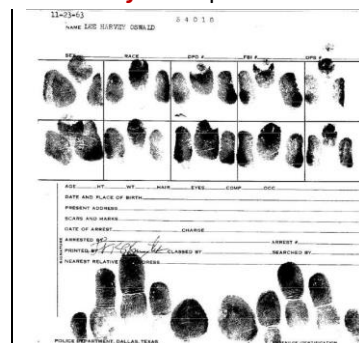
Il lavoro di Bertillon ha influenzato Edmond Locard che introdusse il **principio di Scambio di Locard**: una **azione criminale** di un individuo **non può verificarsi senza lasciare un segno**.

Principio forense fondamentale basato sullo scambio comune di **tracce fisiche** (Impronte digitali, tracce di DNA o residui di polvere da sparo) su una scena del crimine. Le tracce raccolte sulla scena aiutano a **ricostruire** quello che è successo e **identificare** i presenti.

Ad oggi, le tecniche di analisi forense si sono notevolmente evolute dall'epoca di Bertillon e Locard.

Tuttavia, essi hanno introdotto **tre concetti fondamentali** che assistono gli investigatori e la giustizia penale:

- **Documentazione** relativa alla scena del crimine;
- Identificazione di elementi utili per l'indagine (**tracce**);
- Disciplina dell'**analisi** delle tracce.



IMPRONTE DIGITALI:

Edward Henry inventò un sistema di classificazione di impronte, fu presentato presso la polizia di Londra, successivamente vi fu la prima condanna giudiziaria grazie alle impronte digitali.

Però, l'affidabilità delle prove relative alle impronte digitali, è stata recentemente contestata in diverse giurisdizioni, con preoccupazioni per la mancanza di standard validi per la valutazione di due impronte «imprese» su carta.

DNA:

Tramite l'acido desossiribonucleico (DNA) è possibile determinare le caratteristiche ereditarie di ogni persona. Il DNA può essere estratto da Saliva (es., da francobolli usati, buste, il filo interdentale) o Campioni ematici (es., da rasoi usati, i capelli, i vestiti, ecc.).

Nel 1987 le prove basate su DNA vengono utilizzate per la prima volta per avere un risultato affidabile per la comparazione del DNA di un sospetto ed il DNA individuato nella scena del crimine.

Le prove ottenute dal DNA sono state usate anche in "casi freddi" (detti anche *cold case*, ovvero casi irrisolti), dimostrando l'innocenza di soggetti ingiustamente condannati. Da sottolineare che, data la complessità delle prove del DNA, molte giurie hanno esitato in relazione a tali prove. Con l'evoluzione delle tecniche, invece, tali prove sono state maggiormente accettate nelle corti.

PASSI FONDAMENTALI DI UN ESAME FORENSE:

- **Preservare la Scena del Crimine**: passo fondamentale ed è importantissimo. Se l'evidenza è contaminata, persa o semplicemente non identificata e/o trascurata, tutto ciò che segue può avere un **valore limitato** per gli investigatori, i quali mettono insieme le prove del caso;
- **Riconoscere le Prove**: riconoscere le prove e identificarle risulta estremamente rilevante. Individuare i punti in cui cercare può solo migliorare l'esito di un esame forense. Una volta individuate, le prove devono essere **raccolte** e **classificate**;
- **Visione d'Insieme**: le prove **non possono essere viste in maniera isolata**. Dovrebbero essere confrontate con altre prove e dovrebbero essere identificate prove «effettive». A tal punto, dovrebbe essere descritte in termini scientifici.

CYBERCRIME:

Le **informazioni digitali** sono memorizzate sui dispositivi informatici e possono essere sfruttate anche per attività *non autorizzate* e/o *illeghi*. Il notevole incremento del desktop computing ha consentito la proliferazione anche della criminalità informatica (**cybercrime**).

Un **cybercrimine** è un atto criminale compiuto utilizzando un dispositivo informatico e/o tramite internet. Questo atto **trascende i confini nazionali** e **internazionali** e solleva diversi problemi giurisdizionali, che una Nazione, da sola, non può mitigare.

NASCITA DELLA DIGITAL FORENSICS:

In risposta al **cybercrime** e all'utilizzo di sistemi informatici come oggetto di **crimini**, è emersa la **Digital Forensics**.

La Digital Forensics getta effettivamente le basi, come **disciplina**, negli anni '80. Le **principali motivazioni** sono state la maggiore accessibilità dei computer, sia economica sia nell'usabilità, e le prime interconnessioni tra computer mediante reti locali.

Una grande problematica, però, è la **obsolescenza** delle leggi tradizionali e degli standard legali.



Esempi di problematiche:

1. Il furto di un dispositivo informatico (smartphone/tablet, computer, ecc.) potrebbe essere confrontato con il furto di informazioni sensibili ottenute dal dispositivo stesso e utilizzate senza autorizzazione legale.
2. Le informazioni possono rimanere sul dispositivo anche se ne è stata effettuata una copia (senza il permesso del proprietario). In tal modo, il «ladro» assumerebbe permanente proprietà delle informazioni (anche se condivise con il proprietario).

In virtù dei crescenti attacchi a computer ed infrastrutture, varie organizzazioni hanno iniziato a realizzare e definire politiche di sicurezza informatica e contromisure. Negli anni dal 1999 al 2007, vi sono state **notevoli evoluzioni** per quanto riguarda la **Digital Forensics**, ad esempio, la possibilità di individuare **azioni del passato** di un individuo mediante tracce digitali come File Cancellati, Note, E-Mail, ecc. La Digital Forensics era considerata principalmente una disciplina di nicchia, al giorno d'oggi, invece, è oggetto di romanzi, fatti di cronaca, serie TV (es., CSI – Scena del crimine, NCIS, ecc.).

CENNI DI LEGISLAZIONE ITALIANA:

Delitto di Garlasco:

Stasi Alberto consegnava spontaneamente alla polizia giudiziaria il proprio computer portatile, quando il reperto informatico veniva consegnato ai consulenti tecnici che procedevano all'effettuazione delle copie forensi dello stesso, i carabinieri accedevano ripetutamente e scorrettamente (senza l'utilizzo delle necessarie tecniche forensi) alla quasi totalità del contenuto del computer.

Il collegio peritale evidenziava che le condotte scorrette di accesso da parte dei carabinieri hanno determinato la sottrazione di contenuto informativo. Le procedure per il trattamento delle **evidenze digitali** non sono state regolamentate fino al 2008.

Successivamente, sono state inoltre apportate modifiche al Codice Penale, in merito alle **evidenze digitali**.

È saggio utilizzare le **best practices**, ovvero tecniche, metodologie, linee guida, ecc., raccolte dalle esperienze più significative, che si considera possano ottenere risultati migliori. Ad esempio, quelle della **UK Association of Chief Police Officers**:

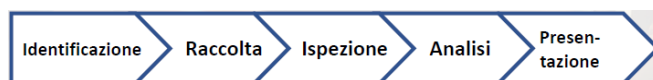
- **Principio 1:** nessuna azione intrapresa dalle forze dell'ordine o dai loro agenti dovrebbe modificare i dati conservati su un dispositivo digitale o supporti di memorizzazione che possono essere successivamente utilizzati in tribunale;
- **Principio 2:** nelle circostanze in cui una persona ritiene necessario accedere ai dati originali su un dispositivo digitale, tale persona deve essere competente a farlo ed essere in grado di fornire prove che spieghino la rilevanza e le implicazioni delle proprie azioni;
- **Principio 3:** dovrebbe essere conservata una traccia documentativa o una registrazione dei processi applicati alla prova elettronica basata su dispositivi. Una terza parte indipendente dovrebbe essere in grado di esaminare tali processi e ottenere lo stesso risultato;
- **Principio 4:** il responsabile dell'indagine ha la responsabilità generale di assicurare che la legge e questi principi siano rispettati.

Workflow della Digital Forensics:

1. L'innescio del workflow è la **denuncia o segnalazione** di un'attività illecita;
2. Segue una fase di **investigazione** che costituisce l'elemento fondamentale;
3. Il tutto culmina in un **dibattimento** in sede giudiziale.



FASI PRINCIPALI DELL'INVESTIGAZIONE:



Il **processo di investigazione** è articolato in 5 fasi principali consecutive:

1. **Identificazione** del crimine e di fonti contenenti potenzialmente prove digitali;
2. **Raccolta** (o **Acquisizione**) di dati *raw* («grezzi»), copiandoli, in maniera opportuna, dai dispositivi digitali;
3. **Ispezione** (**examination**) dei dati raccolti con l'obiettivo di realizzarne una struttura migliore ai fini dell'analisi e della comprensione;
4. **Analisi** si cerca di ottenere una migliore comprensione e si cerca di determinare i fatti di un evento o un'azione illegale;
5. **Presentazione**, le prove digitali, individuate nelle fasi precedenti, vengono adeguatamente presentate nei tribunali e/o negli enti preposti.

Durante il processo di investigazione forense, possono esservi diverse iterazioni di una o più fasi.

1. IDENTIFICAZIONE:

Un crimine può essere identificato, basandosi su: Segnalazioni/Reclami, Denunce, Allarmi, Indicazioni, Ecc.

Definizione:

Nella fase di **identificazione** si ha il compito di **rilevare**, **riconoscere** e **determinare** l'evento o il crimine da **investigare**. L'identificazione è una fase importante, poiché vengono *identificate informazioni* o *fonti di informazioni*. Si identificano anche i dispositivi informatici che potrebbero contenere **prove digitali**, ad esempio Computer desktop.

OSSERVAZIONE IMPORTANTE: Le fonti di prove digitali **NON** sono sempre facili da identificare.

Una corretta pianificazione e preparazione delle attività è una preconditione per una investigazione *efficace* ed *efficiente*.

La scelta degli strumenti e delle tecnologie da impiegare è strettamente dipendente dalla disponibilità di risorse. La **solidità legale** dei suddetti strumenti deve essere preventivamente valutata, in quanto essi devono supportare i principi di integrità.

Nelle scene in cui sono presenti dispositivi informatici, è necessario effettuare una **fase di preparazione**, in cui si configurano adeguatamente hardware e software specifici per l'analisi forense.

- FIRST RESPONDER:

Nel momento in cui viene scoperto o sospettato un crimine, dovrebbe esserci un **first responder** (*primo soccorritore*), il quale deve allertare gli investigatori forensi e convocarli sulla scena del crimine. In generale, il primo soccorritore ha **competenze/conoscenze** in relazione alle infrastrutture informatiche (reti, sistemi operativi, ecc.) ad esempio *Amministratori di Rete, Manager IT, ecc....*

Se il primo soccorritore non avrà competenze sufficienti, dovrà comunque mettere in sicurezza i dati, le periferiche, i supporti di memorizzazioni, ecc. In questo modo non verranno utilizzati, alterati o rimossi da soggetti non autorizzati.

Fra i **doveri** del primo soccorritore troviamo:

1. Effettuare le prime valutazioni;
2. Documentare la scena e la stanza integralmente: il centro della stanza diviene il punto focale della descrizione;
3. Assicurare la scena da soggetti non autorizzati;
4. Preservare e/o impacchettare i dispositivi per il trasporto.

- DOCUMENTAZIONE E PRESERVAZIONE DELLE PROVE:

La **documentazione della scena** dovrebbe essere effettuata dal primo soccorritore, al fine di fornire maggiore supporto agli investigatori. La documentazione dovrebbe includere fotografie, video, registrazioni audio della stanza dove è allocato il dispositivo (Scrivania, entrata/uscita, finestre, prese elettriche, ecc.), stato del dispositivo (acceso/spento/luce di accensione lampeggiante), contenuto dello schermo (se il device è avviato), libri, annotazioni, pezzi di carta e cavi connessi/non connessi.

Il primo soccorritore dovrebbe avere con sé diversi strumenti al fine di svolgere adeguatamente la documentazione e la preservazione delle stesse, come vestiti e occhiali protettivi, braccialetti anti-statici, etichette, adesivi, ecc., torce e lenti di ingrandimento, contenitori, scatole, materiale per l'imballaggio (Evidence Bag).

- LIVE SYSTEM:

Denotiamo con **live system** un **sistema in fase di attività** che potenzialmente detiene prove, le quali sarebbero difficili da acquisire o potrebbero essere perse nel caso in cui il sistema venga spento.

Per i sistemi «attivi», è necessario prestare particolare attenzione a causa della **volatilità dei dati**, siccome con lo spegnimento di un sistema, possono portare alla sovrascrittura di dati su un supporto di memorizzazione (hard disk), perdita dei dati contenuti nella memoria RAM e perdita del file di paging.

IMPORTANTE: Il **file di paging** è un file molto **importante** dal punto di vista della digital forensics.

Alcune **precauzioni** da prendere quando si lavora con i live system:

- Muovere il mouse o spostare leggermente le dita sul touchpad, per verificare se il device è in stato di stand-by o in sospensione;
- Fotografare e registrare lo schermo del dispositivo, considerando tutti i programmi visibili, data, ora e gli oggetti sul desktop;
- Staccare la spina su PC desktop o rimuovere (in caso di notebook) la batteria così che il file di swap rimanga su disco.

- DEAD SYSTEMS:

Denotiamo con **dead system** un **sistema NON in fase di attività**, tutti i dati temporanei (RAM, cache, ecc.) sono tipicamente persi.

I sistemi «inattivi» o «spenti» non dovrebbero **mai essere riaccesi**, se non da parte di un investigatore forense. È necessario adottare attenzioni particolari al fine di garantire che i dati esistenti non vengano cancellati e che non vi sia sovrascrittura dei dati. È inoltre importante accertarsi che il sistema sia effettivamente spento e non sia in stato di sospensione. È comunque consigliato fotografare lo schermo e le porte del PC.

L'analisi dei **dead systems** è denotata come **analisi post mortem** (**post mortem analysis**), mentre l'analisi dei **live systems** è denotata come **live analysis**.

- FILE DI PAGING:

I sistemi operativi hanno la possibilità di utilizzare una porzione del disco fisso come una estensione della memoria RAM: la memoria virtuale (o virtual memory). Il disco fisso è più lento riguardo gli accessi (lettura/scrittura) rispetto alla memoria RAM, ma sistemi con memoria RAM limitata utilizzano la memoria virtuale (memorizzata un file speciale, detto file di paging, page file o file di swap). In questo modo, all'interno del file di paging possono essere memorizzati dati e *processi* che vengono utilizzati di meno rispetto ad altri (lasciando così più spazio nella memoria RAM).

Il file di paging **non è volatile quanto la memoria RAM**, proprio perché esso è memorizzato sul disco fisso, esso dovrebbe essere sempre ispezionato, utilizzando appositi strumenti, poiché potrebbe rivelare informazioni (come password, documenti, ecc.). I dati sulle unità meccaniche (come i dischi fissi) sono tipicamente memorizzati in maniera frammentata. Il vantaggio del file di paging è che i suoi dati vengono tipicamente allocati in **maniera contigua** (lo si fa per fornire un accesso più veloce possibile ai dati). È consigliabile avere un file di paging di dimensioni pari a circa **una volta e mezzo** la dimensione totale della memoria RAM.

- **CHAIN OF CUSTODY (CATENA DI CUSTODIA):**

È necessario tracciare lo **stato di una prova** (una volta identificata) e la relativa **responsabilità** in qualsiasi momento della sua esistenza. Per ciascuna prova devono essere documentati:

- Dove, quando e da chi è stata scoperta e acquisita;
- Dove, quando e da chi è stata custodita o analizzata;
- Chi l'ha avuta in custodia e in quale periodo;
- Come è stata conservata;
- Ad ogni passaggio di consegna, deve essere specificato dove, come e tra chi è stata trasferita (da qui, **chain of custody** o catena di custodia o, abbreviato, **CoC**).

Ogni volta che la prova è affidata ad un nuovo investigatore, nel documento bisogna aggiungere:

- Nome dell'incaricato all'analisi;
- Data e ora di presa in carico del supporto;
- Data e ora di restituzione del supporto.

Gli **accessi alla prova** devono essere estremamente ristretti e chiaramente documentati.

Tipicamente, la CoC è *stampata direttamente* sul contenitore della prova (evidence bag). Oppure, è stampata su un'etichetta da allegare o attaccare al contenitore della prova (evidence bag).

EVIDENCE			
Sottoposta dall'Ente/Autorità			
Data e Ora dell'Acquisizione			
Numero del Referto		Numero del Caso	
Acquisita Da			
Descrizione			
Luogo Acquisizione			
Tipo di Reato			
CHAIN OF CUSTODY			
Ceduta Da		Preso in Custodia Da	
Data e Ora			
Ceduta Da		Preso in Custodia Da	
Data e Ora			
Ceduta Da		Preso in Custodia Da	
Data e Ora			
Esempio di una CoC [NOTA: Adattamento da una CoC reale]			

2. **RACCOLTA:**

La **fase di raccolta** è riferita all'**acquisizione** e/o **copia** di dati digitali. L'investigatore accede al dispositivo informatico, identificato come **rilevante** (nella fase di **identificazione**), contenente dati digitali utili per l'indagine. Al fine di evitare eventuali compromissioni dei dati originali e, conseguentemente, compromettere le prove, è necessario lavorare su delle **copie «esatte» dei dati**.

Definizione:

La **fase di raccolta** (o **acquisizione**) consiste nella copia di dati digitali, utilizzando appropriati e adeguati strumenti e tecniche forensi. Alcune fonti di prove digitali sono:

- **HDD e SSD:**
Dispositivi principali dove vengono memorizzate le informazioni. Questo tipo di supporto è utile per individuare le prove, poiché i dati non sono convenienti da **eliminare definitivamente**. Negli ultimi anni si sono diffusi sempre più i dischi a stato solido (SSD), essendo più veloci, rispetto agli HDD, ed hanno una logica di funzionamento generalmente complessa. Gli SSD memorizzano tipicamente i dati in blocchi, suddivisi in «pagine» composte da grandi array di transistor, detti Negative AND (NAND). A causa della loro natura, gli SSD **svolgono delle operazioni di pulizia «automatica»**, al fine di mantenere veloci gli SSD stessi e allungarne la vita. Ciò comporta, però, possibili **difficoltà nel reperimento di tracce digitali**.
- **RAM:**
All'interno della memoria centrale (Random Access Memory) vengono memorizzati, in binario, dati ed Istruzioni. Le istruzioni sono elaborate dalla Central Processing Unit (CPU). La RAM è una **memoria volatile**. Fare una «istantanea» (**dump**) della RAM può essere molto importante, in quanto la RAM fornisce dettagli sull'uso più recente dell'elaboratore come processi, alcune attività della tastiera e altre evidenze. Tuttavia, realizzare un **dump** della RAM può **contaminare il sistema**.
- **INFRASTRUTTURE DI RETE:**
Qualora i dati dovessero essere memorizzati su server di rete (o altri apparati interconnessi in rete), l'accesso può essere fornito collegando un dispositivo alla rete, specificando eventuali dettagli sull'autenticazione. Tuttavia, in diversi casi, è preferibile creare «copie esatte» del server di rete invece di recuperare i dati tramite l'accesso (logico) al sistema operativo del server.
- **CPU e GPU:**
Potrebbero contenere informazioni memorizzate all'interno delle rispettive cache.

- **PROBLEMI RELATIVI ALLE FONTI DI PROVE DIGITALI:**

I dati e/o i dispositivi hardware potrebbero essere alterati o danneggiati, in maniera **intenzionale**, al fine di rendere difficile l'acquisizione agli investigatori, o **non intenzionale**, guasti meccanici (dovuti ad acqua, polvere, piccoli incendi, ecc.). Talvolta vi è necessità di ricostruire dati da hardware o da dati danneggiati.

È importante sottolineare che vi sono **più minacce per i dati digitali**, rispetto ai cosiddetti dati «cartacei»:

Alcune Minacce
per i dati «cartacei»

- Acqua
- Fuoco e Umidità
- Insetti
- Età
- Disastri naturali

Alcune Minacce
per i dati digitali

- Errori Umani/Negligenze
- Campi elettromagnetici e/o magnetici
- Acqua e Condensa
- Polvere
- Calore
- Impatti fisici
- Voltaggio
- Elettricità statica
- Disastri naturali

- INTEGRITÀ DELLE PROVE DIGITALI:

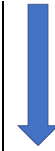
L'**integrità** di una prova è fondamentale per l'investigazione forense. È importante che la prova **non venga alterata** durante la fase di raccolta (durante la copia di file, ecc.). Ci sono dispositivi hardware e strumenti software che **proteggono i dati originali** da modalità diverse dalla lettura. Per verificare se l'integrità delle prove è preservata, si utilizza il concetto di **digital fingerprint**. Si realizza mediante le funzioni crittografiche di hash (dette anche funzioni one-way, *esempi: MD5, SHA-1, SHA-256, ecc.*).

- ORDINE DI VOLATILITÀ DELLE PROVE DIGITALI:

In un'analisi digitale forense, si tiene conto di diverse fonti. È necessario considerare che è **impossibile** ottenere alcuni dati da un sistema, **senza modificarne lo stato**. Per questo motivo si definisce un **ordine di volatilità (Order Of Volatility – OOV)**. I dati «più volatili» devono essere acquisiti prima dei dati «meno volatili».

Definizione:

L'**ordine di volatilità** definisce la priorità con la quale devono essere acquisiti i dati da dispositivi, in base alla volatilità dei dati stessi.

- 
- Registri, memorie di periferiche, ecc.
 - Memoria RAM
 - Stato della Rete
 - Processi in Esecuzione
 - Disco Rigido
 - CD-ROM/DVD/Ecc.

3. ISPEZIONE:

L'**ispezione** richiede la **ristrutturazione**, la **riorganizzazione** ed il **processing** dei dati grezzi. L'obiettivo è rendere i dati più comprensibili agli investigatori. Vengono tipicamente utilizzati **strumenti forensi** e **appropriate tecniche** per l'estrazione di informazioni rilevanti.

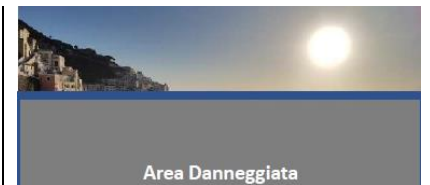
Definizione:

Nella fase di **ispezione** vi è la preparazione e l'estrazione di potenziali prove digitali dai dati raccolti, nella fase precedente.

- RIPRISTINO (RECOVERY DEI DATI):

Nei moderni sistemi operativi si lavora con **puntatori a file**. Quando un file viene eliminato dall'utente, il relativo puntatore viene contrassegnato come **unallocated** (non allocato) o **available** (disponibile). Questo significa che lo spazio, allocato per tale file, è disponibile, pertanto, può essere fisicamente sovrascritto da un nuovo file.

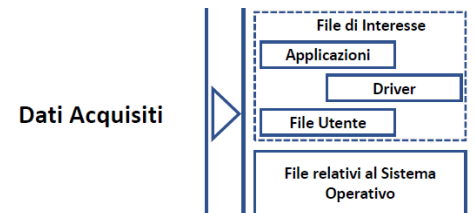
D'altro canto, un file può essere **ripristinato (recovered)**, dal supporto di memorizzazione, se non è stato sovrascritto da altri file.



- RIDUZIONE E FILTRAGGIO DEI DATI ACQUISITI:

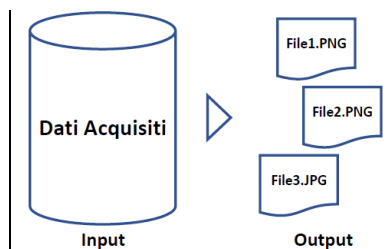
I dispositivi informatici analizzati dagli investigatori possono contenere svariati terabytes di dati. È pertanto impossibile fare una analisi completa su una siffatta mole di dati.

La soluzione è effettuare una fase di **filtraggio dei dati** (tramite strumenti forensi appositi), individuando quelli potenzialmente significativi. Ad esempio, i file relativi al Sistema Operativo risultano di scarso interesse, dal punto di vista forense, pertanto, possono essere parzialmente ignorati.



- CARVING DI FILE E DATI:

I dati raccolti sono solitamente «non strutturati» e/o difficili da interpretare, da parte degli investigatori forensi, ad esempio mancanza di metadati per alcuni file. Capita spesso di avere la necessità di individuare file corrotti, cancellati, frammentati, ad esempio documenti, fogli di calcolo, PDF. Tramite appositi strumenti forensi di data **carving** (letteralmente *intaglio*) è possibile **ripristinare** i suddetti file anche se contenuti in dati «non strutturati».



4. ANALISI:

Definizione:

Nella **fase di analisi**, vengono processate le informazioni con gli obiettivi di determinare i **fatti**, in relazione ad un evento, e di determinare l'importanza e/o la significatività di una prova e il/i soggetto/i responsabile/i.

NOTA: La fase di analisi è un **processo iterativo**, dato che può capitare che la relazione tra i file venga scoperta man mano che si procede con l'analisi dei file stessi.

Ricerche mediante stringhe e keyword risultano utili nella fase di analisi, in quanto semplificano il lavoro dell'investigatore. Ad esempio, in una analisi forense, si cercano informazioni di un individuo, di cui si conosce il nome e cognome o il soprannome. È possibile effettuare ricerche utilizzando proprio il **nome, cognome e soprannome** come parole chiave.

- CENNI DI TECNICHE ANTI-ANALISI FORENSE:

Sono state sviluppate alcune tecniche note che sono deliberatamente attuate al fine di **provare a rendere più difficile l'analisi forense**:

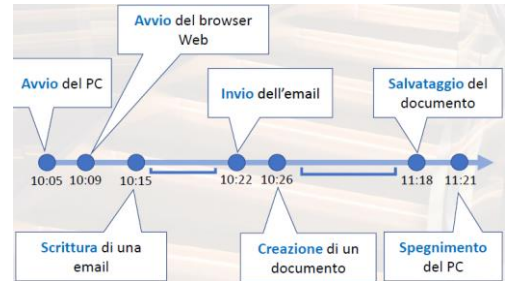
▪ Computer Media Wiping:

Strumenti che fanno wiping (letteralmente *pulizia*) con l'obiettivo di eliminare definitivamente i file. **Remote Wiping:** Utilizzato per la cancellazione remota di file su un dispositivo (ad esempio, un dispositivo rubato).

▪ Cifratura e/o Offuscamento dei Dati:

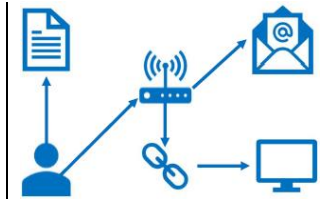
Alcuni malware tendono a offuscare/cifrare file di configurazione, ecc. (ad esempio, i ransomware). È necessario individuare la motivazione relativa alla cifratura di un file (se per questioni di protezione di un file o cifratura effettuata da un malware).

È estremamente utile realizzare delle **timeline** di eventi (*esempio in figura*), basandosi sulle informazioni raccolte (ad esempio, le **timestamp** inerenti a file, processi, ecc.):



- ANALISI DEI COLLEGAMENTI:

L'**analisi dei collegamenti** (*link analysis*) è una potente ed emergente disciplina, nell'ambito della Digital Forensics. L'obiettivo principale è la costruzione di una **presentazione strutturata degli oggetti collegati ed interconnessi**, al fine di comprendere al meglio le associazioni e i collegamenti fra gli oggetti.



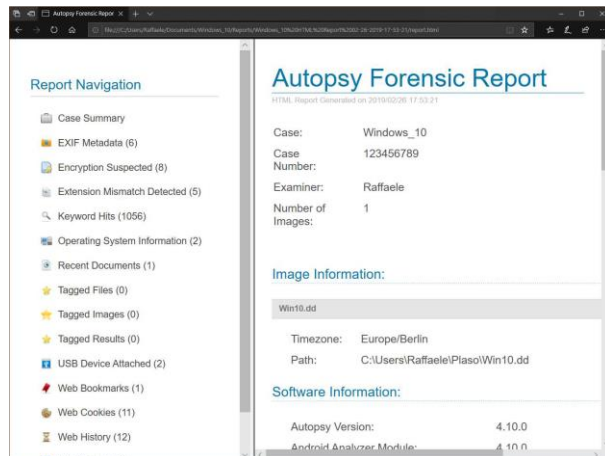
5. PRESENTAZIONE:

La **fase di presentazione** è la fase in cui viene prodotta la **documentazione finale** relativa al risultato dell'investigazione. Tale documentazione deve essere presentata in tribunale o negli uffici preposti.

Definizione:

Nella **fase di presentazione**, l'investigatore condivide, alle parti interessate, i risultati dell'analisi, in forma di report.

Esempio di report generato da un tool di analisi forense (Autopsy):



1BIS. VIRTUALIZZAZIONE

Lo scopo della **virtualizzazione** è quello di **eseguire contemporaneamente** più istanze di sistemi operativi "guest" in un'unica macchina fisica "host". I sistemi operativi "guest" colloquiano con le risorse messe a disposizione dalla macchina fisica "host" attraverso un componente software di livello intermedio generalmente denominata "**hypervisor**" o "**virtual machine monitor**" (VMM).

Uno sviluppatore di software potrà quindi eseguire la sua applicazione in diversi ambienti senza dover disporre di più macchine fisiche (android, apple ...) o un amministratore di sistemi potrà testare uno scenario complesso che veda interagire più servizi su host diversi, ricreandolo su più macchine virtuali (VM) ospitate in una singola macchina fisica.

Ma sono gli utenti finali a trarre i **maggiori benefici**:

- **Aumento dell'affidabilità del sistema:**

Sarà infatti possibile **dedicare** una macchina virtuale all'esecuzione di pochi servizi che notoriamente non vanno in conflitto tra di loro. Inoltre, l'hypervisor è in grado di **isolare** le macchine guest in esecuzione sullo stesso host affinché eventuali problemi che compromettono il funzionamento di una singola macchina virtuale, non influenzino la stabilità delle altre.

- **Consolidamento dei server:**

Molte aziende hanno visto crescere vistosamente il numero dei server proprio a causa dell'aumento dei servizi da fornire ai propri utenti. Attraverso la virtualizzazione si possono eseguire più macchine virtuali nella stessa macchina fisica **riducendo il numero dei server di 10 volte o più**. Infatti, è noto che la maggior parte dei server x86 ha un basso utilizzo di CPU e con le attuali tecnologie multiprocessore multicore non è raro spingersi a rapporti di consolidamento superiori.

- **Riduzione del Total Cost of Ownership (TCO):**

Il consolidamento ad un numero inferiore di server permette una notevole **riduzione dei costi legati all'energia** utilizzata per alimentare i server e per mantenere la temperatura ambientale adatta alle sale server. Inoltre, **si riducono i costi di acquisto** e i canoni di **manutenzione** dei server fisici.

- **Disaster Recovery:**

L'intero sistema operativo "guest" può essere facilmente salvato e ripristinato riducendo notevolmente i tempi di indisponibilità in caso di guasto.

- **Alta disponibilità:**

Se è presente una infrastruttura di server fisici con delle caratteristiche hardware tra loro compatibili e questi server condividono una area dati sulla quale risiedono le macchine virtuali, sarà possibile spostare l'esecuzione di una macchina virtuale su un altro host in caso di failure. Alcuni sistemi prevedono lo spostamento automatico delle macchine virtuali tra gli host in funzione al carico.

- **Esecuzione di applicazioni legacy:**

È frequente che alcune organizzazioni utilizzino applicazioni sviluppate per sistemi operativi che girano su **hardware ormai obsoleto**, non supportato o addirittura introvabile. Attraverso la virtualizzazione si possono continuare ad utilizzare quelle applicazioni che diversamente dovrebbero essere migrate ad una architettura più attuale affrontando i costi relativi al porting e al debug.

- **Sviluppo e Testing:**

Possibilità di predisporre ambienti di sviluppo e di testing di varie tipologie in maniera agevole. Isolamento all'interno degli host rispetto all'ambiente di lavoro principale. Orientato sia ai server che ai client.

In informatica, la virtualizzazione è un termine generico che si riferisce all'**astrazione di risorse di calcolo** (computing resources):

1. **Platform virtualization** ovvero virtualizzazione di piattaforme hardware (concetto di VM);
2. **Resource virtualization** ovvero virtualizzazione di risorse (concetto di qualità del servizio).

HYPERVISOR (VMM) E VIRTUAL MACHINE (VM):

Una **VMM** (Virtual Machine Manager o **Hypervisor**) astrae l'hardware di un singolo calcolatore:

1. Crea e controlla molti diversi **ambienti di esecuzione** (VM);
2. Ciascuno di questi ambienti può avere un proprio sistema operativo;
3. Ciascuno di questi ambienti crede di controllare l'intero sistema hardware.

Quindi una Virtual Machine (VM) è un'ambiente di esecuzione creato da un'hypervisor (VMM).

HYPERVISOR:

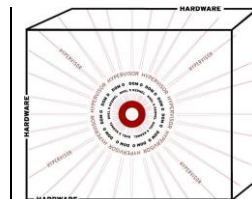
Fornisce protezione, networking, coordinamento dei driver e gestione delle risorse in modo che ogni sistema operativo virtuale si consideri in esecuzione su un server bare-metal.

Consente di creare, controllare, monitorare, distruggere, mettere in pausa o migrare nuove macchine virtuali.

NOTA: un server bare-metal è un computer fisico progettato per eseguire servizi senza interruzioni per periodi prolungati. È altamente stabile, durevole e affidabile, e le risorse fisiche di un singolo server non possono essere condivise tra due o più tenant.

Gli **hypervisor** (VMM) sono attualmente classificati in **3 tipi**:

1. **Hardware-based**, è un software che viene eseguito direttamente su una determinata **piattaforma hardware** (come programma di controllo del sistema operativo). Un sistema operativo "guest" viene quindi eseguito al secondo livello sopra l'hardware. Esempi sono Xen, ESX Server di VMware e Hypervisor di Sun.
2. **OS-like**, è un software che viene eseguito all'interno di un **ambiente del sistema operativo**. Un sistema operativo "guest" viene quindi eseguito al terzo livello sopra l'hardware. Esempi sono server VMware e Microsoft Virtual Server.
3. **Applications**, sono applicazioni eseguite su sistemi operativi standard per fornire funzionalità VMM ai sistemi operativi guest. Example: VMware Workstation and Fusion, Parallels Desktop, Oracle Virtual Box.



PRINCIPALI METODOLOGIE DI VIRTUALIZZAZIONE:

Il problema della virtualizzazione è stato affrontato in diversi modi. Tutte le **4 metodologie di virtualizzazione** attualmente impiegate danno l'illusione di utilizzare un sistema operativo stand alone, non virtualizzato. Esse sono:

1. **Emulation:**

Con l'emulazione l'hypervisor simula l'intero hardware set che permette al sistema operativo guest di essere eseguito senza alcuna modifica. Il software di virtualizzazione si incarica di presentare al sistema operativo guest un'architettura hardware completa a lui nota, indipendentemente dall'architettura hardware presente sulla macchina host.

L'emulatore simula una architettura hardware diversa da quella fisica:



Limiti della Emulation:

Gli emulatori presentano al sistema operativo guest un'**architettura hardware standard** precludendo quelle che potrebbero essere le funzionalità alle quali siamo abituati, ad esempio quelle implementate in hardware. Inoltre, deve **interfacciare la CPU** (con istruzioni semanticamente equivalenti), la **memoria** (accesso esclusivo e riserva) e l'**I/O** tra sistema host e sistema guest. Questo carico di lavoro dell'emulatore rende difficoltoso l'uso di questa tecnica di virtualizzazione quando bisogna emulare sistemi guest che richiedono processori di velocità equivalente al processore dell'host.

2. **Full virtualization:**

La Full (o Native) Virtualization è simile alla Emulation ma i sistemi operativi guest devono essere compatibili con l'architettura hardware della macchina fisica. In questo modo molte istruzioni possono essere eseguite direttamente sull'hardware senza bisogno di un software di traduzione garantendo prestazioni superiori rispetto all'emulazione. Recentemente Intel e AMD hanno introdotto VT-x ed AMD-v. Esempi di software che utilizzano la full virtualization sono VMware, Virtual Box, e Xen, limitatamente ai sistemi operativi proprietari non modificabili.

La Full Virtualization presenta al sistema operativo guest la stessa architettura hardware presente sull'host fisico:

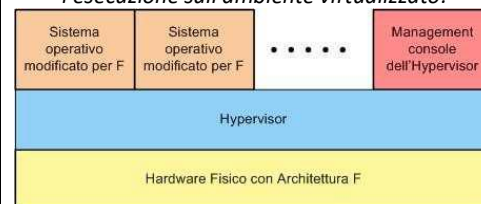


3. **Paravirtualization:**

L'hypervisor presenta alle macchine virtuali una versione modificata dell'hardware sottostante, mantenendone tuttavia la medesima architettura (stessa ABI). Il sistema operativo in esecuzione sulle macchine virtuali è invece modificato per evitare alcune particolari chiamate di sistema.

Questa tecnica permette di ottenere un decadimento delle prestazioni minimo rispetto al sistema operativo non virtualizzato, dato che le istruzioni provenienti dalle macchine virtuali vengono eseguite quasi tutte direttamente sul processore senza che intervengano modifiche.

La Paravirtualization è simile alla full virtualization, ma è necessario modificare il sistema operativo guest per ottimizzare l'esecuzione sull'ambiente virtualizzato:



4. **Operating system level virtualization:**

Non si utilizza un Hypervisor, ma la virtualizzazione è creata utilizzando copie del sistema operativo installato sull'host. I sistemi guest creati saranno a tutti gli effetti istanze del sistema operativo host con un proprio file system, configurazione di rete e applicazioni. Il vantaggio principale di questa tecnica è il miglior utilizzo delle risorse grazie alla condivisione di spazi di memoria. Essendo i sistemi operativi delle macchine guest equivalenti a quello della macchina host, le istanze guest non richiederanno un kernel privato, ma utilizzeranno lo stesso con un conseguente minor utilizzo di memoria fisica. Non adatto a sistemi operativi diversi sullo stesso host. Poca stabilità, poco isolamento.

Esempi: Virtuozzo, Linux VServers, Solaris Containers, HPUX 11i Secure Resource Partitions.

La Operating System level Virtualization mette a disposizione dei sistemi guest l'immagine del sistema operativo in esecuzione sull'host.



2. ACQUISIZIONE DEI DATI

La **fase di raccolta** (o **acquisizione**) del **processo di investigazione**, dovrebbe garantire **4 proprietà**:

1. **Affidabilità**: Non devono esservi dubbi e/o perplessità in merito all'autenticità e sui risultati ottenuti;
2. **Completezza**: Devono essere acquisite tutte le informazioni rilevanti, non solo quelle di una parte del caso;
3. **Accuratezza**: Non devono essere presenti errori nella raccolta dei dati;
4. **Verificabilità**: La metodologia deve essere chiara e **riproducibile**, cioè un altro investigatore dovrebbe essere in grado di arrivare allo stesso risultato, partendo dai medesimi dati.

L'**acquisizione dei dati** (**data acquisition**) è l'attività principale della **fase di raccolta** (o **acquisizione**) del **processo di investigazione**.

I dati rilevanti vengono **acquisiti**, da parte di un investigatore, principalmente da **2 fonti**:

1. **Live Systems**: si considerano anche i live data, ad esempio, il contenuto della memoria RAM;
2. **Dead Systems**: deve essere effettuata una «copia esatta», attraverso passi ben stabiliti e senza errori di acquisizione.

BLOCCARE LE SCRITTURE:

Le **prove originali** devono essere **utilizzate esclusivamente per effettuare delle «copia esatte»**, sulle quali condurre l'analisi forense.

Per evitare che vi siano alterazioni dei dati, durante la creazione di una «copia esatta», è necessario utilizzare un **write blocker** («bloccatore» di scritture), che ha il compito di **evitare che vi siano scritture sui dati** (è possibile esclusivamente effettuare operazioni di lettura). I write blocker possono essere:

▪ Implementati via software [Write Blocker Software]:

Vantaggi:

La scelta di un write blocker software è principalmente economica. Non richiede l'acquisto di particolari dispositivi.

Tuttavia, l'investigatore deve testare costantemente la validità di questa metodologia con la nascita e lo sviluppo dei nuovi standard.

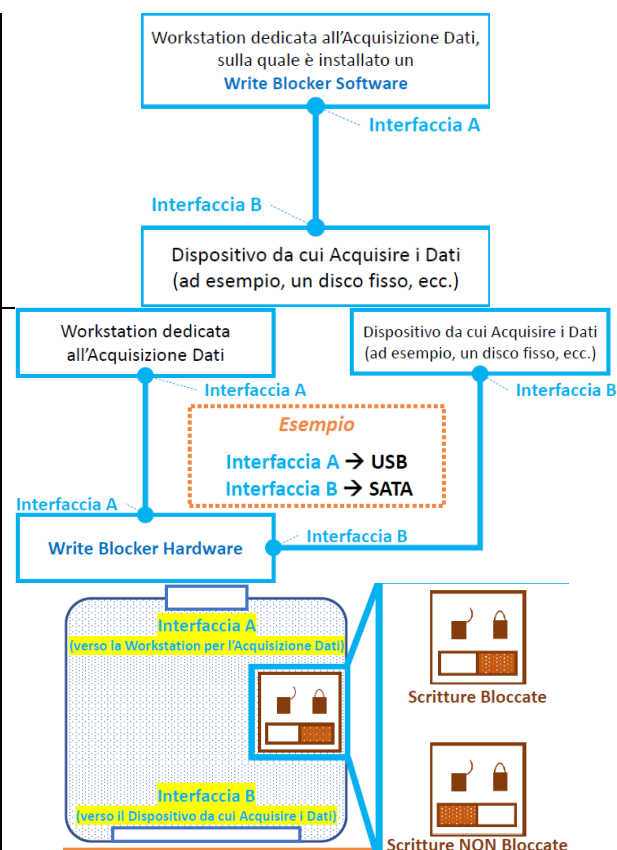
Svantaggi:

Non utilizzabile in alcuni scenari.

▪ Dispositivi hardware dedicati [Write Blocker Hardware]:

Collega la Workstation per l'Acquisizione Dati (**Interfaccia A**) al Dispositivo da cui Acquisire i Dati (**Interfaccia B**) e può bloccarne le scritture.

In alcuni modelli, è possibile specificare, mediante un **interruttore**, se bloccare o meno le scritture.

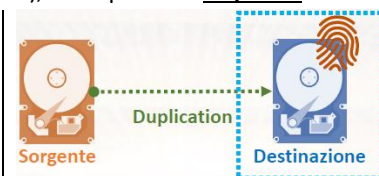


IMMAGINI FORENSI:

Per la creazione di una «copia esatta» di un disco fisso (o un altro tipo di supporto di memorizzazione), sono possibili **2 opzioni**:

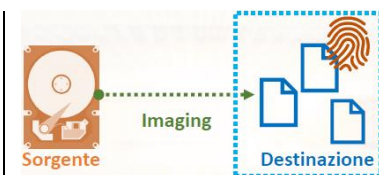
1. Con il **processo di duplication**, la **destinazione** è un **altro disco fisso**.

Il disco fisso destinazione deve essere della **stessa marca, modello e taglia** del disco fisso sorgente. Tutto il disco sorgente (contenuto di tutti i settori) viene replicato (duplicato) nella destinazione. L'obiettivo è quello di ottenere una **copia esatta**, identica in ogni aspetto.



Esistono dei dispositivi hardware che si occupano del processo di duplication e sono chiamati **forensic hardware duplicator**, che integrano la funzionalità di **write blocking** del disco fisso originale e la verifica dell'**esattezza della copia**, ottenuta come risultato. Il vantaggio di tali dispositivi è l'estrema rapidità nell'esecuzione del processo di duplication.

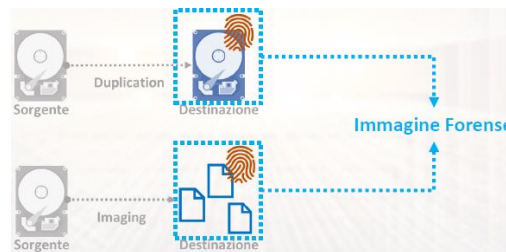
2. La destinazione del processo di **imaging** è un insieme costituito da **uno o più file**, i quali conterranno, al termine del processo, la **copia esatta** dell'intero disco fisso sorgente.



Quando si effettua una **copia logica** (ovvero, la copia «tradizionale») di file e/o cartelle, **non tutti i file potrebbero essere copiati** (a causa di mancanza di permessi, file nascosti, ecc.). Per evitare che vengano persi dei file/directory, durante la copia logica, è necessario effettuare una **copia bit-per-bit** dei dati grezzi («raw») dalla sorgente alla destinazione, senza che vi sia alcuna aggiunta o modifica.

Il risultato del processo di duplication o di imaging è quindi una copia esatta, denotata come **immagine fisica (physical image)** o **immagine forense (forensic image)**.

Al fine di verificare che, il processo di duplication o di imaging, abbia effettivamente restituito una copia esatta, si effettuano dei controlli, mediante l'utilizzo di una o più **funzioni crittografiche di hash**.



Calcolo dell'hash della **sorgente**: H_s
Calcolo dell'hash della **destinazione** (immagine forense): H_p
Se H_s e H_p sono uguali
• **Sorgente** e **destinazione** risultano **effettivamente «identiche»**
Altrimenti
• La **destinazione differisce** dalla **sorgente** (anche di un solo bit)

ALGORITMI DI HASHING E CRITTOGRAFIA:

I valori di hash sono ottenuti da appositi algoritmi. Un valore di hash può essere immaginato come una sorta di «**digital fingerprint**» che è unico e fondamentale nella verifica di integrità delle evidenze. Uno degli algoritmi utilizzati è il **Message Digest (MD5)**, nonostante non sia recente e contenga delle vulnerabilità. MD5 restituisce un valore di hash di 128 bit.

Esempio:

```
Stringa: Ciao, Mondo!
• Valore Hash MD5: 2B0A9B27997C7E4CC82030E26A7D6E14

Stringa: Ciao, Mondp!
• Valore Hash MD5: EF4C64FB6C7F5414CC92D897CDCC9F80
```

Valori di hash **diversi**, su input **diversi**.

Un altro algoritmo utilizzato è **Secure Hashing Algorithm-1 (SHA-1)**. Più sicuro di MD5 e produce un valore di hash di 160 bit. Ad oggi, una delle funzioni più valide e sicure è la funzione SHA-2. Ci sono poi SHA-224, SHA-384 e SHA-512, le quali producono rispettivamente output di dimensione 224, 384 e 512 bit. Da notare che **più la funzione crittografica è robusta, più è difficile che vi siano manomissioni/alterazioni**, pertanto, è possibile accertare che un'immagine forense rimanga inalterata.

Esempio:

```
Stringa: Raffaele Pizzolante
• Valore Hash SHA-1: 0e53662a8d984e47a14efe394186ad86e4782500

Stringa: Raffaele Pizzolante
• Valore Hash SHA-1: 788510614b375327e815b4d6686f41f45271e7a2
```

Valori di hash **diversi**, su input **diversi** (due spazi fra nome e cognome, nel secondo esempio).

IMMAGINI FORENSI (CATEGORIE):

Esistono principalmente **3 categorie** di formati, in merito alle immagini forensi:

- **Formato RAW**: Restituito da tools che operano a basso livello, ad esempio con una **copia bit-per-bit** da un drive a un file.
Vantaggio: Velocità di trasferimento, tolleranza a errori di natura minore e diversi tool sono in grado di leggerli;
Svantaggio: Richiede lo stesso spazio della sorgente, i controlli di validazione vanno conservati a parte (valore hash).
Alcune estensioni di questi file sono: **.dd .raw .img**
- **Formati Proprietari**: In genere, il formato proprietario ingloba l'immagine RAW, ma ne può effettuare la compressione **lossless**.
Suddivisione immagini in più file, detti anche **segmenti** (per memorizzazione su uno o più supporti rimovibili).
Vantaggio: Possono integrare metadati, come Hash dei dati, data di acquisizione e anagrafica di investigazione (numero del caso...).
Svantaggio: Non necessariamente supportati da tutti i tool e limitazioni nella taglia dei file, in cui si suddivide l'immagine.
Il formato **Expert Witness Format (EWF)** è ormai uno standard *de facto*. Usato da diversi software, fra i quali EnCase, Forensics Toolkit (FTK), X-Ways Forensics, ecc.
Permette la produzione di file compressi o non, in base alle preferenze. Estensione dei file Expert Witness Format: **.E01, .E02**.
- **Advanced Forensics Format (AFF)**: L'obiettivo è immagazzinare immagini RAW compresse e non compresse, con nessuna restrizione alla taglia delle immagini, aggiunta di metadati, design semplice ed estensibile, formato Open-Source e per multiple piattaforme, check interni di consistenza e integrità. Alcune estensioni sono: **.AFD** per i segmenti e **.AFM** per i metadati.

Un algoritmo di compressione per dati forensi deve necessariamente utilizzare una strategia **lossless (senza perdita di informazioni)**.

Con gli algoritmi di compressione che usano strategie lossless, è possibile riottenere i dati originali, partendo dal file compresso. Un buon algoritmo potrebbe ridurre la dimensione di una immagine di oltre il 50%. In alcuni casi, un algoritmo di compressione può essere inefficace. Introduce rischi di «perdite» di evidenze, in caso di problemi durante il processo di compressione.

IL TOOL DC3DD:

Il primo tool è **DC3DD**, che è una variante del tool **Data Dump (DD)**, utilizzato per l'acquisizione forense.

Caratteristiche di uno **strumento di Data Dump**:

- **Acquisizione** e **clonazione** di un supporto di memorizzazione, mediante **Bitstream** (bit-per-bit);
- **Copia** delle **partizioni** di un **disco**;
- Copia delle **cartelle** e dei **file**;
- **Check** degli **errori** di un disco fisso;
- **Pulizia forense** di tutti i **dati presenti** su un supporto.

Tipicamente un dispositivo di memorizzazione (storage device), in Linux, è indicato nel modo seguente: **/dev/sda**

- **/dev** fa riferimento al percorso di tutti i device e di drivers, riconosciuti da Linux;
- **/sda** fa riferimento ad un dispositivo di memorizzazione.

sd è relativo a **storage device** (o driver) ed è seguito da una lettera, la quale rappresenta il numero del device di memorizzazione.

Le **partizioni** è una suddivisione logica di una unità di memorizzazione (disco fisso o penna USB), vengono definite per varie motivazioni, come, ad esempio, installazione di più sistemi operativi, ecc.

Esempio:

- **sda1** fa riferimento alla partizione1 sul primo disco (sda);
- **sda2** fa riferimento alla partizione2 sul primo disco (sda);
- **sdb1** fa riferimento alla partizione1 sul secondo disco (sdb);
- **sdb2** fa riferimento alla partizione2 sul secondo disco (sdb).

Per verificare che non vi siano manomissioni, dovrebbe essere calcolato un hash prima, durante e dopo un'acquisizione. In Kali Linux, è possibile utilizzare il comando **md5sum** seguito dal path del dispositivo (ad esempio, un dispositivo che costituisce una prova), per ottenere il valore **hash** MD5 associato a tale dispositivo (è possibile utilizzare **md5sum** anche per i file).

Caratteristiche principali del tool DC3DD:

- **Hashing «on-the-fly»** usando più algoritmi di hash: MD5, SHA-1, SHA-256 e SHA-512;
- Indicazione del progresso ed indicazione del tempo di esecuzione;
- **Scrittura degli errori** individuati su un file di log;
- Suddivisione dei file di output, in più parti;
- **Verifica** dei file;
- **Pulizia forense**.

DC3DD è uno strumento utilizzabile da linea di comando (Command Line Interface).

Esempio di Utilizzo:

```
dc3dd if=/dev/sdb hash=md5 log=dc3ddusb of=test_usb.dd
```

- **if**: specifica il **file di input** (ovvero, il dispositivo di cui si intende effettuare la copia esatta). Nell'esempio, si fa riferimento a **/dev/sdb**.
NOTA: Si tratta di un device secondario (una penna USB, da 8 GB);
- **hash**: specifica l'algoritmo di hash che verrà utilizzato per verificare l'integrità. Nell'esempio, si fa riferimento a MD5.
- **log**: specifica il nome del file di **log**, all'interno del quale verranno riportati tutti i dettagli del dispositivo, del processo di acquisizione ed eventuali errori riscontrati. Nell'esempio, si fa riferimento al file **dc3ddusb**.
- **of**: specifica il **file di output** relativo all'immagine forense creata dal tool (l'estensione può essere .dd, oppure, .img).

PROCESSO DI CLONAZIONE (con DC3DD):

DC3DD permette anche di clonare una immagine forense, acquisita precedentemente, su un nuovo dispositivo. Questo processo è denominato **processo di clonazione**.

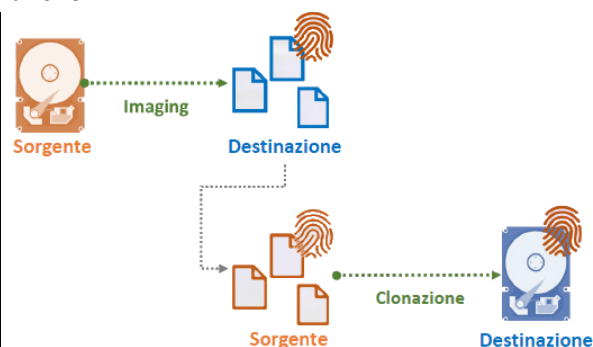
Esempio:

```
dc3dd if=test_usb.dd of=/dev/sdclog=drivecopy.log
```

In questo esempio, l'immagine forense, denotata **test_usb.dd** precedentemente acquisita, viene copiata esattamente (clonata) sul device, identificato dal path **/dev/sdc**.

Esecuzione del processo di duplication tramite i processi di imaging e di clonazione:

Il disco fisso **destinazione**, relativo processo di clonazione, deve avere le stesse caratteristiche (modello, marca e taglia) del disco fisso **sorgente**, relativo al processo di imaging.



PULIZIA FORENSE:

Si supponga che un investigatore abbia utilizzato, nell'ambito di una indagine forense, attualmente conclusa, un certo disco fisso. Tale investigatore **NON può riutilizzare il medesimo disco fisso**, così com'è, per una nuova indagine. Il suddetto disco fisso deve essere preliminarmente preparato, al fine di essere riutilizzato, mediante una fase di preparazione.

La fase di preparazione del disco fisso è necessaria, onde evitare qualsiasi rischio legato al fatto che tracce, relative alla nuova indagine, possano «interfogliarsi» con tracce della precedente indagine conclusa. Questo comporterebbe l'individuazione di potenziali tracce «non corrette» e potrebbe invalidare la nuova indagine.

È consigliabile svolgere la fase di preparazione di un disco fisso, direttamente alla conclusione di una indagine poiché tale fase potrebbe essere onerosa in termini di tempo.

Per la preparazione di un disco fisso, è necessaria una pulizia forense **forensic wiping** detta anche **secure wiping** di tale dispositivo.

La pulizia forense, di un disco fisso, prevede la sovrascrittura del contenuto di ciascun settore (di traccia), con valori nulli (zero) con specifici pattern o con dati random.

Il tool DC3DD fornisce anche la possibilità di effettuare la pulizia forense (opzione **wipe**).

Sono previste **3 principali modalità** per la pulizia forense:

▪ **Modalità 1:**

La pulizia forense viene eseguita sovrascrivendo, con valori zero, il contenuto di ciascun settore del dispositivo specificato.

Comando: `dc3dd wipe=/dev/sdb`

La pulizia forense viene eseguita sul dispositivo, identificato dal path `/dev/sdb` (specificato nell'opzione *wipe*).

▪ **Modalità 2:**

La pulizia forense viene eseguita sovrascrivendo, con un pattern esadecimale (ripetuto), il contenuto di ciascun settore del dispositivo specificato. Il pattern viene specificato dall'utente, mediante l'opzione *pat*.

Comando: `dc3dd wipe=/dev/sdb pat=101010`

La pulizia forense viene eseguita sul dispositivo, identificato dal path `/dev/sdb` (specificato nell'opzione *wipe*), utilizzando, ripetutamente, il pattern esadecimale 101010 (opzione *pat*), per la sovrascrittura dei settori.

▪ **Modalità 3:**

La pulizia forense viene eseguita sovrascrivendo, con una stringa (ripetuta), il contenuto di ciascun settore del dispositivo specificato. La stringa viene specificata dall'utente, mediante l'opzione *tpat*.

Comando: `dc3dd wipe=/dev/sdb tpat=digf`

La pulizia forense viene eseguita sul dispositivo, identificato dal path `/dev/sdb` (specificato nell'opzione *wipe*), utilizzando, ripetutamente, la stringa *digf* (opzione *tpat*), per la sovrascrittura dei settori.

TOOL GUYMAGER:

Un altro tool per l'acquisizione di immagini forensi è **Guymager** (Open-Source), sviluppato da Guy Voncken, presenta **molteplici caratteristiche di DC3DD** ed è disponibile esclusivamente per sistemi operativi Linux-based (preinstallato su Kali Linux), fornisce una interfaccia grafica (GUI).

Guida alle interfacce grafiche ed esempio di utilizzo sulle slide.

UTILIZZO DI DC3DD SU DISPOSITIVI ANDROID-BASED:

È ampiamente diffuso e fornisce l'eventuale integrazione con uno o più store (ad esempio, Google Play Store, ecc).

Guida alle interfacce grafiche ed esempio di utilizzo sulle slide.

3. FILE RECOVERY

Dopo aver recuperato immagini forensi da un supporto di memorizzazione (disco fisso), bisogna effettuare il **file recovery** (ripristino di file) e **data carving** (carving = intaglio).

I SO utilizzano il **file system** ai fini di permettere *accesso*, *memorizzazione* e *modifica* dei dati. Allo stesso modo, i *supporti di memorizzazione* permettono le stesse operazioni, in accordo al file system utilizzato. Con l'aiuto dei **metadati**, il SO è in grado di *identificare la tipologia dei dati*. I metadati contengono informazioni tecniche utili (come data di creazione, nome, dimensione).

Coi metadati è anche possibile migliorare l'indicizzazione e la ricerca, invece senza considerarli (magari non disponibili), ma **sfruttando alcune caratteristiche della struttura dei file**, è possibile comunque effettuare il file recovery e data carving attraverso lo **Slack space** (spazio allentato) e **Unallocated space** (spazio non allocato).

SLACK SPACE:

Il **cluster** è l'unità più piccola che è possibile indirizzare da un file system. Per la memorizzazione di un file **possono servire più cluster**.

Ogni rettangolo rappresenta un cluster (quelli bianchi non sono allocati).

Nei cluster aventi lo stesso colore è memorizzato lo stesso file, esempio:

- File 1 (cluster in **ocra**)
- File 2 (cluster in **verde**)
- File 3 (cluster in **rosso**)

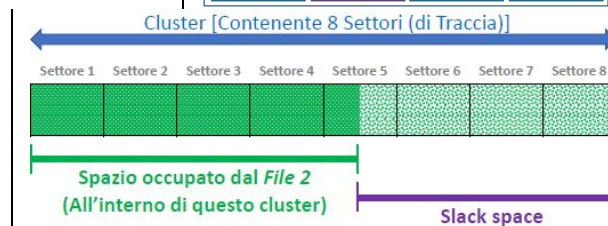
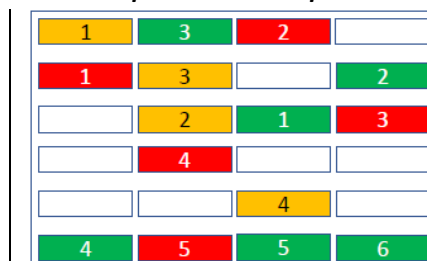
Ad ogni cluster è associato un numero che indica la parte di file memorizzata dal cluster.

OSSERVAZIONE: Il cluster che memorizza l'ultima parte di un file potrebbe essere non completamente utilizzato.

Lo spazio che intercorre dalla fine del file alla fine dell'ultimo cluster è chiamato **slack space**.

Il cluster 6 (in verde), il quale memorizza l'ultima parte del File 2, supponiamo che i dati dell'ultima parte del File 2 occupino solo parzialmente il suddetto cluster →

Lo slack space è importante nelle indagini forensi, poiché, al suo interno possono esservi dati appartenenti a **file cancellati**.



UNALLOCATED SPACE:

Quando un **file viene eliminato** i cluster allocati per esso divengono non allocati (**unallocated space**). Tali cluster sono contrassegnati come liberi (*unallocated*) e **non saranno modificati finché non saranno riallocati** per memorizzare altri file. Tramite i **metadati**, qualora disponibili, è possibile comunque provare a **recuperare file eliminati**. L'unallocated space diviene importante per la digital forensics. All'interno dell'unallocated space, i **metadati** potrebbero non essere presenti o **corrotti**, per via della memorizzazione di altri file (di conseguenza i metadati vengono sovrascritti).

In questi scenari, è comunque possibile **sfruttare alcune caratteristiche** strutturali del contenuto dei file (nello specifico gli **header** e/o i **footer**) al fine di provare a ricostruire i file eliminati. Queste considerazioni sono sfruttate nei processi di data carving e file recovery.

HEADER E FOOTER DI UN FILE:

Ogni file appartiene ad una certa **tipologia** (Word, fogli Excel, filmati AVI) può non essere identificata dall'**estensione** del file stesso (.docx, .xlsx, .avi) siccome, nel contenuto di un file, sono presenti un **header** (sequenza di byte all'inizio del file) e un **footer** (sequenza di byte alla fine del file; NOTA: il footer può essere generalmente omesso) i quali caratterizzano la tipologia di tale file.

PROCESSI DI FILE RECOVERY E DATA CARVING:

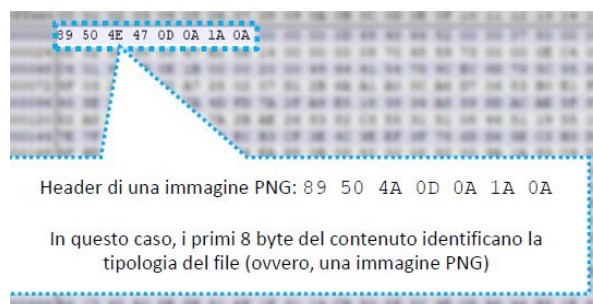
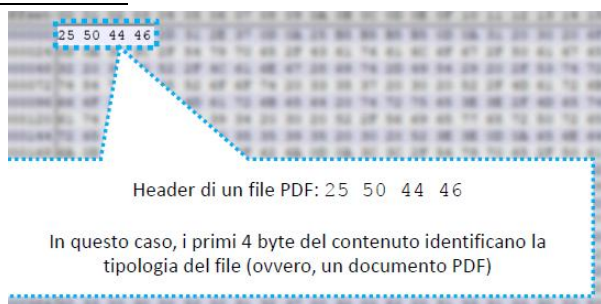
I processi di **file recovery** e **data carving** consistono proprio nell'**individuare** ed **estrarre dati** significativi dallo **slack space** e dall'**unallocated space**. Per cui, anche in caso di modifica dell'estensione del file (da .jpg a .ppp), tramite l'analisi dell'header e/o del footer è possibile provare ad effettuare il recupero.

OSS1: Il processo di recupero è un processo particolarmente oneroso, in termini di tempo di esecuzione.

OSS2: È consigliabile utilizzare strumenti automatizzati, al fine di risparmiare tempo.

OSS3: Può essere significativo, per migliorare l'efficacia del recupero e l'investigazione, l'utilizzo di più di un tool.

Esempi di Header:



TOOL FOREMOST:

Il **tool Foremost** è presente in Kali Linux, a linea di comando, è usato per il recupero dei file, è in grado di leggere l'header e/o il footer dei file al fine di individuarne la relativa tipologia e recuperarli.

Sintassi comando:

foremost -i <file> -o <dir> [options]

- **-i**: Permette di specificare il percorso del file di input. OSSERVAZIONE: il file di input deve essere una immagine forense, precedentemente acquisita, con gli appositi tool (ad esempio, con il tool DC3DD);
- **-o**: Permette di specificare la directory di output, dove verranno memorizzati i file, recuperati mediante il processo di file recovering;
- **[options]**: Eventuali opzioni (facoltative), da specificare solo se necessarie.

Esempio di utilizzo:

È stata utilizzata una immagine forense (11-carvefat.dd) che contiene diversi file (anche eliminati), inoltre, il boot sector della partizione è stato volutamente danneggiato (rende impossibile effettuare un mounting dell'immagine, per visionarne il contenuto).

Per avviare il processo di recovery sull'immagine 11-carve-fat.dd, effettuiamo i seguenti step:

1. Posizioniamoci nella cartella che contiene il file dell'immagine (ad esempio, la Scrivania);
2. Digitiamo il seguente comando: **foremost -i 11-carve-fat.dd -o Ripristinati**
3. L'output del processo verrà riportato nella cartella specificata, ovvero, Ripristinati.

Il processo genererà diversi file nella cartella "Ripristinati":

Gli elementi ripristinati vengono inseriti in apposite sottocartelle, in base alla loro tipologia (ad esempio, le immagini JPEG sono state inserite nella sottocartella jpg).

È possibile notare anche la presenza di un file testuale, denominato audit.txt →

Il contenuto (parziale) del file testuale audit.txt.

Viene riportata una lista dei file ripristinati. Per ciascuno di tali file, viene riportato il nome associato (colonna Name), la dimensione (colonna Size) ed altre informazioni (fra cui eventuali commenti, nella colonna Comment).

Num	Name (bs=512)	Size	File Offset	Comment
0:	00019717.jpg	29 KB	10095104	
1:	00019777.jpg	433 KB	10125824	
2:	00020645.jpg	96 KB	10570240	
3:	00020841.gif	5 KB	10670592	(88 x 31)
4:	00000321.wmv	7 MB	164352	
5:	00021929.wmv	1012 KB	11227648	
6:	00020853.mov	537 KB	10676736	
7:	00016021.wav	311 KB	8202752	
8:	00000281.ole	20 KB	143872	
9:	00016693.ole	24 KB	8546816	
10:	00023957.ole	6 MB	12265984	
11:	00023981.zip	77 KB	12278272	
12:	00016741.pdf	1 MB	8571392	(PDF is Linearized)
13:	00019477.pdf	119 KB	9972224	

14 FILES EXTRACTED

jpg:= 3
gif:= 1
wmv:= 2
mov:= 1
rif:= 1
ole:= 3
zip:= 1
pdf:= 2

Viene poi riportata una sintesi dei file ripristinati, mostrando il numero totale dei file ripristinati ed il numero di file ripristinati, per ciascuna categoria (ad esempio, sono state ripristinate 3 immagine JPEG, denotate dalla voce jpg).

Il tempo di elaborazione può coprire un arco temporale anche molto lungo, in base alla dimensione del file di input ed altri fattori. Se si conosce la tipologia di file che si intende ripristinare, è possibile utilizzare l'opzione -t, in modo da ridurre le tempistiche elaborative.

TOOL SCALPEL:

Il **tool Scalpel** (scalpello) è presente in Kali Linux, tramite linea di comando, basato su Foremost, tuttavia, **significativamente più efficiente** di quest'ultimo. Scalpel risolve i problemi di Foremost, relativi all'utilizzo elevato di CPU e RAM durante la fase di recovering.

A differenza di Foremost, con Scalpel è necessario specificare le tipologie di file che si intende cercare di ripristinare. Scalpel deve essere configurato mediante il relativo file di configurazione, denominato scalpel.conf, individuabile nella directory /etc/scalpel.

Contenuto (parziale) del file di configurazione scalpel.conf:

- 1° colonna: Estensione, associata alla tipologia;
- 4° colonna: Header rappresentato in esadecimale;
- 5° colonna: Footer rappresentato in esadecimale.

```
# GIF and JPG files (very common)
#   gif      y      5000000      \x47\x49\x46\x38\x37\x61      \x00\x3b
#   gif      y      5000000      \x47\x49\x46\x38\x39\x61      \x00\x3b
#   jpg      y      5242880      \xff\xd8\xff??Exif          \xff\xd9
```

Sintassi comando:

scalpel -o <dir> <file>

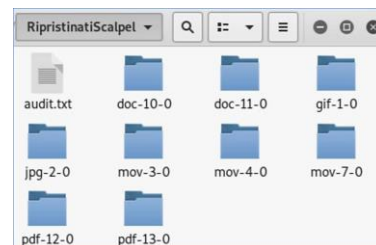
- **-o**: Permette di specificare la directory di output, dove verranno memorizzati i file, recuperati mediante il processo di file recovering;
- **<file>**: Permette di specificare il percorso del file di input.

Per avviare il processo di recovery sull'immagine forense 11-carve-fat.dd (utilizzata con Foremost), effettuiamo i seguenti step:

1. Posizioniamoci nella cartella che contiene il file dell'immagine (ad esempio, la Scrivania)
2. Digitiamo il seguente comando: **scalpel -o RipristinatiScalpel/ 11-carve-fat.dd**
3. L'output del processo verrà riportato nella cartella specificata, ovvero, RipristinatiScalpel.

Il processo genererà diversi file nella cartella "RipristinatiScalpel".

L'output è molto simile a quello di Foremost, inoltre, è possibile notare la presenza del file audit.txt.



Nella cartella RipristinatiScalpel/jpg-2-0, Scalpel ha individuato 5 file JPEG (in base a header e/o footer), tuttavia, solo tre mostrano un'anteprima, mentre gli altri due no perché sono corrotti, Scalpel ha verosimilmente ripristinato dei «FALSI POSITIVI».

Inoltre, è possibile osservare che gli ultimi due file sono identici (stesso valore di hash). In questo caso ha ripristinato un numero minore di immagini JPEG (ovvero 2), rispetto a Foremost (ne aveva ripristinate 3). Risulta ancor più evidente l'importanza di considerare più tool, i quali potrebbero NON restituire il medesimo risultato sulla medesima immagine forense.

A differenza del file audit.txt prodotto da Foremost, nel file audit.txt prodotto da Scalpel non viene riportata alcuna sintesi in relazione al numero di file estratti per ciascuna tipologia.

The following files were carved:				
File	Start	Chop	Length	Extracted From
00000010.doc	143872	NO	8402944	11-carve-fat.dd
00000002.jpg	10095104	NO	29885	11-carve-fat.dd
00000001.jpg	8522240	NO	24367	11-carve-fat.dd
00000013.doc	143872	YES	10000000	11-carve-fat.dd
00000016.pdf	8571392	NO	1399508	11-carve-fat.dd
00000017.pdf	8571392	NO	1523266	11-carve-fat.dd
00000018.pdf	9972224	NO	122434	11-carve-fat.dd
00000011.doc	8546616	NO	3719168	11-carve-fat.dd
*****	*****	*****	*****	*****

TOOL PHOTOREC:

Il **tool PhotoRec** (*Photo Recovery*) è un software di file recovery da immagini forensi. Capacità del software di **recuperare fotografie dalla memoria di macchine fotografiche**, è preinstallato su Kali Linux e funziona anche nel caso di **supporti danneggiati o formattati**.

Sintassi comando:

photorec [input]

- **[input]**: Parametro opzionale che permette di specificare il percorso del file di input.

OSS1: il file di input può essere una immagine forense (formato .DD oppure .E01) oppure un device;

OSS2: Se non viene specificato il parametro [input], il tool richiede all'utente di selezionare un device (fra quelli disponibili).

Per avviare il processo di recovery sull'immagine 11-carve-fat.dd (utilizzata con Foremost e Scalpel), effettuiamo i seguenti step:

1. Posizioniamoci nella cartella che contiene il file dell'immagine (ad esempio, la Scrivania)
2. Digitiamo il seguente comando: **photorec 11-carve-fat.dd**
3. L'output del processo verrà riportato nella cartella recup_dir.

OSSERVAZIONE: Il nome della cartella di output (recup_dir), non può essere modificato.

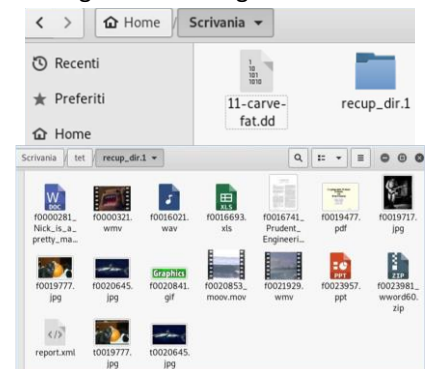
Una volta fatto ciò seguire la GUI selezionando l'unità di supporto e quali file si vuole recuperare scegliendo la categoria.

Al termine del processo sarà possibile accedere alla cartella recup_dir.1.

NOTA: PhotoRec aggiunge automaticamente il suffisso .N al nome della cartella recup_dir (un recupero successivo darà luogo ad una cartella recup_dir.2 e così via).

I file f0019777.jpg e t0019777.jpg, fanno riferimento alla stessa immagine, ma hanno risoluzioni diverse ed una di esse ha una risoluzione particolarmente bassa (potrebbe trattarsi di un ripristino parziale). I file f0020645.jpg e t0020645.jpg, fanno riferimento alla stessa immagine. Alcuni tool di file recovery e data carving, potrebbero recuperare (o tentare di recuperare) solo parzialmente un file.

Il contenuto della cartella recup_dir.1. PhotoRec ha memorizzato anche un report, in formato XML, nel file report.xml, contenente dettagli sui file e sulla fase di recupero.



```
</configuration>
<fileobject>
  <filename>f0000281.doc</filename>
  <filesize>19968</filesize>
  <byte runs>
    <byte_run offset='0' img_offset='143872' len='20480' />
  </byte runs>
</fileobject>
<fileobject>
  <filename>f0000321.wmv</filename>
  <filesize>8037267</filesize>
  <byte runs>
    <byte_run offset='0' img_offset='164352' len='8038400' />
  </byte runs>
</fileobject>
```

Contenuto (parziale) del file report.xml:

Nella prima parte del report, vengono indicate informazioni sulla versione di PhotoRec, sull'immagine sorgente, sulla versione del Sistema Operativo sui cui si è svolta la fase di recupero, ecc.

Nella seconda parte del report, per ciascun file recuperato, vengono diverse informazioni, fra cui: Nome assegnato da PhotoRec / Dimensione del file (espressa in byte) / Ecc.

TOOL BULK EXTRACTOR:

I tool Foremost, Scalpel e PhotoRec sono abbastanza potenti negli ambiti del file recovery e del data carving.

In alcuni scenari, potrebbe essere utile **recuperare dati significativi** (email, numeri di telefono, ecc.), piuttosto che recuperare interi file eliminati (che potrebbero anche non essere recuperati a causa di alterazioni del contenuto).

Mediante **Bulk Extractor**, possono essere individuati ed estratti diverse tipologie di dati, fra cui numero di carte di credito (Credit Card Number – CCN) / Indirizzi email / URL/ ecc...

Sintassi comando:

bulk_extractor -o <dir> <file>

- **-o**: Permette di specificare la directory di output, dove verranno memorizzati i file, contenenti i dati estratti, tramite il processo di data carving, operato da Bulk Extractor;
- **<file>**: Permette di specificare il percorso del file di input.

Per avviare il processo di recovery sull'immagine terry-work-usb-2009-12-11.E01, effettuiamo i seguenti step:

1. Posizioniamoci nella cartella che contiene il file dell'immagine;
2. Digitiamo il seguente comando: **bulk_extractor -o bulk_output terry-work-usb-2009-12-11.E01**
3. L'output del processo verrà riportato nella cartella specificata, ovvero, bulk_output.

Al termine del processo, sarà possibile visionare i file, generati da Bulk Extractor, i quali sono contenuti nella cartella "bulk_ouput".

NOTA: Viene generato anche un report, in formato XML, relativo al processo di estrazione, denominato report.xml.

4. ACQUISIZIONE DELLA MEMORIA

Abbiamo detto che quando ci si occupa di acquisizione di dati, da un disco fisso, esistono due tipi di analisi:

- **Post Mortem Analysis:** Analisi di un dead system (sistema spento) e il contenuto della memoria RAM non è presente;
- **Live Analysis:** Analisi di un sistema acceso (memory analysis della RAM) bisogna minimizzare l'impatto sul sistema.

Alcuni passi della live analysis potrebbero **risultare irripetibili**, per via delle seguenti problematiche:

- **Problematiche di carattere tecnico:**

Non è possibile effettuare l'eventuale analisi di alcuni dati, senza alterare lo stato del sistema (contenuto della memoria RAM, ecc.).

OSSERVAZIONE: La fase di acquisizione di alcuni dati può provocare l'alterazione dello stato della macchina (acquisizione RAM);

- **Problematiche di carattere temporale:**

Lo stato della macchina, all'atto dell'attività, è frutto del «momento», difficile (se non impossibile) riprodurlo successivamente.

Non tutti i dati hanno la stessa volatilità, è indispensabile definire un **ordine di volatilità** (OOV). La memoria RAM e i dati in essa contenuti hanno elevata priorità nell'OOV. Risulta quindi importante acquisire il contenuto della memoria (**memory acquisition**).

OSS: Ci focalizzeremo su due aspetti principali: acquisizione (memory acquisition) e analisi della memoria RAM (memory analysis).

MEMORY DUMP:

Un **memory dump** (o memory image) è una «istantanea» del contenuto della memoria RAM.

Il processo di acquisizione di un memory dump è detto **memory acquisition** (o memory imaging) e può essere effettuato da tool esterni. Tuttavia, il S.O. produce automaticamente un memory dump. Ad esempio, al verificarsi di un problema all'interno del sistema, il quale deve essere riavviato poiché non può più assicurare un comportamento corretto.

Esistono diverse **tipologie** di memory dump, fra cui:

- **Memory dump in formato RAW:**

Effettua una copia «esatta» del contenuto della memoria RAM. Analogia con le immagini forensi, generate dal tool come D3CDD, ecc. La nota negativa è che non contiene lo stato del processore (contenuto dei registri).

- **Memory dump prodotti automaticamente dal S.O. [sistemi Windows-based]:**

- **Memory dump prodotto in seguito ad un grave crash di sistema:**

Nei sistemi Windows-based, al verificarsi di un grave crash del S.O. (necessario il riavvio), viene generata una BSoD (Blue Screen of Death). Nei recenti sistemi, dopo aver mostrato all'utente la BSoD, il S.O. produce automaticamente un memory dump. È possibile utilizzarlo per individuare la causa che ha condotto al crash del sistema, ad esempio un problema con un applicativo o driver.

Questo meccanismo include tutte le pagine gestite dal Windows Memory Manager (sistema di gestione della memoria di Windows). Include aspetti relativi all'utente ed al kernel. La nota negativa è che non è disponibile su sistemi a 32 bit e non acquisisce alcuni dati iniziali (ad esempio, password –cifrate– dell'autenticazione, ecc.).

- **Memory dump prodotto dal processo di ibernazione (dove il processo di ibernazione è avviato dall'utente):**

Grazie al processo di ibernazione (o sospensione), è possibile spegnere il sistema, mantenendo il suo «stato» (contenuto della memoria RAM, del disco fisso, ecc.). Alla riattivazione del sistema, verrà ripristinato lo stato del sistema al momento immediatamente precedente dell'avvio dell'ibernazione. Durante il processo di ibernazione viene acquisito anche un memory dump (utilizzato per ripristinare il contenuto della memoria RAM, alla riattivazione del sistema).

Questo meccanismo memorizza all'interno del file **hiberfil.sys** il contenuto dei registri e memory dump. La nota negativa è che è disponibile solo per alcune versioni di Windows e viene eseguita esclusivamente se l'utente richiede l'ibernazione del sistema.

Vi sono molti **tool esterni** che permettono la memory acquisition. Un tool esterno deve essere eseguito direttamente sul live system.

OSS: L'esecuzione stessa di tale tool impatta sul contenuto della memoria RAM del live system.

Alcuni tool esterni (per sistemi Windows-based) sono **FTK®** (Forensic Toolkit®) **Imager** è un software che permette di creare immagini forensi (da dischi fissi, CD/DVD, ecc.) e memory dump di differenti supporti, l'altro tool è **Dumplt** che è capace di acquisire memory dump, in formato RAW, e di effettuare la comparazione tramite una funzione di hash (che deve essere specificata da linea di comando).

VOLATILITY FRAMEWORK (MEMORI ANALYSIS):

All'interno della RAM (e del file di paging) è possibile individuare diversi dati rilevanti, ad esempio password (cifrate), possibili informazioni dell'utente, processi in esecuzione e processi nascosti, ecc.

Considerati i suddetti aspetti, è estremamente significativo effettuare una **investigazione forense sulla memoria** (**memory forensics**).

OSS: L'analisi della memoria viene condotta su un memory dump (non direttamente sulla memoria).

Il **Volatility Framework** permette l'analisi della memoria ed è preinstallato su Kali Linux. Esso permette di estrarre specifiche informazioni dal memory dump su cui si sta conducendo l'analisi, tramite l'utilizzo dei **plugin**.

Ad esempio, è possibile estrarre informazioni sui processi, sulle attività di rete, ecc.

Volatility è in grado di analizzare diversi formati di memory dump, fra i quali Memory dump automaticamente prodotti da sistemi Windows-based (relativi a crash/ibernazioni) o acquisiti da Virtual Machine, Formati RAW e anche i memory dump acquisiti dai tool.

Sintassi comando:

volatility -f <file> [plugin] [options]

- **-f:** Permette di specificare il percorso del file di input;

OSS: il file di input deve essere un memory dump, in uno dei formati supportati.

- **plugin:** Permette di specificare il nome del plugin, da utilizzare (parametro opzionale);
- **options:** Permette di specificare eventuali opzioni (se necessarie).

Memory Acquisition



L'organizzazione logica della memoria RAM dipende dalla **tipologia** del S.O., esempio un sistema Windows-based avrà una diversa organizzazione logica della memoria RAM rispetto a un sistema Linux-based. È importante considerare anche la **versione** specifica del S.O., esempio Windows 10 avrà una diversa organizzazione logica della memoria RAM rispetto a Windows 8. L'organizzazione logica può variare anche in base all'**architettura della CPU** per il quale il S.O. è progettato, esempio l'organizzazione logica della memoria RAM definita da un S.O. per CPU con architettura a 32 bit sarà diversa rispetto a quella con architettura a 64 bit.

Al fine di analizzare in maniera più accurata un memory dump è necessario conoscere il profilo del S.O. (tipologia, versione, ecc.) da cui è stata effettuata l'acquisizione. Tramite il profilo si risalire all'organizzazione logica della memoria RAM (e memory dump).

NOTA: Non è detto che l'investigatore conosca il profilo del S.O. relativo al memory dump acquisito magari perché il memory dump potrebbe essere stato acquisito da un'altra persona.

OSS: Se non si conosce il profilo del S.O. Volatility è in grado di suggerirci, analizzando il memory dump, il profilo più adeguato mediante il plugin *imageinfo*, col comando da terminale:

volatility --info

Sintassi comando:

volatility -f <file> cridex.vmem imageinfo

cridex.vmem è un memory dump di esempio, mentre *imageinfo* è un plugin per avere indicazioni sui profili suggeriti da Volatility.

Possiamo suddividere i plugin del Volatility Framework in diverse categorie:

■ **Analisi dei Processi:**

Volatility Framework permette di elencare i processi ed ottenere diverse informazioni su di essi, ad esempio data e ora in cui il processo è stato avviato, ecc. Per ottenere tali informazioni è possibile utilizzare i seguenti plugin:

- **pslist:** **volatility --profile=WinXPSP3x86 -f cridex.vmem pslist**
Viene mostrata la lista dei processi (sia avviati direttamente dal sistema sia avviati dall'utente). Per ciascuno dei processi vengono mostrate: PID (Process ID) riferito al processo e PPID (Parent Process ID) riferito al processo padre.
- **pstree:** **volatility --profile=WinXPSP3x86 -f cridex.vmem pstree**
Mostra la lista dei processi, dove la lista è ad «albero», infatti, vengono indentati i processi figli, in modo che risulti più immediato distinguere i processi padri ed i processi figli.
- **psscan:** **volatility --profile=WinXPSP3x86 -f cridex.vmem psscan**
Mostra la lista dei processi, includendo eventuali processi nascosti che potrebbero essere indice della presenza di malware che cercano di nascondersi sia all'utente sia ai software anti-malware, al fine di effettuare azioni malevole.
- **psxview:** **volatility --profile=WinXPSP3x86 -f cridex.vmem psxview**
Individua eventuali processi nascosti (riconducibili a malware). Permette di effettuare una comparazione incrociata, considerando l'output di diversi plugin (come *pslist* e *psscan*) atti ad elencare i processi in memoria.

■ **Identificazione di Attività di Rete:**

Servizi e connessioni di rete utilizzati dai processi possono fornire informazioni rilevanti, esempio indirizzi IP locali e remoti, le porte utilizzate, ecc. I seguenti plugin sono utilizzati per carpire informazioni su servizi e/o connessioni di rete:

- **connections:** **volatility --profile=WinXPSP3x86 -f cridex.vmem connections**
Mostra le connessioni attive, riportando per ciascuna di esse: PID (Process ID) del processo a cui è riferita la connessione, IP locale (con la relativa porta della connessione) e IP remoto (con la relativa porta della connessione).
- **connscan:** **volatility --profile=WinXPSP3x86 -f cridex.vmem connscan**
Mostra le connessioni attive e terminate, riportando per ciascuna di esse: PID (Process ID) del processo a cui è riferita la connessione, IP locale (con la relativa porta della connessione) e IP remoto (con la relativa porta della connessione).
- **sockets:** **volatility --profile=WinXPSP3x86 -f cridex.vmem sockets**
Mostra informazioni aggiuntive sulle connessioni, in cui sono presenti socket in «ascolto». Protocolli supportati: TCP e UDP.
- **netscan:** **volatility --profile=<Profilo> -f <MemoryDump> netscan**
Per memory dump, acquisiti da sistemi Windows-based, è possibile utilizzare il plugin netscan che permette di individuare tracce di attività di rete, relative a connessioni basate su protocolli TCP e UDP. Per ciascuna attività vengono riportati: Indirizzo IP locale (e relativa porta) ed indirizzo IP remoto (e porta), Data/ora relativa al momento in cui la connessione è stata Avviata, Ecc.

■ **Analisi di librerie DLL (Dynamic Link Library) di Windows:**

Una **libreria DLL** (Dynamic Link Library) è costituita da porzioni di codice e dati. Più processi possono utilizzare simultaneamente la stessa libreria. Grazie ad esse è possibile progettare un programma e suddividerlo in moduli (dove ogni modulo è una libreria DLL). Un processo può caricare dinamicamente la libreria DLL (se installata). Si ha una riduzione dei tempi di caricamento, poiché i moduli vengono caricati solo in caso vi sia necessità. L'identificazione dei processi che caricano delle librerie DLL e la versione delle librerie stesse può essere di aiuto nella correlazione di processi, infatti, è possibile mettere in relazione account multipli di un utente.

- **verinfo:** **volatility --profile=WinXPSP3x86 -f cridex.vmem verinfo**
Mostra informazioni dettagliate, in riferimento alle librerie DLL utilizzate da processi generati da Portable Executable (PE) di Windows. Un eseguibile PE è strutturato in modo tale che possa contenere tutto il necessario per l'esecuzione in Windows.
- **dlllist:** **volatility --profile=WinXPSP3x86 -f cridex.vmem dlllist**
Per ciascun processo, vengono mostrate tutte le librerie DLL, utilizzate da tale processo.

■ **Informazioni sul Registro di Sistema del S.O. Windows:**

Il registro è una componente dei sistemi Windows-based e memorizza: Impostazioni/Preferenze del S.O. stesso, di alcuni programmi, di driver, Ecc. Vengono inoltre memorizzate preferenze dell'utente. Dal punto di vista forense, il registro è una risorsa notevole, poiché è una sorta di database, contenente tantissimi valori dai quali estrarre informazioni.

Il registro ha una **struttura gerarchica** ad albero, è suddiviso in cinque chiavi radice all'interno delle quali sono presenti sotto-chiavi e valori. Ogni chiave radice corrisponde ad uno o più file, nel file system, e vengono chiamati **hive** (alveari). Poiché il registro è potenzialmente acceduto in maniera frequente dal S.O., alcune informazioni sugli hive e del registro sono mantenute in memoria.

- **hivelist: volatility** --profile=WinXPSP3x86 -f cridex.vmem hivelist

È possibile ottenere informazioni sugli hive individuati all'interno del memory dump. Viene indicato il percorso degli hive all'interno del file system. In questo modo sarà possibile localizzare gli hive ed analizzarli successivamente.

■ **Altri plugin:**

- **filesan: volatility** --profile=WinXPSP3x86 -f cridex.vmem filesan

Un processo che intende creare, scrivere o leggere un certo file, deve effettuarne preliminarmente l'apertura. Windows mantiene, in memoria, i «riferimenti» di tutti i file aperti. Tramite il plugin filesan, vengono cercati, all'interno del memory dump, tutti i suddetti riferimenti. Il plugin filesan elencherà quindi tutti i **file aperti ed eventuali file non visibili**, da alcuni tool standard (i file non visibili potrebbero potenzialmente essere nascosti da un malware). Per ciascun file, verrà riportato il percorso completo ed i permessi effettivamente garantiti al file.

- **timeliner: volatility** --profile=WinXPSP3x86 -f cridex.vmem timeliner

Il plugin timeliner risulta particolarmente utile agli investigatori, poiché permette di ottenere una timeline degli eventi individuati nel memory dump. Gli eventi vengono raggruppati in base alla data e all'orario. Alcuni esempi di eventi: Avvio di un processo, Utilizzo di una libreria DDL, Utilizzo del registro di Windows, ecc.

- **malfind: volatility** --profile=WinXPSP3x86 -f cridex.vmem malfind

Il plugin malfind aiuta gli investigatori nell'individuazione di eventuali malware, riportando codice potenzialmente malevolo iniettato nella memoria. Si basa su alcune caratteristiche proprie dei malware, per individuare codice potenzialmente malevolo. È importante identificare eventuali malware, poiché potrebbero aver svolto operazioni malevole, all'insaputa dell'utente. Pertanto, l'investigatore dovrebbe essere in grado di individuare quali sono tali operazioni, onde evitare di attribuirle all'utente.
NOTA: L'output di Volatility, con il plugin malfind, può essere particolarmente esteso.

5. ANALISI E SUPER TIMELINE

RECAP Definizione:

Nella **fase di analisi**, vengono processate le informazioni con gli obiettivi di determinare i **fatti**, in relazione ad un evento, e di determinare l'importanza e/o la significatività di una prova e i soggetti responsabili.



TOOL AUTOPSY:

Il **tool Autopsy** permette di effettuare l'analisi di dischi fissi o immagini forensi, supporta diversi formati (incluso RAW, EWF e AFF).

Autopsy è una sorta di GUI, per i tool forniti dalla suite TSK. In Kali Linux, il tool Autopsy permette di svolgere diverse attività:

- **Analisi di immagini forensi:** Permette l'analisi di una immagine forense mostrandone informazioni su file e/o directory;
- **Timeline delle attività sui file:** Permette la realizzazione di una timeline, in base ai timestamp dei file (data di creazione/accesso);
- **Verifica dell'integrità di immagini forensi:** Calcola l'hash MD5 di immagini forensi o di file specifici;
- **Ricerca mediante keyword:** Permette di ricercare informazioni mediante delle keyword o espressioni regolari;
- **Analisi dei file e di metadati:** Permette di visualizzare i dettagli relativi ai metadati e permette l'analisi di specifici file.

Autopsy verifica anche se ci sono eventuali file con **incoerenze** tra l'estensione e la tipologia effettiva. Esempio, si consideri un file avente estensione .doc ma la tipologia effettiva è una immagine JPEG (in accordo all'header di tale file) vi è una incoerenza.

Tali file sono rilevanti per l'indagine forense, poiché l'estensione potrebbe essere stata alterata maliziosamente.

LE SUPER TIMELINE:

Un **timestamp** (marca temporale) registra il momento temporale in cui un evento avviene. Le evidenze dispongono di un timestamp:

- **Timestamp di un File:** È possibile reperire dal file system la data e l'ora di creazione, dell'ultima modifica, ecc...
- **Timestamp di un Processo:** Nei live system, dal memory dump è possibile ottenere la data e l'ora di avvio/terminazione.

Esistono diversi formati di timestamp, provenienti dal file system, relativi ai file:

- Nei sistemi Linux based un timestamp è memorizzato in formato Posix o Epoch (32 bit);
- Nei sistemi Windows based, un timestamp è rappresentato con 64 bit nel formato FILETIME, per ciascun file vengono memorizzati diversi timestamp che riportano le cosiddette informazioni **MACB**.

In alcuni casi, vengono riportate esclusivamente le cosiddette informazioni MAC.

Lettera	Descrizione
M	Data e ora dell'ultima modifica
A	Data e ora dell'ultimo accesso al file
C	Data e ora dell'ultima modifica ai metadati
B	Data e ora di creazione del file

I timestamp sono memorizzati nei metadati di ciascun file. Nel file system NTFS, i timestamp sono memorizzati all'interno della **Master File Table (MFT)**. Tutti i file e gli oggetti memorizzati dal file system sono descritti all'interno della MFT. Ciascuna entry (**record**) della MFT contiene la descrizione di un file ed il puntatore ai dati. Per file di piccole dimensioni la entry conterrà direttamente il suo contenuto.

Rappresentazione parziale di un **record della MFT**:

Standard Information	Nome del File	Security Descriptor	Data
----------------------	---------------	---------------------	------

I timestamp vengono memorizzati all'interno del campo **Standard Information**.

Utilizzando i timestamp è possibile realizzare una **timeline** col quale gli investigatori possono analizzare l'**andamento temporale degli eventi**. Inoltre, è possibile **individuare eventi temporali vicini** ed effettuare delle ipotesi sulla loro eventuale correlazione.

Esempio:

- Creazione di un file → Modifica di una presentazione
- Apertura di un programma → Creazione di un documento
- Modifica di una immagine → Modifica di un file di testo

Le timeline tradizionali sono essenzialmente basate sui timestamp specificati dal file system.

Esempio di una semplice timeline, basata su timestamp del file system

Timestamp	Operazione	Nome del File	Note
03/03/2019 10:00	Ultimo Accesso	C:\DigFor.txt	Apertura di un file di testo
03/03/2019 10:30	Creazione	C:\DF_TEST.doc	Creazione di un documento
03/03/2019 10:45	Modifica del Contenuto	C:\DF_TEST.doc	Salvataggio del documento

PROBLEMI DELLE TIMELINE TRADIZIONALI:

Uno dei principali **problemi della timeline**, basata su timestamp del file system, è che i timestamp **possono essere modificati da:**

- **Utenti**, anche in maniera legittima e/o involontaria (ad esempio, apertura/modifica erranea di un file);
- **Comportamento (legittimo) del Sistema Operativo e/o di software autorizzati**, la scansione di un file da parte di un antivirus farà sì che il suo timestamp venga alterato (**data e ora dell'ultimo accesso** corrispondente alla **data e ora di scansione** dell'antivirus). Oppure, un possibile accesso da parte del S.O. a file per via di una ricerca effettuata dall'utente.

Alcuni S.O., per ragioni legate alle performance, potrebbero NON effettuare sempre l'aggiornamento della data e dell'ora relativa all'ultimo accesso di un file. In Windows e in Linux è possibile disabilitare l'aggiornamento della data ed ora dell'ultimo accesso di un file. Basarsi esclusivamente sui timestamp del file system non indica il contesto, ad esempio un file potrebbe essere ripetutamente acceduto dal sistema operativo per svariate ragioni (come file di log).

OSS: È possibile notare come i timestamp del file system abbiano una natura potenzialmente poco attendibile. Ciò potrebbe portare gli investigatori a realizzare una **timeline non corretta o parzialmente alterata**.

TECNICHE ANTI-FORENSE PER L'ALTERAZIONE DEI TIMESTAMP:

Il **valore del timestamp** (del file system) può essere **alterato maliziosamente** tramite appositi tool dove è possibile **modificare il campo Standard Information** di un record della MFT (File System NTFS). Esistono altre tecniche anti-forensi in grado di alterare i timestamp.

OSS: A causa delle tecniche anti-forensi è importante che gli investigatori abbiano **accesso a informazioni da più punti**, per validare/confutare il risultato ottenuto.

Una soluzione è quella di **estendere la timeline** arricchendola con **informazioni provenienti da più fonti**, ottenendo un quadro più chiaro e **minimizzare l'impatto di tecniche anti-forensi**. Infatti, è più difficile alterare tutte le informazioni ottenute da diverse fonti. La super timeline è una estensione della timeline, che prevede l'inclusione di diverse informazioni provenienti da diverse fonti, fra cui:

- File di log (del sistema operativo, ecc.);
- Metadati del file system;
- Registro di sistema (sistemi Windows-based);
- Ecc.

IL FRAMEWORK PLASO:

Il framework **Plaso** è un tool per la realizzazione di super timeline e presenta un'architettura suddivisa in quattro componenti:

▪ Preprocessing:

Svolto preliminarmente da Plaso e si occupa di reperire alcune informazioni: Versione S.O., fuso orario (timezone), nome della macchina (hostname), applicazioni di default (browser di default, ecc.), utenti e l'eventuale path associato ad essi;

▪ Collection:

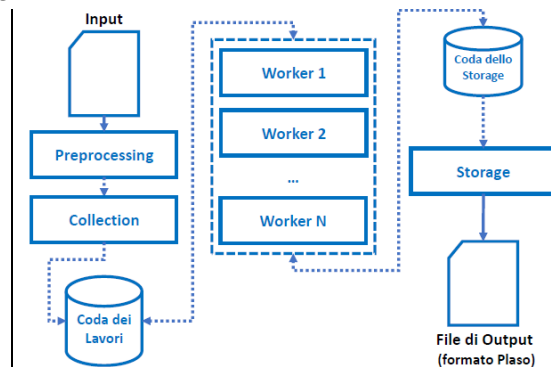
Vengono individuati tutti i file che dovranno essere elaborati nelle fasi successive.

▪ Worker:

Elaborano i file della lista degli input individuata dall'attività di collection. Un worker si occuperà di determinare la tipologia del file e svolgerà diverse attività come determinare quale parser dovrà essere applicato al file (il parser è in grado di elaborare un file in base alla sua struttura), elaborare il file con il parser adeguato, applicare alcuni filtri predefiniti al file, inviare le informazioni estratte alla componente di storage, determinare se si sta elaborando un archivio, il quale potrebbe contenere al suo interno dei file che devono essere elaborati.

▪ Storage:

Si occupa della costruzione/memorizzazione del file di output, in accordo alle specifiche fornite.



Plaso mette a disposizione diversi tool utilizzabili da linea di comando:

- **log2timeline**: È il front-end di Plaso e permette di interagire con il framework, estraendo i vari eventi e memorizzandoli in formato Plaso. I file nel formato Plaso possono poi essere riutilizzati, ulteriormente elaborati ed analizzati;
- **pinfo**: Semplice tool che permette di estrarre e visualizzare informazioni contenute all'interno di un file, in formato Plaso;
- **pprof**: Serve soprattutto per gli sviluppatori, al fine di ottimizzare determinati parser;
- **preg**: Fornisce un front-end diverso dedicato alla gestione dei parser del registro di sistema dei sistemi Windows-based. Permette anche di ottenere informazioni sul registro, partendo da una sua sottochiave;
- **pshell**: Terminale per l'interazione con il back-end di Plaso, permette l'analisi avanzata tramite l'accesso a tutte le librerie di Plaso;
- **psort**: Importante tool che effettua la conversione dal formato Plaso (non human-readable) a diversi formati, che possono essere visualizzati e post elaborati (eventualmente con tool esterni, come, ad esempio, Microsoft Excel, ecc.).

ESEMPI DI UTILIZZO 1 - Cronologia Chrome:

L'analisi della timeline della cronologia del browser è importante dal punto di vista forense, per avere un quadro più chiaro sulle attività effettuate tramite il Web.



Esempio di Utilizzo 1
Cronologia Browser Chrome

1. Il primo passo consiste nel dare in input al comando **log2timeline** il file relativo alla cronologia di Chrome di cui si intende analizzare la timeline, al fine di ottenere in output un file in formato Plaso. Nei sistemi Windows-based, il file relativo alla cronologia è **History**. **Plaso** è in grado, mediante il preprocessing, di riconoscere il formato del file ed analizzarlo.
2. Tramite **psort** è possibile convertire il file in formato Plaso (non human-readable), in un formato human-readable, ad esempio il formato XLSX, leggibile da Microsoft Excel.
3. Analisi del File il formato XLSX Prodotto:

datetime	timestamp_desc	source	source_long	message	parser	display_name
2019-02-08 09:20:11,298	Last Visited Time	WEBHIST	Chrome History	http://www.google.it/ (Google) [count: 0] Visit from: http://sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 09:20:11,298	Last Visited Time	WEBHIST	Chrome History	https://www.google.it/?gws_rd=ssl (Google) [count: 0] Visit sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 09:20:11,298	Last Visited Time	WEBHIST	Chrome History	http://google.it/ (Google) [count: 1] Type: [TYPED - User typ:sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 09:20:15,570	Last Visited Time	WEBHIST	Chrome History	https://www.google.it/search?source=hp&ei=yOldXlbgHYfYf sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 09:20:18,890	Last Visited Time	WEBHIST	Chrome History	https://www.unisa.it/ (UNISA Home) [count: 0] Type: [LIN sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 10:27:16,288	Last Visited Time	WEBHIST	Chrome History	http://docenti.unisa.it/000769/didattica (Alfredo DE SANTI:sqlite/chrome_27_history	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 10:27:54,000	Content Modification Time	FILE	OS Content Modification Time	OS:E:\Users\Raffaele\AppData\Local\Google\Chrome\User filestat	OS:E:\Users\Raffaele\AppData\Local\G	
2019-02-08 10:27:54,000	Metadata Modification Time	FILE	OS Metadata Modification Time	OS:E:\Users\Raffaele\AppData\Local\Google\Chrome\User filestat	OS:E:\Users\Raffaele\AppData\Local\G	

- **Datetime**: riporta il timestamp, esplicitando la data e l'ora, nel seguente formato: YYYY-MM-DD HH:mm:ss,III;
- **Message**: riporta una descrizione, esplicitando alcune informazioni utili;
- **timestamp_desc**, **source** e **source_long**: indicano rispettivamente la tipologia del timestamp (desc), l'identificativo della fonte (source) da cui è stato estrapolato il timestamp, e la descrizione in formato human-readable della fonte (source_long);
- **parser**: Per ciascun evento ne riporta il parser, utilizzato da Plaso;
- **display_name**: è specificato, in questo caso, il percorso completo del file della cronologia: E:\[...]\History.
- **source**: riporta la fonte da cui è stato estrapolato il timestamp, di ciascun evento.

In primo luogo, consideriamo ed analizziamo il campo **message** di alcuni eventi per reperire eventuali informazioni utili. Avendo a disposizione gli URL dei vari siti visitati, estratti dalla cronologia, sarebbe possibile accedere direttamente a tali siti, per visionarne l'eventuale contenuto, senza effettuarne una analisi preventiva (o effettuandone una superficiale).

Gli URL potrebbero essere diretti a siti malevoli. Una richiesta proveniente da una fonte sconosciuta (effettuata con il link diretto dall'investigatore) potrebbe mettere «in allerta» gli «amministratori» dei suddetti siti (alterando il corso di altre indagini). Pertanto, un'accurata analisi è sicuramente preferibile ed è consigliabile accedere all'URL solo in caso vi sia sufficiente sicurezza. Oppure, gli URL potrebbero innescare azioni, ad esempio cancellazione di dati, attivare/disattivare oggetti, ecc.

Analizzando meglio il file (cronologia) è possibile individuare quali siano le due fonti da cui sono stati reperiti i timestamp:

- **Contenuto del file History** (cronologia browser): I timestamp sono relativi al momento dell'ultima visita (Last Visited Time) di un URL;
- **Metadati del file system**, relativi al file History: Tali metadati sono reperiti dalla file table (del file system) considerando l'entry associata al file History. Ad esempio, in NTFS, i suddetti metadati sono contenuti nelle entry della MFT.

Nella tabella sopra, Plaso riporta le informazioni MAC in merito ai timestamp provenienti dai metadati del file system.

Le tracce provenienti dalle due fonti (metadati del file system e cronologia del browser) risultano essere coerenti. Infatti, tutti e tre i timestamp riportano orari coerenti, relativi all'attività di navigazione Web, individuati dalla cronologia del browser.

ESEMPI DI UTILIZZO 2 – File OOXML:

L'analisi dei timestamp dei file Office (memorizzati nel formato Office Open XML) è particolarmente importante, a volte tali file sono utilizzati come alibi, ecc.



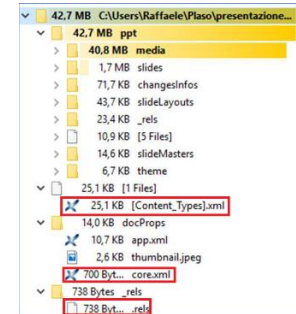
In questo esempio, verrà realizzata una super timeline utilizzando un file (presentazione.pptx) in formato Office Open XML.

Un file OOXML è un **contenitore compresso** in formato ZIP. Contiene diversi elementi e file XML che sono metadati che specificano la struttura del documento ed altre informazioni.

Per estrarre e visionare la struttura interna di un file in formato Office Open XML è sufficiente decomprimerlo in una qualsiasi cartella.

Il file core.xml (nella directory docProps) contiene alcuni metadati che indicano informazioni di carattere generale in merito al documento (titolo, creatore, data e ora di creazione, data e ora dell'ultima modifica, ecc.).

Il file .rels (file XML) nella directory _rels contiene eventuali informazioni sulle relazioni di alcuni elementi strutturali del documento.



1. Come prima, effettuiamo la memorizzazione del file di output in formato Plaso fornendo in input al comando **log2timeline** il percorso relativo al file (presentazione.pptx). Successivamente viene creato il file **EsempioOOXML.plaso**.
2. Mediante il tool **psort**, il file in formato Plaso (output del tool log2timeline) verrà convertito in formato XLSX:
3. Analisi del File Prodotto (EsempioOOXML.xlsx):

datetime	timestamp_desc	source	source_long	message	parser
2019-01-09 07:04:45,000	Creation Time	META	Open XML Metadata	Creating App: Microsoft Office PowerPoint App version: 1 (czip/oxml	
2019-02-06 12:15:46,000	Content Modification Time	FILE	OS Content Modification Time	OS:C:\Users\Raffaele\Plaso\presentazione.pptx Type: file	filestat
2019-02-06 12:15:46,000	Content Modification Time	META	Open XML Metadata	Creating App: Microsoft Office PowerPoint App version: 1 (czip/oxml	

Inerente al primo record in tabella, il file **presentazione.pptx** è stato creato il 09/01/2019 alle ore 07:04:45 (campo datetime). Le informazioni sono state reperite dai metadati del formato OOXML (indicato dai campi source, source_long e parser). Nel campo message vengono riportate altre ulteriori informazioni sul file (autore, applicazione che ha creato il file e relativa versione dell'applicazione, ...).

Per i successivi 2 record in tabella, i timestamp sono reperiti da due fonti diverse:

1. **Metadati del file system**: Tali metadati sono reperiti dalla file table (del file system), considerando l'entry associata al file presentazione.pptx. Ad esempio, in NTFS, i suddetti metadati sono contenuti nelle entry della MFT.
2. **Metadati del formato OOXML**: Sono memorizzati all'interno del file presentazione.pptx (ovvero, sono memorizzati direttamente nel contenuto del file).

I timestamp provenienti dalle due fonti sono coerenti, infatti, la data e l'ora dell'ultima modifica coincidono, sia considerando i metadati del file system sia considerando i metadati del formato OOXML.

Campo	Valore
datetime	06/02/2019 12:15:46
timestamp_desc	Content Modification Time
source	FILE
source_long	OS Content Modification Time

Campo	Valore
datetime	06/02/2019 12:15:46
timestamp_desc	Content Modification Time
source	META
source_long	Open XML Metadata

In questo esempio, Plaso riporta le informazioni MAC in merito ai timestamp. Nella super timeline, non è quindi presente il timestamp relativo alla creazione (B) del file presentazione.pptx.

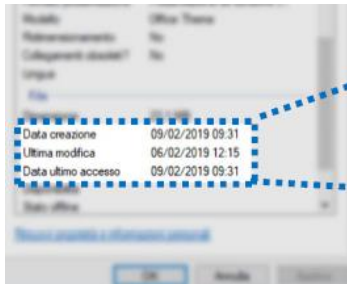
Il timestamp relativo alla creazione del file presentazione.pptx verrà reperito dalla finestra Proprietà di Windows:

Data e ora di creazione fornita da Plaso (ottenuta dai metadati del formato OOXML)	09/01/2019, 07:04:45
Data e ora di creazione, proveniente dai metadati del file system	09/02/2019, 09:31:33

Le informazioni riguardo la data di creazione sono **incoerenti**.

In questo caso, è possibile osservare delle incoerenze direttamente dai timestamp provenienti dai metadati del file system, infatti, è possibile notare che la data di creazione sia successiva alla data dell'ultima modifica.

Questo esempio è stato strutturato con l'obiettivo di enfatizzare e semplificare l'individuazione delle incoerenze, tuttavia, l'individuazione di tali incoerenze non è sempre semplice ed è pertanto necessario porre particolare attenzione.



In generale, una eventuale **incoerenza** tra il timestamp, relativo alla creazione proveniente dai **metadati del file system** e quello proveniente dai **metadati del formato del file** (file OOXML come nell'esempio) dovrebbe far sorgere diversi interrogativi all'investigatore, ad esempio se il file è la copia di un altro file?

- In caso affermativo: L'originale è più aggiornato? L'originale è stato eliminato? per quale motivo? Potrebbe essersi trattato di un errore? L'originale potrebbe contenere dati che volevano essere tenuti nascosti?
- In caso negativo: L'autore del file è il soggetto su cui si indaga? Può averlo creato su un altro dispositivo e poi copiato?

Quando viene recuperato un file eliminato, in formato OOXML, durante le attività di file recovery, potrebbe risultare utile analizzare i metadati del formato OOXML (poiché non è detto che i metadati del file system risultino disponibili).

ESEMPI DI UTILIZZO 3 – File system + cronologia browser:

In questo esempio, verrà realizzata una super timeline, in cui saranno considerate due fonti:

1. **Metadati** dell'intero **file system NTFS** di un **dead system**.
2. **Cronologia del browser** Google Chrome estrapolata dal relativo file (memorizzato nel suddetto **dead system**).

La super timeline viene creata ed analizzata coi tool **fls**, **log2timeline** e **psort**.

Viene eseguito in modo analogo ai precedenti esempi, in più, fls elenca tutti i file e le directory presenti all'interno di un file system, esplicitando per ciascuno di essi diverse informazioni, è stato utilizzato per i timestamp (in formato MACB) di tutti i file/directory.

3. **Analisi** (parziale) **Versione Preprocessata** del File Prodotto:

Il file preprocessato è organizzato in una tabella costituita dalle seguenti tre colonne:

- **Ora**: Riporta l'ora in cui si è verificato l'evento (gli eventi di interesse, si sono verificati tutti nella stessa data);
- **Descrizione**: Riporta una breve descrizione testuale dell'evento (ad esempio, visita di un URL);
- **Path oppure URL**: Specifica il path nel file system o l'URL, coinvolto nell'evento.

OSS: In questo e nei successivi casi, l'attività di navigazione Web, mediante Chrome, presenta tracce osservabili dalla super timeline fra loro coerenti reperite dalle seguenti due fonti:

- Visita di un URL [Informazione ottenuta dalla Fonte 2. – Cronologia di Google Chrome]
- Conseguenti aggiornamenti dei file di cache di Google Chrome [Informazioni ottenute dalla Fonte 1. – Metadati del file system]

La cartella Prefetch:

Spesso, vengono effettuate alcune operazioni, svolte in automatico da Windows (comportamento legittimo), potenzialmente rilevanti per l'investigazione, poiché probabilmente effettuate a seguito di azioni dell'utente.

Volendo analizzare un record in tabella:

18:38:49 | **Creazione** | **C:/Windows/Prefetch/MSPAINTE.EXE-76E10B24.pf**

La cartella **C:\Windows\Prefetch** fa riferimento ad una cartella speciale che viene utilizzata da Windows per incrementare le performance di sistema, effettuando un pre-caricamento di alcune «parti» di codice delle applicazioni usate più comunemente. Dal punto di vista forense è molto utile al fine di individuare quali applicativi sono stati utilizzati nel sistema.

Ogni «parte» di applicazione/processo viene mappata in un file, il cui nome ha il seguente formato:

<NOMEFILE_ESEGUIBILE>-<VALORE_HASH>.pf

In questo caso, Windows ha memorizzato, nella cartella Prefetch, una «parte» di codice dell'applicazione, avente come eseguibile il file **MSPAINTE.EXE**, poiché probabilmente il software è stato avviato dall'utente. L'eseguibile **MSPAINTE.EXE** fa riferimento a Paint.

Proseguiamo analizzando le successive due entry:

18:39:15 | **Modif. del Cont.** | **C:/Users/Raffaele/AppData/Roaming/Microsoft/Windows/Recent**
18:39:15 | **Modif. del Cont.** | **C:/Users/Raffaele/AppData/Local/Microsoft/Windows/History**

Anche queste due entry sono potenzialmente rilevanti, in quanto si riferiscono a due cartelle speciali di Windows, all'interno delle quali sono memorizzate informazioni sui documenti che sono stati utilizzati di recente dall'utente.

OSS: Il fatto che vi siano state delle modifiche nel contenuto di queste directory fa supporre che vi sia stata l'apertura o salvataggio di un file. Considerando che è stato aperto il software Paint, si potrebbe supporre che vi sia stata l'apertura o il salvataggio di una immagine.

Proseguiamo l'analisi, con la **18:39:15** | **Creazione** | **C:/Recycle.Bin/S-1-5-21-3031900839-921391284-2575565698-1001/\$R33G7OQ.png** prossima entry:

OSS: La cartella **C:/Recycle.Bin** è la cartella che Windows utilizza per il Cestino. Il fatto che sia stato creato un nuovo file all'interno del Cestino fa supporre che l'utente abbia eliminato un file, potrebbe essere utile approfondire di cosa si tratti.

Quando un file viene «eliminato» (spostato nel Cestino), il S.O. lo rinomina senza alterarne l'estensione e lo sposta nella cartella **\$Recycle.Bin**, quindi, anche dal «nuovo nome del file», è possibile individuare l'estensione del file «eliminato» (nell'esempio è .png).

Il codice che si trova prima dell'immagine è il Secure Identifier dell'utente (SID) un identificativo univoco associato a ciascun utente all'interno del sistema.

Proseguiamo l'analisi, con la **18:39:15** | **Creazione** | **C:/Users/Raffaele/AppData/Roaming/Microsoft/Windows/Recent/DigitalForensics.Ink** prossima entry:

Con l'estensione .Ink, in Windows, si fa riferimento ad un collegamento rapido (link), tramite il quale si accede al programma o file a cui il collegamento stesso punta. Generalmente, i collegamenti sono creati automaticamente dal sistema e vengono riportati in apposite liste di documenti recenti per essere rapidamente richiamati dall'utente.

C:/Users/Raffaele/AppData/Roaming/Microsoft/Windows/Recent è una delle due cartelle speciali di Windows che memorizza proprio i collegamenti rapidi che vengono mostrati in specifiche liste di file/documenti recenti.



Quando il collegamento viene creato in automatico (come avvenuto probabilmente in questo caso), il nome del collegamento (*DigitalForensics.lnk*) fa riferimento ad un file denominato DigitalForensics (non è possibile individuare l'estensione originale). La navigazione prosegue con la visita ad altri quattro link ed il conseguente aggiornamento della cache:

18:39:26	Visita URL	https://www.di.unisa.it/home/news
[Aggiornamenti Cache di Google Chrome e/o File di Sistema e/o Microsoft Cortana, assistente vocale di Windows 10]		
18:39:28	Visita URL	https://www.di.unisa.it/home/news?archive=1
18:39:40	Visita URL	https://www.di.unisa.it/unisa-rescue-page/dettaglio/id/1401/module/475/row/3837/il-dipartimento-di-informatica-ospita-l-iniziativa-coding-girls
[Aggiornamenti Cache di Google Chrome e/o File di Sistema e/o Microsoft Cortana, assistente vocale di Windows 10]		
18:39:45	Visita URL	https://www.di.unisa.it
[...]		

Dopo la visita all'URL: <https://www.di.unisa.it>, avvenuta alle ore 18:39:45 (09/02/2019), dalla super timeline, è osservabile che non siano stati visitati ulteriori URL, mediante Chrome. Infatti, dal file History, relativo alla cronologia di Google Chrome, non risultano tracce di ulteriori URL visitati.

Inoltre, dalla super timeline è osservabile che l'ultima modifica, al file History, risulti essere avvenuta alle 18:39:49 (09/02/2019):

18:39:49,000	Content Modification Time	OS:E:\Users\Raffaele\AppData\Local\Google\Chrome\User Data\Default\History Type: file
--------------	---------------------------	---

Le suddette tracce, provenienti dalla Fonte 1. (Metadati) e dalla Fonte 2. (Cronologia del browser), sono dunque coerenti: dopo alcuni secondi dalla visita dell'URL, di cui sopra, è stato verosimilmente aggiornato il file della cronologia (e, conseguentemente, la data e l'ora relative alla modifica del file).

18:36:51,000	Last Access Time	OS:E:\Users\Raffaele\AppData\Local\Google\Chrome\User Data\Default\History Type: file
18:36:51,000	Metadata Modification Time	OS:E:\Users\Raffaele\AppData\Local\Google\Chrome\User Data\Default\History Type: file

Anche l'ora dell'ultimo accesso (Last Access Time) e l'ora dell'ultima modifica dei metadati del file (Metadata Modification Time), le quali risultano essere uguali a 18:36:51 (in entrambi i casi la data è 09/02/2019), risultano essere coerenti con l'attività di navigazione Web, individuata dalla cronologia di Google Chrome (Fonte 2.). È possibile supporre che il file sia stato acceduto, da Google Chrome, per fornire suggerimenti all'utente, su un sito già visitato o altre attività, comunque correlate alla navigazione Web.

POSSIBILE SCENARIO:

Dalle osservazioni e le informazioni precedenti, potrebbe essere possibile delineare uno **scenario** verosimile degli eventi, sufficientemente accurato, in riferimento alla (piccola) porzione della super timeline analizzata:

1. Navigazione Web sulla pagina Eventi del Dipartimento di Informatica (DI) dell'Università di Salerno;
2. Visita alla pagina dedicata di un evento archiviato (già svoltosi);
3. Avvio del programma Microsoft Paint;
4. Visualizzazione/Creazione di una immagine (probabilmente tramite Paint);
5. Cancellazione di una immagine PNG (probabilmente l'immagine di cui sopra);
6. Creazione (verosimilmente automatica) di un collegamento rapido (.lnk) ad un file il cui nome è DigitalForensics (non è nota l'estensione del file al quale il collegamento fa riferimento);
7. La navigazione Web prosegue con la visita ad altri URL, relativi ad altre pagine del suddetto Dipartimento.

La versione per Microsoft Windows, del tool Autopsy, prevede diverse opzioni per la gestione della super timeline, tutte gestibili mediante una GUI molto curata ed user-friendly.

6. NETWORK FORENSICS

La **network forensics** è un ramo della digital forensics che si occupa degli **aspetti forensi riguardanti le reti** (apparati di rete, ecc.). Principalmente, la network forensics viene eseguita sui **live system**, viene acquisito il traffico raw prodotto dalle interfacce di rete di un dispositivo informatico come pacchetti, eventuali log, ecc. Tale traffico viene poi analizzato.

IL TOOL XPLICO:

Xplico è un tool Open Source che permette l'**analisi forense di traffico di rete**. Xplico permette di analizzare file che contengono acquisizioni di rete, con estensione **.pcap (packet capture)**. All'interno del traffico di rete possiamo trovare tracce utili per le indagini, ad esempio indirizzi Web di Siti Visitati, contenuto di E-mail, chat di Social Network, pacchetti VoIP o file stampati.

ESEMPIO DI UTILIZZO 1 – HTTP E WEB:

Una volta effettuato il login su Xplico e creato un nuovo caso, sarà possibile importare i file con estensione **.pcap** da far analizzare al tool. Dopo che l'elaborazione è terminata verrà mostrata l'analisi completa dei dati:

Sessione dei dati		Pcap set	
Caso e Sessione Cap. Ora inizio Cap. Ora di fine Stato Host		CasoXplico -> HTTPWEB 2009-12-09 17:42:17 2009-12-09 17:42:50 DECODING COMPLETED Filtro	
		PCAP-over-IP TCP port: 30001. Aggiungi nuovo file pcap. Browse... No file selected. Elabora Lista di tutti i file pcap.	
HTTP Post 0 Get 0 Video 0 Immagini 0		MMS Numero 0 Contenuto 0 Video 0 Immagini 0	
E-mail Ricevute 0 Inviare 0 Non lette 0/0		FTP - TFTP - HTTP di file Connessioni 0 - 0 Scaricato 0 - 0 Caricato 0 - 0 HTTP 0	
Web Mail Totale 0 Ricevute 0 Inviare 0			
Facebook Chat / Paltalk Utenti 0 Chat 0/0		IRC/Paltalk Exp/Msn/Yahoo! Server 0 Canali 0/0/0/0	
Dns - Arp - ICMPv6 DNS res 0 ARP/ICMPv6 0/0		RTP / VoIP Video 0 Audio 0	
NNTP Gruppi 0 Articoli 0			
Feed & Printed files Numero 0 Pdf 0		WhatsApp Connection 0	
Telnet / Syslog Connessioni 0/0		SIP Chiamate 0	
Sconosc. Testi 5/18 Dig 9			

- Nell'area **Sessione dei Dati** vengono mostrate alcune informazioni sui dati acquisiti, fra cui:
 - Data e ora dell'inizio dell'acquisizione
 - Data e ora della fine dell'acquisizione
 - Host da cui è stata effettuata l'acquisizione
- La **Sezione riepilogativa** (i 15 rettangoli in basso) riporta il numero di «artefatti», suddivisi per tipologia.
 - Nella sezione http viene riportato il numero di pacchetti POST, pacchetti GET, pacchetti relativi ad immagini e video
 - Nella sezione E-mail riporta il numero di e-mail ricevute, inviate e non lette
 - Nella sezione Web Mail viene indicato il numero di e-mail (gestite mediante client Web) ricevute, inviate e non lette
 - Nella sezione Facebook Chat/Paltalk vengono riportate le statistiche relative alla chat di Facebook/Paltalk ed ai relativi utenti
 - Nella sezione Sconosc. vengono riportate le statistiche in relazione ad artefatti/oggetti sconosciuti (non decodificati). Nello specifico, sono stati identificati diversi artefatti testuali ed oggetti denominati Dig

OSS: La sezione Sconosc. è l'unica che presenta degli artefatti, pertanto, è necessario approfondire.

È possibile approfondire gli artefatti sconosciuti tramite apposito menù di Xplico, di seguito l'approfondimento:

Per ciascuna connessione viene esplicitata:

- data/ora in cui ha avuto luogo una certa connessione.
- destinazione (indirizzo IP/porta).
- protocollo e durata (in secondi).
- numero di byte generato dalla connessione in esame.

Cliccando sull'**indirizzo IP** di destinazione verrà scaricato un file testuale con il contenuto dei pacchetti.

Cliccando sul link **info.xml** si aprirà una pagina contenente altri dettagli in relazione alla connessione.

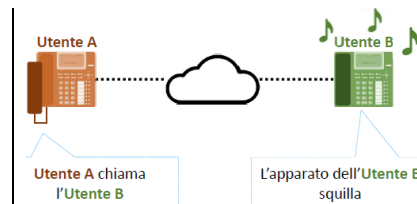
Ricerca				Andare		
Data	Destinazione	Porto	Protocollo	Durata [s]	Dimensioni [byte]	Info
2009-12-09 17:42:47	74.125.77.100	80	Google	0	1190	Info.xml
2009-12-09 17:42:44	75.119.219.154	80	HTTP	6	5763	Info.xml
2009-12-09 17:42:39	67.205.49.173	80	HTTP	10	1065	Info.xml
2009-12-09 17:42:39	67.205.49.173	80	HTTP	10	1041	Info.xml
2009-12-09 17:42:39	67.205.49.173	80	HTTP	10	3723	Info.xml
2009-12-09 17:42:36	67.205.49.173	80	HTTP	4	25182	Info.xml
2009-12-09 17:42:36	67.205.49.173	80	HTTP	13	15419	Info.xml
2009-12-09 17:42:36	67.205.49.173	80	HTTP	13	29258	Info.xml
2009-12-09 17:42:33	67.205.49.173	80	HTTP	6	8381	Info.xml
2009-12-09 17:42:28	67.205.51.26	80	HTTP	6	810	Info.xml
2009-12-09 17:42:27	195.37.77.138	80	HTTP	7	6436	Info.xml
2009-12-09 17:42:27	216.34.181.71	80	HTTP	7	3204	Info.xml
2009-12-09 17:42:22	67.205.51.26	80	HTTP	6	39856	Info.xml
2009-12-09 17:42:20	67.205.51.26	80	HTTP	8	13702	Info.xml
2009-12-09 17:42:20	67.205.51.26	80	HTTP	8	10795	Info.xml
2009-12-09 17:42:20	67.205.51.26	80	HTTP	8	15386	Info.xml
Precedente			1 2 1 of 2	Prossimo		

ESEMPIO DI UTILIZZO 2 – VoIP:

Voice over IP (VoIP) definisce un insieme di protocolli. Il segnale analogico, prodotto dalla voce, viene convertito in un segnale digitale, il quale viene incapsulato in pacchetti. Ciò permette di effettuare chiamate telefoniche mediante le infrastrutture di rete.

Una telefonata VoIP si articola in due fasi principali:

1. **Setup**, viene instaurata una sessione: il chiamante cerca di contattare il chiamato, mediante le varie infrastrutture di rete (router, gateway, ecc.) e utilizzando diversi protocolli (SIP, ecc.);
2. **Flusso Audio**, se chiamante e chiamato hanno accettato la sessione (rispondendo al telefono), viene avviato il flusso audio (ovvero la telefonata). Il protocollo utilizzato è RTP (Real-Time Protocol).



Il protocollo **SIP (Session Initiation Protocol)** permette di iniziare, modificare e terminare sessioni per chiamate telefoniche o conferenze con più flussi multimediali. Con il protocollo SIP, l'identificativo (telefonico) è associato all'utente e non al terminale. È un protocollo di tipo Client-Server con scambio di messaggi testuali (analogia con il protocollo http).

Il **traffico di rete generato dal servizio VoIP** può essere una risorsa per gli investigatori forensi, siccome è possibile individuare tracce di comunicazione come data e ora, durata della comunicazione, identificativo del chiamato/chiamante (chiamate in uscita/in entrata), ecc.

Analogamente all'esempio precedente, si importa il file .pcap e lo si analizza tramite schermata di Xplico.

Nella sezione riepilogativa, sono state individuate 2 chiamate, tramite servizio VoIP (con il protocollo SIP).

Anche in questo caso è possibile approfondire, tramite menù del tool, questi dati.

Sono state effettuate due chiamate da "Freeswitch", in data esplicitata e di una determinata durata

Cerca: <input type="text"/>			Andare
Data	Da	A	Durata
2007-10-31 12:14:23	"FreeSwitch" <sip:5555551212@192.168.1.111>	<sip:6580@192.168.1.12>	0:0:0
2007-10-31 12:14:23	"FreeSwitch" <sip:5555551212@192.168.1.111>	<sip:6580@192.168.1.12>	0:0:19

Cliccando sulla durata è possibile ottenere maggiori informazioni:

Cliccando su cmd.txt è possibile visionare il contenuto di alcuni pacchetti del protocollo SIP, all'interno dei quali potrebbero esservi informazioni utili all'investigatore.

Data:	2007-10-31 12:14:23	
Da:	"FreeSwitch" <sip:5555551212@192.168.1.111>	play
A:	<sip:6580@192.168.1.12>	play
Durata:	0:0:19	
Comandi:	cmd.txt	
Info:	Info.xml	

Tramite il valore del campo User-Agent, nei pacchetti SIP, è possibile individuare il modello (o i modelli) degli apparati che hanno effettuato la comunicazione (nell'esempio, si tratta di un Siemens OptiPoint 400 Standard).

```
SIP/2.0 180 Ringing
Call-ID: a83ec57b-024d-122b-2780-39a48cb53b8d
CSeq: 90798095 INVITE
From: "FreeSwitch" <sip:5555551212@192.168.1.111>;tag=NZcQcBB9gXtSK
To: <sip:6580@192.168.1.12>;tag=bc9a95aff6448a9
Via: SIP/2.0/UDP 192.168.1.111;branch=z9hG4bKNSDyctFHeeF8m;rport
Content-Length: 0
Contact: tel sip <sip:6580@192.168.1.12:5060;transport=udp>
User-Agent: optiPoint 400 standard
```

ESEMPIO DI UTILIZZO 3 – E-mail:

I principali protocolli per l'invio di e-mail sono:

- **Simple Mail Transfer Protocol (SMTP)**: Utilizza la porta 25 ed è usato per l'invio di e-mail;
- **Post Office Protocol (POP3)**: Utilizza la porta 110 ed è usato per la ricezione di e-mail che permette di effettuarne il download dal server e-mail al client;
- **Internet Message Access Protocol (IMAP)**: Utilizza la porta 143 ed è utilizzato per la ricezione di e-mail che permette di effettuarne il download dal server e-mail al client, ma una copia delle e-mail è lasciata sul server, in modo da permettere l'accesso anche da altri client (ad esempio, client Web, ecc.).

Creando una nuova sessione in Xplico e importando il file .pcap, ci verrà mostrata l'analisi corrispondente.

Nella sezione riepilogativa, dalla sezione E-mail, Xplico ha individuato una e-mail.

Per ottenere ulteriori informazioni è possibile usare il menù di Xplico.

E-mail	
Ricevute	0
Inviare	1
Non lette	1/1

È stata individuata una e-mail con tutte le info esplicitate:

Data	Soggetto	Mittente	Ricevitori	Dimensione	Rilevanza (?)
2009-10-05 07:06:08	-(no subject)-	gurpartap@patriots.in	raj_deol2002in@yahoo.co.in	14544	
Precedente		1 of 1			Prossimo

Cliccando sul soggetto dell'e-mail (no subject) è possibile visionare il contenuto della e-mail:

Viene riportato il protocollo (in questo caso, SMTP), la data e l'ora di invio dell'e-mail e viene fornita la possibilità di scaricare l'e-mail in formato EML, il quale è un formato standard, in accordo alla RFC 5322, e può essere aperto da diversi client e-mail.

Infine, viene riportato il contenuto dell'e-mail ed eventuali allegati all'e-mail (allegato testuale NEWS.txt).

Cliccando sul nome di un allegato, è possibile scaricarlo e/o visionarlo.

<raj_deol2002in@yahoo.co.in>	
SMTP	
Gurpartap Singh <gurpartap@patriots.in>	
<raj_deol2002in@yahoo.co.in>	
Mon, 5 Oct 2009 11:36:07 +0530	
email.eml	
Hello	
I send u smtp pcap file	
Find the attachment	
GPS	
text	NEWS.txt

WIRESHARK:

Wireshark è uno sniffer di traffico di rete ed è in grado di catturare il traffico di rete in modalità promiscua (traffico passato alla CPU ed è possibile memorizzarlo). Tutte le operazioni fatte con Xplico possono essere effettuate anche da questo tool.

7. TECNICHE ANTI-FORENSI

L'**Anti-Forensics** (AF) è una collezione di strumenti e tecniche atte a mettere in difficoltà gli strumenti forensi, gli investigatori ed il normale svolgimento dell'indagine. L'Anti-Forensics ha come obiettivi:

- Evitare che vengano individuate tracce di eventi che hanno avuto luogo;
- Innescare dubbi sul report di una indagine;
- Interrompere la raccolta di informazioni;
- Aumentare il tempo necessario per lo svolgimento di una indagine.

ELIMINARE LE EVIDENZE:

In uno scenario reale si eliminano le evidenze come un'arma del delitto oppure pulire una superficie contenente impronte. Nel mondo digitale, invece, è possibile effettuare la **Sovrascrittura di Dati e Metadati**.

Esistono tool che permettono di sovrascrivere dati rilevanti per l'indagine, eliminandoli. Questi tool operano in tre modalità:

1. **Sovrascrittura dell'intero dispositivo di memorizzazione**, tramite pulizia forense che prevede la sovrascrittura del contenuto di ogni settore del disco fisso con valori nulli o random;
2. **Sovrascrittura di singoli file**;
3. **Sovrascrittura dell'unlocated space**, il quale potrebbe contenere file eliminati ma ancora presenti sul dispositivo.

Alcuni tool permettono di sovrascrivere i timestamp contenuti nei **metadati** del file system (data di accesso/creazione/modifica) utilizzati per creare timeline, in tal caso, l'ordine degli eventi risulta essere sbagliato.

Il tool **Attribute Changer** permette di modificare i metadati di un file andandosi ad integrare all'interno dell'interfaccia utente windows che permetterà poi di modificare i metadati di un dato file. Un altro tool per fare ciò è **Timestamp** di Metasploit.

NASCONDERE LE EVIDENZE:

La **crittografia** è molto efficiente per nascondere dati, tuttavia, i dati crittografati sono facilmente rilevabili siccome hanno un'**entropia elevata**. Diversi tool per la crittografia, inglobano metadati o **header particolari** all'interno dei file che li rendono riconoscibili.

OSS: difficilmente, partendo da dati cifrati, è possibile recuperarne il contenuto originale. Tuttavia, se utilizzata la crittografia si potrebbe attirare l'attenzione degli investigatori.

In questa categoria ci sono le seguenti tecniche:

- **File System Crittografato**: Viene eseguita la cifratura di file quando questi vengono memorizzati su dispositivi di memorizzazione. I dati vengono decifrati solo quando vi è necessità di effettuare delle operazioni su di essi (esempio lettura/scrittura del file). Un investigatore non può analizzare i file poiché cifrati.
- **Protocolli di rete crittografati**: Il traffico di rete può essere crittografato, esistono diversi protocolli che permettono di crittografare il contenuto del traffico di rete (SSL/SSH). L'idea è che i pacchetti di rete vengono cifrati ed incapsulati, esiste poi l'**onion routing** il quale fa uso di nodi intermedi per proteggere il traffico di rete da eventuali analisi.

Mediante tecniche di **information hiding** è possibile nascondere informazioni in diverse tipologie di file (video, immagini, documenti). Una tecnica di information hiding è la **stenografia** che ha l'obiettivo di nascondere l'esistenza delle comunicazioni ad un soggetto terzo. Un'altra tecnica di information hiding è il **watermarking** dove è possibile nascondere una "filigrana" all'interno di dati (generalmente multimediali come video, audio, immagini, ecc.). Un watermark può essere una specifica sequenza di bit, una stringa o un logo. Questa tecnica viene principalmente usata per la protezione dei copyright dei dati. Da una copia di una immagine potrebbe essere estratta la filigrana e identificato l'effettivo autore dell'immagine.

MINIMIZZARE LE EVIDENZE PROVENIENTI DAI TOOLS:

Sfruttando le vulnerabilità di **buffer overflow** è possibile iniettare codice malevolo (**memori injection**) nello spazio di indirizzi di un programma vittima in esecuzione alterando il comportamento del programma. I buffer overflow sono utilizzati come punto di ingresso in un sistema remoto, in questo scenario, l'attaccante è in grado di memorizzare i tool per l'AF sul sistema remoto.

I **live CD, penne USB bootable** e **macchine virtuali** sono utilizzati come strumenti per l'anti-forensics siccome **lasciano poche tracce**.

Un **live CD** è un supporto di memorizzazione di sola lettura (CD-ROM, DVD) che permette l'avvio e l'esecuzione di un S.O. senza che venga installato sulla macchina.

Analogo è una **penna USB bootable**, la differenza è che qui è possibile effettuare anche operazioni di scrittura. Un attaccante può memorizzare dei file creati sul S.O. che è stato avviato dalla penna USB.

Con Live CD e USB bootable è possibile effettuare attacchi non lasciando alcuna traccia.

Una **macchina virtuale** si tratta di un S.O. client che viene eseguito in un programma. Il sistema che esegue il suddetto programma e il S.O. client viene detto sistema "host", dove vengono memorizzati gli "stati" del S.O. client ed un piccolo insieme di file di configurazione. A seguito di un attacco sul S.O. client, il malintenzionato dovrebbe solo cancellare in modo sicuro i file associati alla macchina virtuale.

ACCESSI E MEMORIZZAZIONI ANONIME:

Un malintenzionato potrebbe utilizzare diversi **account anonimi** o **falsi su vari servizi Cloud** che, al momento della registrazione, forniscono una quantità di spazio. I malintenzionati potrebbero utilizzare lo spazio al fine di memorizzare dei tool per l'AF ed eventuali informazioni acquisite.

SFRUTTARE BUG DEI TOOL PER L'INVESTIGAZIONE FORENSE:

Allo stesso modo di qualsiasi altro software, anche i tool forensi dovrebbero svolgere adeguati **controlli sull'input**, evitando di incorrere in potenziali attacchi come il buffer overflow. Questi attacchi potrebbero arrecare problemi durante lo svolgimento delle indagini. In alcuni casi, l'uso di risorse (CPU, RAM, disco, ecc.) da parte di alcuni tool forensi è **dipendente dai dati di input**, le suddette risorse potrebbero essere soggette ad attacchi di tipo **DoS (Denial of Service)**. Mediante tecniche di compressione dei dati, è possibile produrre un attacco DoS denominato **compression bombs attack**, dove vengono realizzati particolari file compressi e i tool forensi, analizzando questi file, devono utilizzare notevoli quantità di risorse, soprattutto in termini di spazio del disco. Un esempio di questo tipo di file è il 42.zip che contiene 16 file zippati che a sua volta hanno altri 16 file zippati e così ancora...

Alcuni tool forensi necessitano di conoscere la tipologia di file al fine di una elaborazione efficiente. Per identificare la tipologia di un file, i tool si basano sull'header del file. **Conoscendo le euristiche utilizzate** dai tool, un utente malizioso può sfruttare tutto ciò a suo vantaggio andando ad alterare l'header del file in modo che un tool di file recovery potrebbe non riuscire a ripristinare quel file.

RILEVARE L'UTILIZZO DI TOOL PER L'INVESTIGAZIONE FORENSE:

La maggior parte dei dischi fissi integra una tecnologia chiamata **S.M.A.R.T.** (Self-Monitoring, Analysis and Reporting Technology) che monitora sé stesso fornendo diverse informazioni diagnostiche come numero totale accensioni, tempo totale di attività, temperatura elevate raggiunte e altri attributi. Queste informazioni possono essere lette da specifici tool come CrystalDiskInfo. Non è possibile effettuare il reset delle informazioni tracciate dalla tecnologia S.M.A.R.T., è possibile disabilitare il tracciamento delle informazioni diagnostiche ma solo alcuni modelli implementano questo comando, in realtà, anche se disabilitate, questo tipo di tecnologia continua a tenere traccia di alcune informazioni.

I tool per l'Anti-Forenses possono **trarre beneficio** dalle informazioni dalla tecnologia S.M.A.R.T. per cercare di capire se sono già stati utilizzati determinati tool per l'analisi forense, se non fossero stati usati l'attaccante potrebbe valutare l'uso di strategie per alterare il comportamento dei dischi, ad esempio un aumento del tempo di attività del disco potrebbe far intuire ad un attaccante che è stato usato un tool per l'acquisizione di una immagine forense.

Molti tool per la **network forensics** acquisiscono il traffico di rete in modalità promiscua (acquisisce tutti i pacchetti sulla rete e non solo verso un determinato host). In generale, gli host che effettuano il monitoring della rete non dovrebbero essere in grado di trasmettere sulla rete che stanno monitorando. Tuttavia, i suddetti host non sono spesso configurati correttamente ed è possibile identificare e attaccare questi host analizzando le loro risposte a pacchetti malformati.

ALCUNE CONTROMISURE:

Alcune tecniche anti-forenses possono essere superate migliorando i tool forensi utilizzando controlli più rigidi, ecc.

Inoltre, è possibile mettere in difficoltà i tool per la sovrascrittura dei dati/metadati memorizzando questi ultimi in **supporti di sola lettura** (CD, DVD, ecc.) siccome non possono essere alterati.

Un'altra possibilità è quella di inviare dei log ad un **host remoto** al quale un attaccante non abbia accesso.

I compression bombs attack potrebbero essere evitati con degli **avvisi di comportamenti potenzialmente anomali** da parte dei tool, per esempio se la decompressione di un file richiede troppo tempo rispetto ad una certa soglia.

Per la crittografia e file system crittografati è possibile **recuperare password o chiavi crittografiche** utilizzando spyware, key logger e altre tecniche.