CAPITOLO QUARTO

LA TIMELINE: ASPETTI TECNICI E RILEVANZA PROCESSUALE

Vincenzo Calabrò, Paolo Dal Checco, Bruno Fiammella

Sommario: 1. Premessa. - 2. La Timeline. - 3. Il problema dell' "ora esatta". - 4. Metodologia. - 4.1 Analisi dei timestamp presenti nei file system. 4.2 Analisi dei timestamp contenuti all'interno dei file. 4.3 Riscontro con altri riferimenti temporali rilevabili. 4.4 Contestualizzazione dei timestamp. 5. Case study: creazione ed analisi della timeline. - 5.1 I problemi dell'analisi tradizionale. 5.2 Anti-forensics. 5.3 Estensione delle timeline. 5.4 Impostazione del case study. 5.5 I tool per la generazione di timeline. 5.6 I tool per la generazione di supertimeline. 5.7 Log2Timeline e le supertimeline. 5.8 Ambiente di lavoro. 5.9 "Montaggio" dell'immagine forense. 5.10 Estrazione della tabella. MFT 5.11 Generazione della supertimeline. 5.12 Analisi dei risultati. 5.13 Strumenti per raffinare l'analisi. 5.14 Note pratiche sulla sincronizzazione temporale. 6. Valore probatorio. - 7. Conclusioni. - 8. Bibliografia.

1. PREMESSA

La riconducibilità di un determinato fatto, in un preciso spazio temporale, è, senza ombra di dubbio, uno degli elementi primari per la corretta interpretazione della *scena criminis*, in quanto consente di rivelare la dinamica degli eventi nell'ordine in cui gli stessi si sono verificati. In caso di omicidio, la prima domanda che gli inquirenti rivolgono al medico legale è quella di stabilire la data e l'ora del decesso, rappresentando un momento fondamentale per la ricostruzione degli avvenimenti che si sono verificati prima dello stesso e subito dopo. La ricostruzione della sequenza temporale degli eventi che hanno determinato un fatto è quindi d'interesse vitale per la risoluzione di qualsiasi caso, di natura legale o professionale, indipendentemente dal contesto di riferimento. Uno degli strumenti che consentono di effettuare l'analisi forense sul tempo è la cosiddetta timeline, ovvero la rappresentazione o esposizione cronologica e concatenata di avvenimenti chiave all'interno di un particolare arco temporale. Nell'informatica forense, con la locuzione timeline s'intende una "fotografia" di tutti gli eventi storici avvenuti in un determinato sistema informatico o telematico. Viene ricostruita mettendo in ordine cronologico tutti gli eventi successivi in un determinato tempo di vita del sistema posto sotto

analisi. Avere dei punti di riferimento temporali aiuta sia la ricerca, che l'analisi delle informazioni e riduce significativamente il numero di ipotesi che si possono formulare durante lo svolgimento delle indagini preliminari. Un'azienda privata, per esempio, preoccupata della sicurezza delle proprie informazioni, può decidere di utilizzare la timeline, come strumento di prevenzione da un eventuale evento doloso, per individuare una procedura di attacco alla sicurezza informatica, oppure per evidenziare le eventuali vulnerabilità dell'infrastruttura tecnologica.

Al contrario, le forze dell'ordine potrebbero utilizzare la timeline per sostenere l'identificazione di una persona sospettata di aver commesso un crimine o per preservare le potenziali fonti di prova.

È comunque evidente, in entrambi i casi, che la ricostruzione di una timeline, precisa e credibile, risulta indispensabile per la ricerca della soluzione o per ridurre la durata di un'indagine. L'assenza di una cronologia accurata degli eventi può comportare, a volte, anche la difficile ricostruzione degli eventi e la conseguente applicazione al caso di specie, della fattispecie penale tipica di riferimento per una determinata condotta.

Lo scopo di questo lavoro è di offrire uno spunto di riflessione sull'argomento, utile sia ai consulenti tecnici di parte (o periti informatici nominati dal tribunale), che agli operatori del settore giuridico (magistrati, avvocati, forze dell'ordine ed investigatori).

Su questo tema le investigazioni digitali non fanno eccezione, anzi, come vedremo, l'ambiente digitale offre spunti di grande interesse sotto l'aspetto investigativo e probatorio.

Ancora oggi, sfortunatamente, non è universalmente percepita l'importanza e la delicatezza della prova digitale nel procedimento giudiziario. Per esempio, vi sono casi in cui evidenze fondamentali (fonti di prova o indizi) non emergono durante la fase delle indagini, perché i periti o consulenti del pubblico ministero non li considerano rilevanti, e di conseguenza vengono ritenute apparentemente inutili, ma in realtà rivestono un peso specifico di alto valore probatorio. Ciò può diventare un grosso problema nel momento in cui queste stesse informazioni possono contribuire a disegnare il quadro completo di un crimine tecnologico. Per questo motivo è importante esaminare tutti gli elementi di prova, indipendentemente dal fatto che possano apparire insignificanti.

L'investigatore esperto, infatti, è colui che esamina tutti i dati, compresi quelli temporali, a sua disposizione al fine di trovare fonti di prova o elementi indizianti utili a fornire una corretta ricostruzione degli eventi, capire il *modus operandi* di un eventuale attacco informatico, il livello di abilità, di conoscenza e la localizzazione. E' chiaro che per far questo, anche il perito informatico deve avere compreso quello che è il vero compito degli investigatori: non avvalorare

una tesi e ricercare elementi di forza per la stessa, ma ricercare ed analizzare ogni elemento alla fine della ricostruzione della verità processuale, che sia il più possibile vicina a quella reale.

2. LA TIMELINE

Il primo esempio di timeline, che viene in mente ad un investigatore digitale, è il log file[1], per la finalità comune che entrambi vogliono raggiungere. Infatti, il log file rappresenta sia un efficace strumento per la sicurezza informatica, perché agevola la ricostruzione delle azioni di un soggetto che "viola" un sistema informatico o telematico, che un'ottima fonte d'informazioni per l'analisi *a posteriori* di un evento.

Se, viceversa, il sistema di protezione o di rilevamento è stato compromesso, perché mal configurato o completamente disabilitato, l'investigatore, non avendo a disposizione i relativi log, è costretto a cercare altre fonti di prova per poter effettuare l'analisi temporale degli eventi.

I Timestamp[2] associati ad un file sono una valida alternativa in quanto, attraverso le informazioni che rappresentano, possono essere utilizzati per realizzare un log semplice degli eventi accaduti. Sebbene le informazioni di timestamp possono essere considerate unidimensionali, nel senso che registrano solamente l'orario dell'ultima operazione eseguita su un determinato file, possono essere una valida fonte di prova quando rimangono poche alternative.

Purtroppo il trattamento di queste informazioni diventa complesso se il volume dei dati da esaminare è di grandi dimensioni. Inoltre, se alla sovrabbondanza di prove digitali si aggiunge l'esigenza di doverle processare

[1] Con il significato di *giornale di bordo*, o semplicemente *giornale*, su cui vengono registrati gli eventi in ordine cronologico. Il termine è stato importato nell'informatica (1963) per indicare la *registrazione cronologica* delle operazioni man mano che vengono eseguite.

[2] Una marca temporale (timestamp) è una sequenza di caratteri che rappresentano una data e/o un orario per accertare l'effettivo avvenimento di un certo evento. Un timestamp è il momento in cui viene registrato un evento da un computer, non il tempo in cui si manifesta. In molti casi, la differenza può essere senza conseguenze: il momento in cui un evento viene registrato da un timestamp (ad esempio, registrato in un file di log) devono essere molto vicino al momento in cui si è verificato l'evento registrato. Questo dato è di solito rappresentato in un formato standard, consentendo un facile confronto di due diversi registri, la funzione di registrazione di una data e ora in modo coerente con i dati effettivi si chiama timestamping.
I Timestamp sono generalmente utilizzati per la registrazione degli eventi, nel qual caso ogni evento è contrassegnato con un timestamp. Nei filesystem, il timestamp può rappresentare la data di memorizzazione / l'ora di creazione o di modifica di un file.

in poco tempo, si potrebbe configurare il problema della riduzione di audit[3].

Il problema della riduzione di audit descrive la situazione in cui la presenza di troppe informazioni oscura l'obiettivo principale delle indagini. In questo caso la riduzione sull'oggetto della verifica prevarrebbe rispetto all'analisi dell'intero materiale indiziante.

L'analisi dei timestamp dei file è un eccellente esempio del problema della riduzione di audit. La capacità di memorizzazione dei moderni hard drive è dell'ordine di centinaia di gigabyte, fino ad arrivare a decine di terabyte, di conseguenza su questi device possono essere registrati un gran numero di file, ognuno con le proprie informazioni di timestamp associate.

Anche se la maggior parte di queste informazioni potrebbero sembrare irrilevanti in un caso giudiziario, alcune di queste potrebbero rilevarsi la chiave di volta per la risoluzione dello stesso. Pertanto, se questi dati dovessero essere semplicemente trascurati, si potrebbe giungere a conclusioni errate che potrebbero comportare conseguenze gravi sia per l'imputato, che per la squadra investigativa.

La ricostruzione temporale dei fatti, nell'ambito della digital forensics e dell'incident response, può coinvolgere gli eventi di un singolo sistema, così come gli eventi generati da diversi elaboratori, anche geograficamente lontani tra loro, ciascuno con un proprio orologio di sistema.

Come anticipato, una particolare tecnica utilizzata per la ricostruzione degli eventi è la "time-lining". Con questa tecnica gli eventi discreti, che hanno un timestamp associato, vengono ordinati in una sequenza temporale.

Un timestamp può essere ottenuto dai metadati di un file system, dai metadati[4] dei files, dai log files gestiti da un sistema operativo, da un dispositivo di rete, da un database e dalle informazioni registrate sui documenti o sulle banche dati.

A seconda della sorgente è possibile ottenere una o più sequenze dettagliate dei fatti che hanno avuto luogo in uno o più sistemi di elaborazione, permettendo all'investigatore, attraverso la sovrapposizione degli stessi, di ricostruire

[3] Corey, V., Peterman, C., Shearin, S., Greenberg, M. S., and Bokkelen, J. V. Network forensics analysis. IEEE Internet Computing 6, 6 (2002), 60–66.

[4] Un metadato (dal greco meta- "oltre, dopo" e dal latino datum "informazione" – plurale: data), letteralmente "dato su un (altro) dato", è l'informazione che descrive un insieme di dati. come ad esempio: lo Strumento utilizzato per la creazione dei dati, lo Scopo dei dati, l'Ora e data di creazione, il Creatore o autore dei dati, il Posizionamento su una rete di computer in cui è stato creato i dati, gli Standard utilizzati. Ad esempio, una immagine digitale può includere metadati che descrivono quanto è grande l'immagine, la profondità del colore, la risoluzione dell'immagine, quando è stata creata, e altri dati. I Metadati di un documento di testo possono contenere informazioni su quanto è lungo il documento, chi è l'autore, quando è stato scritto, e una breve sintesi del documento stesso.

l'avvenimento d'interesse.

Quando un timestamp registrato su un sistema di calcolo si considera utile per essere sostenuto quale elemento di prova di un'indagine forense, devono essere presi in considerazione alcuni fattori che potrebbero compromettere la qualità del risultato, vediamo i principali:

- il tempo su un sistema di calcolo è tenuto dall'orologio hardware del sistema e, in alcuni casi, da un ulteriore software di gestione del clock (p.e. i Server NTP[5]). A seconda della precisione di questi orologi, di come sono stati inizializzati e/o sincronizzati, l'orario può differire in maniera considerevole dal tempo "reale",
- l'orologio del sistema può essere mal configurato per essere nella time zone (fuso orario) sbagliata o essere impostato con un orario non corretto,
- l'orologio può essere manipolato arbitrariamente,
- inoltre, un orologio può essere eseguito più velocemente o lentamente rispetto al tempo standard (clock skew).

Analizziamo il caso in cui si debba effettuare una ricostruzione temporale forense il cui obiettivo è, semplicemente, quello di ordinare tutti gli eventi rilevati, in modo tale che le relazioni di causa-tempo possano essere collegate tra loro. Se tutti gli eventi di cui abbiamo bisogno per la ricostruzione sono "timestamped" dall'orologio dello stesso sistema, alcuni dei fattori di cui sopra potrebbero non inficiare il risultato atteso in quanto i "tempi" saranno diversi dal tempo "reale" in modo coerente e di conseguenza si potrà, comunque, risalire all'esatta sequenza degli eventi accaduti.

La possibilità che l'orario possa essere stato manipolato, tuttavia, è un fattore che deve essere, in ogni caso, preso in considerazione. In particolare, se l'orologio è stato impostato indietro nel tempo, potrebbe verificarsi il caso che, ad un determinato orario, possano essere stati registrati eventi verificatisi prima e dopo il cambio di clock, sfalsando l'ordine corretto delle registrazioni.

Occorre tener presente, inoltre, che pochi sistemi informatici sono

[5] Il Network Time Protocol, in sigla NTP, è un protocollo per sincronizzare gli orologi dei computer all'interno di una rete a commutazione di pacchetto quindi con tempi di latenza variabili ed inaffidabili. I diversi server NTP sono organizzati in una struttura gerarchica di "strati", dove lo strato 1 è sincronizzato con una fonte temporale esterna quale un orologio atomico, GPS o un orologio radiocontrollato, lo strato 2 riceve i dati temporali da server di strato 1, e così via. Un server si sincronizza confrontando il suo orologio con quello di diversi altri server di strato superiore o dello stesso strato. Questo permette di aumentare la precisione e di eliminare eventuali server scorretti. Un server NTP è in grado di stimare e compensare gli errori sistematici dell'orologio hardware di sistema, che normalmente è di scarsa qualità.

completamente isolati dalla rete e, di conseguenza, possono contenere dei timestamp introdotti o importati dall'esterno con modalità diverse a causa dell'inevitabile scambio di informazioni tra sistemi i cui clock possono essere non sincronizzati tra loro.

Per esempio le informazione contenute nell'header (intestazione) di un email o i dati di un cookie HTTP possono svolgere un ruolo importante in una indagine forense, per questo motivo i timestamp esterni ed interni devono essere obbligatoriamente correlati e confrontati tra loro. Inoltre, ogni qualvolta una prova digitale viene utilizzata per stabilire o sostenere il momento in cui determinati eventi si sono verificati nel mondo fisico, i timestamp dei sistemi coinvolti devono essere tradotti a questo tempo di riferimento.

In questo articolo presenteremo uno studio che mira a dare una migliore comprensione dei problemi che devono essere considerati ed affrontati quando ci si riferisce ai clock dei computer, illustrare alcune metodologie per la costruzione ed analisi delle timeline e tradurre il risultato di tale attività tecnica in un oggetto che abbia un valore probatorio.

3. IL PROBLEMA DELL' "ORA ESATTA"

Nelle indagini digitali forensi conoscere l'ora esatta in cui sono successi gli eventi può apparire di vitale importanza, perché potrebbe essere necessario correlare eventi provenienti da sistemi diversi, oppure correlare eventi digitali ad altri del mondo fisico[6].

L'importanza di conoscere l'ora esatta è stata già trattata da Boyd e Foster[7] e da Weil[8].

Gli errori che determinano questo fenomeno possono essere di tipo sistemico, nel caso in cui dipendono da sorgenti interne o esterne (p.e. l'effetto collaterale di un'operazione ordinaria); oppure di natura interventistica, nel caso in cui dipendono da una manomissione del sistema.

[6] E. Earl Eiland. Time Line Analysis in Digital Forensics, CS589, Digital Forensics, Fall 2006 - New Mexico Institute of Mining and Technology - Socorro, New Mexico, September 18, 2006.

[7] Boyd Chris, Foster Pete. Time and date issues in forensic computing – a case study. Digit Investig 2004;1(1):18–23.

[8] Weil Michael C. Dynamic time & date stamp analysis. Int J DigitEvid 2002;1(2).

Causa Sorgente dell'errore	SISTEMICA	INTERVENTISTICA
INTERNA	OROLOGIO IMPRECISO	MALCONTENTO DI UN DIPENDENTE
ESTERNA	RIFERIMENTO INESATTO	INTRUSO MALEVOLO

Idealmente vorremmo che i clock di tutti i computer fossero sincronizzati ad un certo orario di riferimento comune come l'Universal Coordinated Time (UTC)[9]. Ci sono provider su Internet che offrono questo servizio, uno di questi è il National Institute of Standards and Technology (NIST)[10].

Questi server sono precisi nell'ordine di $1x10^{-16}$ secondi. Ma anche se i server fossero tutti stabili, il problema si presenterebbe durante il passaggio dell'informazione da un dispositivo ad un altro. Per cui più il clock da sincronizzare si allontana dalla sorgente di riferimento, più diminuisce la qualità della misura ed aumenta il grado d'incertezza.

L'errore che più frequentemente si presenta nei casi di digital forensics è la sincronizzazione con una fonte non corretta o con un orologio impreciso. Inoltre, se le fonti esterne venissero aggiornate di rado, il problema della sincronizzazione si amplificherebbe ulteriormente. Sfortunatamente le fonti esterne difficilmente registrano queste sincronizzazioni, per cui diventa impossibile trovare un riferimento di scostamento.

Questo errore ha un impatto diretto sui timestamp generati dai sistemi localmente. Per cui, finché l'indagine si sviluppa su un solo dispositivo il

[9] United States Naval Observatory: Astronomical Applications Department. What is universal time?, <http://aa.usno.navy.mil/faq/docs/UT.html>; 2003.

[10] Il National Institute of Standards and Technology (NIST) è un'agenzia del governo degli Stati Uniti d'America che si occupa della gestione delle tecnologie. Fa parte nel Dipartimento del Commercio e il suo compito è la promozione dell'economia Americana attraverso il lavoro con l'industria per sviluppare standard, tecnologie e metodologie che favoriscano la produzione e il commercio.

problema è minimo, perché gli orari sono internamente coerenti tra loro; se, viceversa, l'indagine comporta il confronto di timeline provenienti da sistemi diversi, si configura il problema dell'ora esatta.

Un altro errore di tipo sistemico è dovuto dalla errata conversione delle date. Alcuni dispositivi utilizzano il Greenwich Mean Time (GMT), altri usano l'ora locale, che può essere legale o non, nei vari formati di rappresentazione e di conversione. Anche questo, se non viene preso in considerazione, può diventare un problema con conseguenze serie.

È anche possibile che alcuni orologi possano, intrinsecamente, andare leggermente più lenti o più veloci del tempo reale. Ciò può dipendere dalla qualità dei loro cristalli di quarzo o da alcuni fattori ambientali, quali la temperatura[11]. A causa di ciò, e dal fatto che possono essere facilmente impostati su valori arbitrari, è del tutto possibile che gli orologi dei computer non mostrino valori ovunque prossimi al tempo reale. Anzi, potrebbero presentare valori diversi a differenti momenti del tempo reale.

Un metodo per ovviare a tale difetto, ossia per tenerli agganciati al tempo reale, consiste nel sincronizzarli periodicamente con l'orario di un computer più preciso.

La precisione più elevata di questi server del tempo (Time Server) è realizzabile grazie all'utilizzo di hardware dedicato, come un ricevitore GPS o un orologio atomico, ed al protocollo più comunemente utilizzato per la sincronizzazione dell'orario in una rete di computer: il Network Time Protocol (NTP)[12].

Se tutti i computer sincronizzassero i loro orologi con un apposito time server, usando l'NTP, i casi forensi e la correlazione col tempo sarebbero facilmente risolvibili.

È, tuttavia, improbabile che tutti i computer eseguano questo tipo di sincronizzazione degli orologi. Inoltre, attualmente non è possibile rivelare quanti host sincronizzano i loro orologi con un Time Server attraverso Internet. Ci sono addirittura un gran numero di server su Internet che non sincronizzano mai i propri clock[13].

Tutto questo sottolinea ulteriormente l'importanza della ricerca per stabilire la correlazione temporale e le mappature ai fini delle indagini digitali forensi. Ciò diventa particolarmente rilevante se si considera l'analisi di fonti di prova

[11] Symmetricom: Timing, Test, and Measurement Division. Stochastic model estimation of network time variance, <http://www.symmttm.com/pdf/Network_Timing/wp_Stochastic_Model.pdf>.

[12] Mills D., "Network Time Protocol (version 3): specification, implementation and analysis. Technical Report RFC 1305", Network Working Group; March 1992.

[13] Florian Buchholz, Brett Tjaden. Digital Investigation 4S (2007) S 31 – S 42.

i cui timestamp sono stati registrati da orologi diversi.

Un metodo per tradurre i timestamp rilevati su un host rispetto all'UTC è stato sviluppato sia da Stevens[14] che da Buchholz[15]. In sintesi questi modelli cercano di "descrivere" il clock di un host rispetto ad un determinato tempo di riferimento.

Un altro criterio per misurare in modo affidabile il clock di un computer remoto è per mezzo della Rete. Questo sistema può essere sfruttato nelle indagini forensi perché spesso si trovano dei timestamp provenienti da fonti esterne al sistema. Alcuni timestamp sono inclusi nei dati memorizzati nella cache HTTP o nei cookie, nelle intestazioni delle e-mail ed in altri dati salvati durante le comunicazioni in rete. Se si potesse risalire all'orario di tale clock esterno, potrebbe essere considerata una valida fonte di prova indipendente ed essere utilizzata per stabilire o confermare un timestamp locale. Naturalmente è impossibile ottenere un'informazione dal clock di un host esterno per un lasso di tempo che si trova nel passato. Tuttavia, se la ricerca dovesse rivelare che gli orologi si comportano in modo prevedibile nel tempo, potrebbe essere possibile misurare l'host esterno in un secondo momento ed effettuare delle previsioni su come l'orologio si sia potuto comportare nel passato. Anche se ciò non provasse il modo di funzionamento di un orologio, se le previsioni dovessero essere coerenti con le evidenze locali, potrebbe essere utilizzato come fonte indipendente per aumentare l'attendibilità delle prove.

Misurare gli orologi dei computer remoti potrebbe essere utile anche per monitorare una macchina all'interno di una rete. Se una macchina dedicata mantenesse le informazioni dei clock degli altri computer della rete, avremmo ulteriori prove utili alle indagini forensi o per il rilevamento di intrusioni. In primo luogo per l'investigatore sarà disponibile una descrizione completa dei clock di tutte le macchine della rete. Analogamente ad un server di log esterno, le informazioni degli orologi potranno essere più affidabili di qualsiasi altra informazione presente sulla macchina compromessa. In secondo luogo un clock monitor potrà evidenziare un comportamento insolito dei tempi ai fini della rilevazione di un intrusione o di un uso improprio. Se un attaccante di un computer locale modificasse l'orologio di sistema per nascondere le sue tracce, potrebbe essere rilevato e generare un allarme.

Se volessimo controllare in maniera attendibile gli orologi dei computer remoti, dovremmo rispondere ai seguenti quesiti:

- Quale metodo possiamo utilizzare per misurare il tempo degli host

[14] Stevens Malcolm W. Unification of relative time frames for digital forensics. Digit Investig 2005;1(3):225–39.

[15] F. Buchholz, "An Improved Clock Model for Translating Timestamps", JMU-INFOSEC-TR-2007-001, James Madison University.

remoti su Internet?
- Che tipo di accuratezza possiamo ottenere?
- Quanti dati devono essere conservati?
- Guardando il comportamento dell'orologio osservato, cosa possiamo dedurre circa il suo comportamento nel passato?

In questo documento non verrà trattato questo aspetto, perché già approfondito negli articoli citati, ma è utile ricordare che è possibile indagare sui clock dei computer da remoto e che la correlazione dei timestamp è un problema che può verificarsi nella pratica forense.

4. METODOLOGIA

Abbiamo già spiegato perché l'analisi temporale è un strumento indispensabile per poter effettuare la ricostruzione dell'ordine delle operazioni di un utente in un sistema informatico e che, altresì, può rivelarsi utile per poter valutare l'attendibilità e l'usabilità delle informazioni digitali raccolte come elementi di prova.

Spesso, però, questo tipo di esame si sfrutta solo per ottenere una data o un orario, ma sarebbe utile invece ragionare su come possa essere sfruttato per rispondere ad altri quesiti, quali ad esempio:
- Chi ha inserito lì quel file ?
- Come ci è arrivato?
- Da dove proviene ?
- Dove è andato a finire?
- Perché è stato creato?
- Che cosa vuol dire?

Prima di passare all'analisi tecnica, del caso di studio preso in esame, è utile introdurre alcune considerazioni sistematiche sull'analisi forense, di reperti digitali, finalizzata alla ricostruzione storica dei fatti.

L'analisi del tempo applicato alle evidenze digitali può essere suddivisa in quattro classi differenti:
1. l'analisi dei timestamp presenti nei file system;
2. l'analisi dei timestamp contenuti all'interno dei file;
3. il riscontro con altri riferimenti temporali rilevabili;
4. la contestualizzazione dei timestamp.

4.1 Analisi dei timestamp presenti nei file system

Ogni file memorizzato su qualsiasi supporto digitale ha associati più timestamp che tengono traccia del suo tempo di "Creazione", "Ultima Modifica", "Ultimo Accesso" e "Ultima Modifica della Voce" (quest'ultima indicata la modifica della voce "Entry" sulla Master File Table - MFT). Questi quattro valori, che vengono comunemente indicati con l'acronimo "MACE", insieme ad altri contenuti in un MFT possono assurgere a valido elemento di prova in un procedimento giudiziario.

I valori delle voci di una MFT cambiano a seguito di determinate operazioni effettuate dagli utenti; di conseguenza a posteriori, analizzando i valori di dette voci, si possono ipotizzare le azioni compiute dagli utenti su gli stessi file.

Pertanto è intuitivo dedurre che servendosi di queste informazioni, con l'ausilio di tools specifici, si può ricostruire la sequenza di operazioni compiute da un operatore, o dal sistema stesso, sul file system di un host e le stesse si possono rappresentare sotto forma di un file di log – o cronologia - degli eventi.

Il limite di quest'analisi è strettamente connessa all'atomicità del file, ovvero non consente di analizzare operazioni complesse effettuate sullo stesso file, basti pensare ad una banca dati rappresentata da un singolo file che al suo interno riproduce milioni di operazioni.

Per esempio è possibile rivelare il momento di apertura di un documento, l'orario delle ultime modifiche, l'istante di creazione o di copia, di stampa, ecc.

In questa analisi non si devono dimenticare i file cancellati perché anch'essi, attraverso i timestamps associati, rappresentano eventi ed azioni di un sistema (per esempio i files temporanei spesso assumono un ruolo fondamentale per verificare l'attendibilità delle prove digitali) ed aiutano a tracciare un quadro completo dei fatti.

4.2 Analisi dei timestamp contenuti all'interno dei file

Tutti i sistemi di elaborazione prevedono la creazione di determinati file che tengono traccia della cronologia delle operazioni eseguite per motivi di sicurezza, di auditing o per prescrizioni normative. Per esempio i file di log degli eventi di sistema, i log delle connessioni di rete e di autenticazione, i file di journal, i file di registro, i redolog file dei database, per finire con la miriade di timestamp contenuti sulle banche dati degli applicativi gestionali.

L'obiettivo principale, per cui sono stati progettati, era fornire uno strumento

che fosse in grado di storicizzare la sequenza di modifiche ed inserimenti su un determinato sistema di calcolo e, nel contempo, verificare il corretto funzionamento delle procedure. Successivamente, con la diffusione dei primi attacchi informatici, gli stessi si sono rivelati utili per finalità legate alla sicurezza informatica, perché consentono di rivelare eventuali comportamenti malevoli, e come elementi di prova per sostenere determinati eventi delittuosi.

Questi file sono spesso visibili in chiaro e quindi facilmente interpretabili, altre, invece, essendo difformi dai formati standard di riferimento, per poter essere letti richiedono software dedicati o proprietari.

La loro presenza è indubbiamente rilevante nel contesto di un indagine digitale, in particolar modo per il tipo di informazione che rappresentano.

Il vantaggio principale, che agevola l'attività investigativa, è dato dal fatto che spesso possono essere utilizzati direttamente come fonte di prova. Viceversa, la difficoltà che si incontra nell'analisi deriva dalla mancanza di software universali in grado di leggere e comparare qualsiasi file di log. Da qualche anno, anche grazie all'introduzione di alcune previsioni normative che li rendono obbligatori, alcuni di questi file stanno adottando formati standard.

Nelle grandi e medie realtà elaborative vi è, fortunatamente, una produzione sovrabbondante di questi file, per cui l'analisi è facilmente perseguibile; al contrario nelle piccole realtà o in contesi domestici queste informazioni sono assenti o insufficienti per essere utilizzate come elementi di prova.

I file di registro di Windows riproducono, per alcuni versi, un tipo particolare di log file perché registrano varie sequenze di eventi che accadono su un sistema come: l'avvio e lo spegnimento del sistema stesso, l'esecuzione di determinati programmi o servizi, l'installazione delle periferiche, l'apertura dei file non presenti sul file system, la cronologia di navigazione in Internet, ecc.; pertanto è necessario tenerne particolarmente conto nella analisi temporale di un sistema basato su questo sistema operativo.

Questo tipo di analisi comporta tre fasi:

a. La raccolta dei file di log e di registro,
b. La normalizzazione delle informazioni contenute con i relativi riferimenti temporali,
c. La sovrapposizione delle timeline risultanti, compresa quella estratta dal file system, per costruire una supertimeline.

4.3 Riscontro con altri riferimenti temporali rilevabili

Questo tipo di analisi riguarda essenzialmente il riscontro di tutte le altre informazioni temporali contenute o reperite all'interno dei sistemi informativi.

Ricordiamo che sono costituiti da componenti hardware, da software di sistema, da applicativi e dai dati.

Si fa riferimento per esempio:

- alla verifica della data di rilascio del software utilizzato rispetto alla data di creazione di un file,
- al confronto tra la versione della definizione dei virus ed il log degli eventi di sistema,
- al riscontro tra il programma che ha generato un file e quelli installati o disinstallati sulla macchina presa in esame,
- al controllo della data di fabbricazione o alla versione dei dispositivi hardware rispetto alle informazioni temporali contenute,
- alla comparazione tra i riferimenti temporali contenuti all'interno di un file (testo, immagine, audio o video) e le stesse informazioni ricavate dal contesto reale.

Questo argomento non verrà trattato in questo articolo perché rientra tra le attività investigative che non necessitano di analisi strumentali, ma ha senso che venga citato perché rientra comunque tra le verifiche che un investigatore digitale deve eseguire per rendere attendibili le prove oggetto di analisi (per esempio risulta indispensabile per verificare la bontà di un alibi informatico).

4.4 Contestualizzazione dei timestamp

Infine si accenna all'attività di analisi che riguarda la contestualizzazione delle informazioni temporali estratte dai sistemi digitali, rispetto a fatti rinvenibili in altri ambienti.

Un'indagine digitale scaturisce spesso da un indagine investigativa tradizionale e serve a supportare o suffragare determinati fatti. E' fondamentale, pertanto, cercare di contestualizzare le operazioni digitali con le azioni che sono scaturite da un evento.

L'unione dei due contesti investigativi agevola la comprensione dei fatti e restringe il ventaglio di incertezza che spesso scaturisce nei casi complessi.

Anche questa attività non verrà trattata in questo studio, perché rientra nella sfera di competenze delle investigazioni tradizionali, ma è utile tenerne conto per sottolineare che si ottengono risultati eccellenti se la squadra investigativa collabora con i tecnici e viceversa.

5. CASE STUDY: CREAZIONE ED ANALISI DELLA TIMELINE

Il focus dell'analisi della timeline tradizionale è quello di estrarre ed analizzare i timestamp presenti nei file system che contengono informazioni digitali. Anche se non tutti i file system memorizzano le stesse informazioni, la maggior parte ha qualche timestamp in comune, come le informazioni sull'ultimo accesso e la data di ultima modifica. Alcuni file system registrano anche le informazioni relative alla cancellazione, alla creazione di un file ed alla modifica dei metadati[16].

Questa analisi, anche se estremamente utile, non è sufficiente perché non tiene conto, in maniera integrata, delle registrazioni presenti all'interno dei file o dei registri di sistema. Di conseguenza, l'investigatore che utilizza unicamente questa funzionalità, ignora alcune informazioni di contesto essenziali ad ottenere una descrizione completa e precisa degli eventi che hanno avuto luogo.

Per raggiungere questo obiettivo è necessario effettuare un'analisi più approfondita ed integrare la timeline del file system con le informazioni estratte dai log file e da altri reperti digitali che contengono timestamps. In altre parole occorre creare una sorta di supertimeline. E' utile sottolineare che questi file potrebbero risiedere sia sul sistema oggetto dell'analisi, che su altri dispositivi esterni, come per esempio un firewall o un proxy.

5.1 I problemi dell'analisi tradizionale

Come già detto in precedenza, sebbene una timeline tradizionale, ovvero basata esclusivamente sui timestamp del file system, sia in grado di fornire all'investigatore utili informazioni sugli eventi che hanno avuto luogo su un disco sospetto, non è esente da alcuni problemi.

Uno di questi è causato dalla frequente modifica dei timestamp durante l'esecuzione "normale" di alcuni servizi o del sistema operativo. Per esempio l'attività di scansione del disco rigido effettuata da un programma antivirus potrebbe aggiornare l'ora di ultimo accesso dei file.

Diversamente alcuni sistemi operativi moderni, per aumentare le prestazioni del file system, non aggiornano la data di ultimo accesso.

È evidente che in entrambi i casi si avrà un notevole decadimento della qualità delle informazioni che possono essere estratte dai timestamp dei files.

[16] I metadati possono essere considerati come "dati sui dati", ovvero in altre parole i dati che descrivono o aggiungono altre informazioni al contenuto del file. Nel contesto di un file system queste informazioni contengono il nome del file, la posizione dei blocchi all'interno del file system, così come le informazioni sulla cartella principale di memorizzazione.

Un altro effetto negativo che si ottiene, basandosi unicamente sui timestamp del file system, è l'assenza della cronologia delle modifiche di un determinato file, che può essere descritta solo all'interno di un file di log o all'interno dei metadati dello stesso file.

Le marche temporali, in genere, condividono lo stesso problema che si riscontra nella natura volatile delle prove. Nel mondo digitale spesso viene memorizzata solo l'ultima occorrenza di un particolare evento. Questo non rappresenta un problema se l'investigatore arriva sulla scena del crimine subito dopo il presunto reato o evento di interesse. Tuttavia, col passare del tempo aumenta la probabilità che alcuni timestamp, rilevanti per l'indagine, vengano soprascritti, rendendo l'analisi inaffidabile o addirittura fuorviante.

Per ovviare in parte a quest'ultimo problema è utile inserire nell'analisi anche i timestamp dei file cancellati.[17]

5.2 Anti-forensics

Il fatto che la data e l'ora siano memorizzate su un supporto digitale crea un ulteriore problema. È notorio che tutti i dati digitali, compresa la data e l'ora, possono essere alterati arbitrariamente. Questo rende l'analisi della timeline vulnerabile a questo tipo di attacco. Ci sono tantissimi strumenti di anti-forensics creati appositamente per modificare i timestamp in modo da ingannare o indurre in errore gli investigatori digitali.[18]

Altre tecniche di anti-forensics consentono la modifica dei valori di identificazione di alcuni file per ingannare gli strumenti di forensics, oppure inseriscono false informazioni nei log file per mandare in crash i software d'analisi.

In sintesi le opportunità offerte dall'anti-forensics obbligano l'investigatore ad analizzare ed incrociare tutte le informazioni raccolte, anche da fonti diverse, per verificare i risultati ottenuti e rilevare eventuali incongruenze o irregolarità. Così come è importantissimo avere un'ottima intuizione e formazione per individuare le anomalie che solitamente queste tecniche possono introdurre nei sistemi.

[17] Liu Zhi jun, Zhang Huan guo (2007): "Time Bounding Event Reasoning in Computer Forensic".

[18] Vincent, Liu (2005). Timestomp. Tool's web site: http://www.metasploit.com/research/projects/antiforensics/

5.3 Estensione delle timeline

Una delle possibili soluzioni per rimuovere i difetti presenti nell'analisi tradizionale consiste nell'estensione dell'indagine ad altre informazioni, provenienti da fonti diverse, per ottenere un quadro probatorio più completo, nonché per ridurre al minimo l'impatto delle tecniche di anti-forensics. Per esempio si potrebbero introdurre nella timeline i timestamp provenienti da fonti dove non è possibile che possano essere modificati.

Attualmente solo pochi toolkit di analisi forense[19] hanno incluso un software in grado di generare queste supertimeline. Il motivo potrebbe risiedere nel fatto che la gestione delle informazioni, provenienti da fonti diverse, è molto complessa, oppure perché non è ancora stata compresa l'importanza di avere questo tipo di risultato. Infatti i timestamp vengono memorizzati in vari formati differenti e questo rende molto laboriosa l'estrazione e l'analisi con un unico strumento. Inoltre, uno strumento che sia in grado di estrarre ogni timestamp, deve analizzare diversi tipi di file, sia binari che testuali, comprendere i diversi formati utilizzati e tradurre il risultato in un unico formato in grado di rappresentarli simultaneamente.

Molti tipi di file includono metadati che descrivono sia il contenuto del documento, che le azioni effettuate su di esso (p.e. l'ultima volta che il documento è stato stampato, modificato e da chi). Si comprende, perciò, il grado di rilevanza che queste informazioni possono assumere nel corso di un'indagine. Per esempio ricavare la data di stampa di un documento o l'autore che ha effettuato le ultime modifiche, potrebbe diventare la prova per risolvere un caso.

Anche i file di registro del sistema operativo Windows includono informazioni utili per l'analisi della timeline, p.e. l'ultimo momento di scrittura delle chiavi di registro nella UserAssist, le voci che contengono le informazioni sull'uso del computer di un determinato utente.

Ci sono tanti altri file che contengono informazioni importanti di timestamp: la cronologia del browser, i log dell'instant-messaging, i log dell'anti virus, le e-mail, ecc.

La creazione di una supertimeline[20] consente di abbreviare notevolmente i tempi di un'indagine. Analizzando la supertimeline l'investigatore ottiene immediatamente un'ottima panoramica del disco sospetto ed ha la possibilità di trovare rapidamente le prove utili che necessitano di ulteriori analisi.

[19] DEFT 6 di Stefano Fratepietro www.deftlinux.net

[20] Il termine "supertimeline" viene utilizzato, dal 2010, per indicare una timeline arricchita di elementi aggiuntivi rispetto ai tradizionali "filesystem timestamp", estratti da log, metadati, browser history, etc.

Non ci sono riferimenti che dimostrano, in maniera esplicita, il vantaggio di utilizzare una supertimeline rispetto ad un'analisi tradizionale, ma, tuttavia, il questionario condotto da Olsson e Boldt riguardo all'utilizzo del loro strumento CyberForensicsTimeLab ha dimostrato che il loro test di prova è stato risolto molto più velocemente rispetto al tempo impiegato con un prodotto tradizione, vale a dire in 14 minuti contro 45 minuti. [21]

La quantità di dati raccolti in una supertimeline può rappresentare un ulteriore carico di lavoro soprattutto se, come nella stragrande maggioranza dei casi, le voci di interesse non sono circoscritte ad un determinato evento. È come cercare un ago in un pagliaio. Per questo è utile avere a disposizione uno strumento che sia in grado di estrarre facilmente tutti i dati temporali e che sia, altresì, capace di ridurli per accelerare l'indagine.

L'analisi della Timeline è, ed è sempre stata, un processo molto manuale dal momento che ogni indagine è solitamente completamente diversa da quella precedente. Cioè le voci rilevanti non sono quasi mai uguali in due diverse indagini, altresì può capitare che voci del tutto irrilevanti per un indagine, possono essere decisive in un altro caso. Questo problema ha sempre scoraggiato la creazione di un unico strumento in grado di ridurre il set dei dati in maniera automatica ed ha costretto la gestione manuale della timeline per ogni caso. Pertanto vi è la necessità di disporre di uno strumento che sia in grado di eliminare alcune voci statiche o irrilevanti dalla timeline, che disponga di un filtro per eliminare facilmente quelle voci che non fanno parte dell'inchiesta e che sia dotato di un potente motore di ricerca per recuperare quelle che invece potrebbero essere rilevanti. Lo strumento dovrebbe avere, inoltre, la capacità di trovare un evento a partire da una lista di parole, oltre ad essere in grado di limitare la ricerca ad un determinato intervallo temporale. Infine, sarebbe utile poter disporre di una rappresentazione visuale della timeline per facilitare la lettura della stessa e rendere i report più interessanti e comprensibili sia al ricercatore che alle persone non tecniche che spesso sono destinatari delle perizie forensi.

5.4 Impostazione del case study

Il nostro caso di studio sarà basato sull'analisi di un'immagine di un disco su cui è installato un sistema Windows XP SP3 con due utenti, "domexuser1" e "domexuser2". Affinché le prove siano ripetibili anche da chi legge, è stata

[21] Olsson, Boldt, J., M. (2009). "Computer forensic timeline visualization tool. Digital investigation", 6, 78-87.

scelta un'immagine del repository "Digital Corpora[22]" creato da Simson L. Garfinkel (l'autore delle librerie AffLib e dell'utilissimo tool bulk_extractor). L'immagine "domexusers" è distribuita[23] in diversi format, raw, aff ed ewf: di tutti e tre verrà illustrata la modalità di montaggio sul filesystem per l'analisi. Per comodità, si consiglia di scaricare dal sito l'immagine in formato compresso ewf o aff, poiché quella raw (non compressa) occupa circa 40GB contro i circa 4GB di ognuna delle altre due.

5.5 I tool per la generazione di timeline

Prima di inoltrarci nel vasto mondo delle supertimeline, vediamo quali strumenti sono a nostra disposizione per la realizzazione di tradizionali timeline del filesystem. Sostanzialmente, almeno nel caso di filesystem NTFS che è il più diffuso, la generazione di una timeline standard avviene tramite parsificazione dei record MFT nella tabella attiva. Diversi tool, gratuiti, commerciali, open source o closed source, sono in grado di eseguire questo semplice task, pur presentando alcuni limiti nell'immediatezza o nella compatibilità con diversi tipi di immagini. Pochi tool offrono direttamente l'opzione "timeline", ma attraverso il listing o il parsing dei record MFT è possibile generare degli elenchi ordinabili di file che possono essere paragonabili a regolari timeline.

Ecco un breve elenco di alcuni dei principali strumenti che possono essere utilizzati per generare tradizionali timeline del filesystem:

- Fls, di Brian Carrier, parte della suite TSK [www.sleuthkit.org/]
- FTK Imager, di AccessData [accessdata.com/support/adownloads#FTKImager]
- NTFSwalk, di TzWorks [www.tzworks.net/prototype_page.php?proto_id=12]
- AnalyzeMFT, di David Kovar [www.integriography.com/]
- mft.pl, di Harlan Karvey [code.google.com/p/winforensicaanalysis]
- MFTView, della Sanderson Forensics [www.sandersonforensics.com]
- Encase [www.guidancesoftware.com/]
- X-Ways Forensics [www.x-ways.net/forensics]

Il più utilizzato, per la sua versatilità e perché storicamente inserito nel Live CD Helix[24], è il tool "fls" di Brian Carrier, inserito nella suite The Sleuth Kit.

[22] http://digitalcorpora.org
[23] http://digitalcorpora.org/corp/images/nps/nps-2009-domexusers/
[24] http://www.e-fense.com/helix/

Tramite la comoda interfaccia grafica web "Autopsy", che fa uso delle librerie e dei tool TSK, chiunque è in grado di creare un caso, aggiungervi un'immagine o un device fisico e creare un'ottima timeline del filesystem. Tutto questo senza invischiarsi in particolari come offset, fuse o loop device ed ottenendo anche alcune statistiche sul timestamping giornaliero e sull'orario dei file del sistema.

5.6 I tool per la generazione di supertimeline

Dopo aver fatto una panoramica degli strumenti a nostra disposizione per poter creare tradizionali timeline di filesystem, vediamo cosa e dove ha portato la necessità di poter ampliare il loro raggio di azione.

Sono stati prodotti, nel corso degli ultimi anni, diversi piccoli tool o script autonomi in grado di estrarre singoli tipi di artefatti e timestamp (registro di sistema, dati exif, log della navigazione web, etc…), che non presentano una visione globale del sistema, ma richiedono spesso parecchio lavoro manuale e rendono difficile il lavoro di integrazione dei dati.

La visione globale delle timeline generate, a partire dai più disparati artefatti, ha cominciato a prendere forma con la creazione, da parte di Harlan Carvey[25], del formato TLN, nato proprio per poter fungere da punto di contatto dei diversi strumenti di timelining. Tale formato non ha avuto il successo sperato (pur essendo tuttora mantenuto) ed i tool che ne fanno uso sono in numero troppo esiguo. Il formato mactime, nato molto prima ed utilizzato dai tool della suite TSK per la generazione dei bodyfile, si è dimostrato invece più adatto ad integrare diversi output. Diversi tool si sono adeguati per adottare, tra gli altri, anche il formato mactime in output.

I nuovi tool, compreso il nascente log2timeline, l'hanno utilizzato sin dall'inizio, rendendo necessaria una conversione del risultato finale in formato CSV per una più comoda analisi da parte dell'investigatore. L'ultima versione di log2timeline, la 0.60, ha cominciato a spingere verso l'utilizzo diretto del formato CSV anche per la fase di creazione ed integrazione. Parte dei tool nati nel tempo si sono evoluti e sono mantenuti tutt'ora. Log2timeline, al momento, sembra essere il più quotato ed il più completo nella comunità della Computer Forensics, ma non è l'unico strumento in grado di produrre timeline complete ed integrate di un sistema o di una immagine.

[25] http://windowsir.blogspot.com, autore di diversi volumi di computer forensics tra i quali spiccano "Windows Forensics Analysis" e "Windows Registry Forensics".

5.7 Log2Timeline e le supertimeline

L'insieme di script raccolti sotto il nome di "log2timeline", scritti in Perl dal ricercatore Kristinn Gudjonsson[26] nel 2009 ed a tutt'oggi mantenuti, viene incontro alle necessità presentate da chi vuole raccogliere informazioni che vanno oltre la "semplice" timeline del filesystem.

Sostanzialmente, con pochi comandi è possibile ottenere una timeline che raccoglie decine di timestamp di origine diversa, ordinati e rappresentati secondo una struttura comune. Avremo, ad esempio, i timestamp del filesystem inframmezzati con quelli relativi alla navigazione Internet, ai dati exif delle fotografie, ai metadati dei documenti Office, delle liste di prefetch, dei Recent, degli eventi di sistema di Windows, delle informazioni estratte dal registro quali dispositivi USB utilizzati, programmi avviati, file aperti e tanto altro ancora.

Sviluppato inizialmente in Perl per Linux, ma ora compatibile anche con Mac OS X (10.5.7+) e Windows, il progetto ha subito diverse evoluzioni. Da iniziale "parser" di artefatti, che richiedeva parecchia manualità e strumenti accessori, è arrivato, con la recentissima versione 0.60 pubblicata il 6 giugno 2011, ad essere un maturo e completo framework modulare basato su un nuovo engine, tutto sotto licenza GPL v3. Oltre all'engine di parsificazione dei timestamp (log2timeline), è stato introdotto uno script (l2t_process) per elaborare i risultati eliminando duplicati, filtrarli in base a parole chiave ed intervalli temporali ed eseguire, nel contempo, un'analisi sulle possibili compromissioni ai metadati NTFS. Sono stati altresì creati nuovi ed interessanti plugin, quali quello per parsificare alcuni log di Skype, oppure quello per analizzare direttamente i record MFT (e valutare così l'esistenza di possibili inconsistenze tra i timestamp contenuti in $STDInfo e $Filename), sono stati introdotti moduli per estrarre direttamente i dati temporali dal registro e molte altre novità.

Il framework log2timeline è organizzato in quattro moduli indipendenti: front-end, librerie condivise, modulo di input e modulo di output. I front-end disponibili erano in origine tre:

- *log2timeline*: tool da linea di comando che parsifica i file utilizzando un singolo modulo alla volta (l'ultima versione di log2timeline è in grado di esplorare ricorsivamente file e cartelle utilizzando i moduli di volta in volta necessari rendendo sostanzialmente superfluo il terzo front-end "timescanner");
- *Glog2timeline*: interfaccia grafica per il tool log2timeline;
- *timescanner*: tool da linea di comando che parsifica ricorsivamente diversi file ed acquisisce artefatti eterogenei parsificati da diversi

[26] http://log2timeline.net/

moduli, scelti di volta in volta, in base al tipo di file ed al tipo di timestamp in esso contenuto.

Nell'ultima versione del framework (0.60), l'interfaccia grafica è stata abbandonata (è in programma una riscrittura del codice) ed il tool "timescanner" di applicazione ricorsiva del parser reso superfluo. Sostanzialmente dalla versione 0.60 è possibile utilizzare il tool "log2timeline" direttamente sul filesystem, lasciando che vengano scelti di volta in volta i moduli necessari per parsificare i diversi file incontrati. E' possibile, comunque, impostare delle categorie di moduli da utilizzare in base al tipo di sistema in analisi (Linux, WinXP, Win7, etc...) per facilitare il lavoro del tool ed evitare di utilizzare moduli non adatti. Sempre nella versione 0.60 è stata inserita una funzionalità di preprocessing che analizza i file contenuti nella directory sottoposta all'analisi per cercare le informazioni sul sistema operativo installato, la timezone configurata, l'hostname, etc…

Come accennato in precedenza, esistono tool che analizzano e presentano singolarmente i suddetti timestamp, ma non ne esistono che riescano a raccoglierli in un unico insieme valutabile dall'investigatore. Vediamo in dettaglio, nella tabella seguente, quali sono i tipi di timestamp che l'ultima versione di log2timeline è in grado di elaborare grazie ai suoi moduli di input.

Name	Description	Name	Description
apache2_access	Apache2 access log file	oxml	Parse the content of an OpenXML document (Office 2007 documents)
apache2_error	Apache2 error log file	pcap	PCAP file
chrome	Chrome history file	pdf	Parse some of the available PDF document metadata
encase_dirlisting	CSV file that is exported from FTK Imager (dirlisting)	prefetch	Parse the content of the Prefetch directory
evt	Windows 2k/XP/2k3 Event Log	recycler	Parse the content of the recycle bin directory
evtx	Windows Event Log File (EVTX)	restore	Parse the content of the restore point directory
exif	Extract metadata information from files using ExifTool	safari	Parse the contents of a Safari History.plist file
ff_bookmark	Firefox bookmark file	sam	Parses the SAM registry file
firefox2	Firefox 2 browser history	security	Parses the SECURITY registry file
firefox3	Firefox 3 history file	setupapi	Parse the content of the SetupAPI log file in Windows XP
ftk_dirlisting	CSV file that is exported from FTK Imager (dirlisting)	skype_sql	Skype database
generic_linux	Parse content of Generic Linux logs that start with MMM DD HH:MM:SS	software	Parses the SOFTWARE registry file
iehistory	Parse the content of an index.dat file containg IE history	sol	.sol (LSO) or a Flash cookie file
iis	IIS W3C log file	squid	Squid access log (http_emulate off)
isatxt	ISA text export log file	syslog	Linux Syslog log file
jp_ntfs_change	CSV output file from JP (NTFS Change log)	system	Parses the SYSTEM registry file
mactime	body file in the mactime format	tln	body file in the TLN format
mcafee	log file	userassist	Parses the NTUSER.DAT registry file
mft	NTFS MFT file	volatility	Volatility output files
mssql_errlog	Parse the content of an ERRORLOG file produced by MS SQL server	win_link	Windows shortcut file (or a link file)
ntuser	Parses the NTUSER.DAT registry file	wmiprov	Parse the content of the wmiprov log file

I moduli in input possono essere utilizzati, tramite il parametro "-f", in gruppi basati sostanzialmente sul tipo di sistema in analisi e sul grado di dettaglio che si vuole ottenere. In particolare i gruppi utilizzabili nella versione 0.60 di log2timeline sono rappresentati nella tabella seguente.

List Name	Modules Included
linux	apache2_access, apache2_error, pcap, syslog, generic_linux,
web	chrome, firefox3, firefox2, ff_bookmark, opera, iehistory, iis, safari,
webhist	chrome, firefox3, firefox2, ff_bookmark, opera, iehistory, iis, safari, sol,
win7	chrome, evt, exif, ff_bookmark, firefox3, iehistory, iis, mcafee, opera, oxml, pdf, prefetch, recycler, restore, sol, win_link, xpfirewall, wmiprov, ntuser, software, system,
win7_no_reg	chrome, evt, exif, ff_bookmark, firefox3, iehistory, iis, mcafee, opera, oxml, pdf, prefetch, recycler, restore, sol, ntuser, win_link, xpfirewall, wmiprov,
winvista	chrome, evt, exif, ff_bookmark, firefox3, iehistory, iis, mcafee, opera, oxml, pdf, prefetch, recycler, restore, sol, userassist, win_link, xpfirewall, wmiprov,
winxp	chrome, evt, exif, ff_bookmark, firefox3, iehistory, iis, mcafee, opera, oxml, pdf, prefetch, recycler, restore, setupapi, sol, win_link, xpfirewall, wmiprov, ntuser, software, system,
winxp_no_reg	chrome, evt, exif, ff_bookmark, firefox3, iehistory, iis, mcafee, opera, oxml, pdf, prefetch, recycler, restore, setupapi, sol, ntuser, win_link, xpfirewall, wmiprov,

L'utente può impostare la modalità di rappresentazione della timeline, prodotta dal tool, scegliendo tra i formati illustrati nella tabella seguente.

Name	Description
beedocs	tab-delimited file to import into BeeDocs
cef	ArcSight Commen Event Format (CEF)
cftl	Output timeline in a XML format that can be read by CFTL
csv	CSV (Comma Separated Value) file
mactime	mactime format
mactime_l	legacy version of the mactime format (version 1.x and 2.x)
simile	Output timeline in a XML format that can be read by a SIMILE widget
sqlite	Output timeline into a SQLite database
tab	TDV (Tab Delimited Value) file
tln	H. Carvey's TLN format
tlnx	H. Carvey's TLN format in XML

5.8 Ambiente di lavoro

Come si è detto, utilizzeremo il tool log2timeline, scaricabile liberamente dal sito http://log2timeline.net, per generare le supertimeline. Sono disponibili diverse modalità di installazione per diversi sistemi operativi. Si va dalla compilazione del sorgente, ai repository per Debian, Ubuntu, Fedora, OpenBSD e Max OS X tramite macports. Nel file "INSTALL.txt", contenuto nello zip scaricabile dal sito, oltre alle precedenti modalità di installazione, è illustrato un procedimento per poter installare log2timeline anche in Windows utilizzando ActiveState Perl.

Per chi non avesse tempo per compilare, o anche soltanto per installare, sono disponibili due distribuzioni Live CD forensi che contengono il framework log2timeline preinstallato e perfettamente funzionante con alcuni tool accessori che verranno utilizzati durante il presente case study. La prima, per la quale parteggiamo perché chi scrive collabora con il team di Stefano Fratepietro, è DEFT Linux[27], mentre la seconda è SIFT Workstation[28] di Rob Lee distribuita dal SANS Institute.

Con entrambe le distribuzioni si può lavorare, per comodità, in ambiente Windows tramite macchina virtuale e condividere una directory del sistema host (ad esempio quella con le immagini da analizzare) con la macchina virtuale guest. VmWare Player e VirtualBox sono due ottimi prodotti gratuiti che permettono di lanciare l'immagine dei CDROM in pochi minuti ed avere, quindi, a disposizione tutti gli ottimi strumenti in essi contenuti. SIFT Workstation è scaricabile direttamente anche in versione VmWare, così da poter essere lanciato ed utilizzato direttamente sui casi contenuti nel proprio ambiente Windows, tramite le directory condivise ed i VmWare Tools.

5.9 "Montaggio" dell'immagine forense

Il tool log2timeline non può essere lanciato direttamente sull'immagine del disco da analizzare[29], sia essa RAW, EWF o AFF. Il parsing avviene sfogliando le directory ed i file di un filesystem montato, anche in sola lettura, che deve

[27] http://www.deftlinux.net/

[28] http://computer-forensics.sans.org/community/downloads

[29] Kristinn Gudjonsson ha scritto, appositamente per l'inserimento in SIFT Workstation, un front-end di log2timeline chiamato "log2timeline-sift" che dovrebbe permettere di passare come parametro direttamente un'immagine del disco o di una partizione in esso contenuta. Lo script è contenuto, oltre che nel Live CD SIFT Workstation, nella cartella "extra" dello zip dei sorgenti scaricabile dal sito http://log2timeline.net.

essere reso disponibile e liberamente accessibile in una directory del filesystem locale, sia esso Live oppure un'installazione.

Se si sta lavorando su un sistema reale (ad esempio si sta realizzando una timeline di un PC durante operazioni sul campo tramite Live CD) basta montare in read-only le partizioni di interesse e lanciare lo script log2timeline. In genere la partizione contenente la maggior parte degli timestamp è quella di sistema, le partizioni dati contengono timestamp di artefatti interessanti (exif, pdf, doc, etc...), ma mancano le grosse fonti di informazione come il registro, le history dei browser, i recent, i prefetch, etc...

Quando, come nel case study in esame, si ha a che fare con immagini disco, è necessario identificarne le partizioni e montare il contenuto in una directory del filesystem locale. A differenza del mounting dei filesystem di un disco locale, è necessario utilizzare le loop devices[30] (perché stiamo montando una parte di un file, non un disco/device fisico) affinché vengano creati dei device virtuali da utilizzare per il mount. Il comando per creare loop device è "losetup", ma le implementazioni del "mount" di Windows prevedono un parametro che lo rende superfluo integrandolo come parametro "loop". Poiché log2timeline può far uso di file riservati del filesystem in analisi, cioè metadati che esistono sul filesystem, ma non sono utilizzati direttamente dal sistema operativo, come ad esempio $Boot, $LogFile, $BadClus, etc..., è necessario che il mount della partizione li renda "visibili". A tal fine il mount sarà eseguito utilizzando l'opzione "show_sys_files" che mostra, oltre ai file presenti nella partizione, anche tutti i file/metadati riservati al filesystem stesso.

Il primo problema da affrontare è il tipo di compressione (e quindi di formato) con il quale è stata acquisita l'immagine del disco. Il caso più semplice, ed è quello cui dobbiamo ricondurci anche nel gestire i casi più complessi, è quello in cui l'immagine sia stata acquisita nel formato "RAW", cioé senza alcuna compressione, tramite una copia bit-a-bit di tutti i settori del disco. Avremo quindi in questo caso il file immagine "nps-2009-domexusers.dd" (a volte viene usata l'estensione ".raw") del quale dovremo soltanto identificare le partizioni e montarle utilizzando gli offset corretti.

Nel caso in cui, invece, l'immagine sia stata acquisita nel formato Encase EWF, è necessario un passaggio intermedio per "virtualizzare" l'immagine "traducendola" in tempo reale in un'immagine raw. Escludiamo la conversione reale da EWF in RAW in quanto, seppur possibile, richiede tempi biblici, aumenta notevolmente lo spazio occupato e può tranquillamente essere evitata con una sorta di "mount" che ne virtualizzi il contenuto. Tale procedura può essere eseguita, nel caso del formato EWF, tramite due tool equivalenti,

[30] http://it.wikipedia.org/wiki/Loop_device

entrambi contenuti nei Live CD DEFT e SIFT: mount_ewf.py e xmount. Che si usi il primo o il secondo, è necessario creare una cartella che definiremo come "/mnt/raw" nella quale verrà montato il filesystem che si presenterà come "raw" pur essendo basato su un "ewf".

Il tool xmount[31], che fa uso del filesystem virtuale "fuse" per poter "astrarre" in tempo reale un formato in un altro, viene lanciato nel modo seguente:

```
# xmount --in ewf --out dd nps-2009-domexusers.E* /mnt/raw
```

Quando si utilizza xmount è necessario specificare l'estensione utilizzando i wildcard ('*' o '?'): non basta come per gli altri tool inserire soltanto il primo "chunk" dei file ewf (cioè il file che termina con estensione "E01"). Nel caso specifico non ci interessa poter scrivere sul filesystem, quindi non abiliteremo la cache che, tramite il tool xmount, permetterebbe di poter abilitare la scrittura sull'immagine, senza comprometterne però l'integrità e mantenendo perciò le modifiche in un file separato.

Dopo aver eseguito il comando sopra riportato, avremo nella directory /mnt/raw due file: nps-2009-domexusers.dd e nps-2009-domexusers.info.

```
# ls -al /mnt/raw/
total 4
drwxrwxrwx 2 root root 0 1970-01-01 00:00 .
drwxr-xr-x 12 root root 4096 2011-09-03 09:47 ..
-r--r--r-- 1 root root 42949672960 1970-01-01 00:00 nps-2009-domexusers.dd
-r--r--r-- 1 root root 228 1970-01-01 00:00 nps-2009-domexusers.info
```

[31] Xmount è un tool scritto da Gillen Dan, compatibile con Linux e Mac Os X, reperibile all'indirizzo https://www.pinguin.lu/index.php, in grado di convertire on-the-fly filesystem in diversi formati, creando un filesystem virtuale basato su FUSE (Filesystem in Userspace). La rappresentazione virtuale (cioè la conversione "virtuale" in output) può essere in raw DD, VirtualDisk di VirtualBox, VmWare VMDK. Il formato dd è tipicamente utilizzato quando si desidera convertire "al volo" un'immagine in formato raw per poterla montare più agevolmente sul filesystem. I due formati VirtualBox sono molto utili per poter creare on-the-fly un disco virtuale su cui lanciare una macchina virtuale – ad esempio per avviare in modalità virtuale il sistema operativo contenuto in un file immagine. Pe completezza, si riporta la possibilità di utilizzare il tool LiveView (http://liveview.sourceforge.net/) in Windows per poter creare una macchina virtuale VmWare a partire da un'immagine raw (oppure ewf o aff virtualizzata tramite il ben noto FTK Imager). Le immagini in input possono essere nei formati raw DD, EWF (Expert Witness Compression Format) o AFF (Advanced Forensic Format). Particolare molto interessante, xmount supporta un accesso "virtuale" in scrittura alle immagini montate, redirigendo le modifiche su un file di cache e ovviamente non sull'immagine stessa. Tale file di cache può essere mantenuto in modo da rendere persistenti le modifiche in sessioni successive, senza mai intaccare l'immagine originale.

Il primo conterrà la conversione in raw, in tempo reale, dell'immagine in formato EWF, mentre il secondo conterrà le informazioni ricavate dagli header ewf.

```
# cat /mnt/raw/nps-2009-domexusers.info
The following values have been extracted from the mounted image file:

Acquiry date: 2010-05-03T02:37:46
System date: 2010-05-03T02:37:46
Acquiry os: Linux
Acquiry sw version: 20100126
MD5 hash: 8e7176524a64376631cd7dc9d90339f1
```

Alternativamente al tool xmount è possibile, altresì, utilizzare lo script Python mount_ewf.py, che ha una funzione simile e fa anch'esso uso del file system virtuale "fuse" in questa maniera:

```
# mount_ewf.py nps-2009-domexusers.E01 /mnt/raw
```

Il risultato sarà lo stesso del commando precedente, con le estensioni dei file leggermente diverse, e ritroveremo due file nella directory /mnt/raw: nps-2009-domexusers e nps-2009-domexusers.txt.

```
# ls -al /mnt/raw/
total 41943045
dr-xr-xr-x 2 root root 0 1970-01-01 00:00 .
drwxr-xr-x 12 root root 4096 2011-09-03 09:47 ..
-r--r--r-- \ root root 42949672960 1970-01-01 00:00 nps-2009-domexusers
-r--r--r-- 1 root root 199 1970-01-01 00:00 nps-2009-domexusers.txt
```

Pur non avendo alcuna estensione, il primo file conterrà, come si evince dalla sua dimensione, tutta l'immagine. Ricordiamo che, anche in questo caso, il file /mnt/raw/nps-2009-domexusers è un file "virtuale", nel senso che non esiste realmente sul filesystem, ma viene "astratto" in tempo reale nel momento in cui si accede ad esso. Il file /mnt/raw/nps-2009-domexusers.txt contiene, come nel caso precedente, le informazioni relative agli header EWF (autore, data, note, etc...).

```
# cat /mnt/raw/nps-2009-domexusers.txt
# Acquiry date: 2010-05-03T02:36:52
# System date: 2010-05-03T02:36:52
# Operating system used: Linux
# Software version used: 20100126
8e7176524a64376631cd7dc9d90339f1 */mnt/raw/nps-2009-domexusers
```

Nel caso in cui volessimo lavorare sull'immagine in formato AFF potremmo fare riferimento al tool xmount, utilizzato in precedenza, con il seguente comando:

```
# xmount --in aff --out dd nps-2009-domexusers.aff /mnt/raw
```

In alternativa, la libreria AffLib (nata appunto per gestire il formato forense compresso AFF) mette a nostra disposizione il tool "affuse", che consente di creare un file virtuale avente la rappresentazione RAW DD della nostra immagine in input AFF.

```
# affuse nps-2009-domexusers.aff /mnt/raw
```

Per completezza riportiamo il caso in cui l'immagine a nostra disposizione sia del tipo "split raw", cioè un'immagine bit-a-bit di tutto il filesystem, suddivisa in singoli file distinti, la cui concatenazione costituisce il file originale. Le immagini split raw si utilizzano in genere per poter memorizzare le acquisizioni su filesystem con limitazioni sulla dimensione massima del file (quali ad esempio FAT). Per montare un'immagine raw suddivisa in file è sufficiente utilizzare il tool affuse, che fa uso della libreria Afflib[32] scritta e mantenuta da Simson Garfinkel. Ipotizziamo quindi di avere l'immagine nps-2009-domexusers.raw suddivisa in 20 file da circa 2GB l'uno: nps-2009-domexusers.001, nps-2009-domexusers.001, nps-2009-domexusers.002 fino ad arrivare alla nps-2009-domexusers.020. Non sarebbe ovviamente possibile utilizzare direttamente la loop device tramite il comando mount, poiché mount e loop device hanno visibilità sul singolo file. Eseguiremo, allora, il comando

```
# affuse nps-2009-domexusers.001 /mnt/raw
```

Il comando affuse raggruppa virtualmente tutti i singoli componenti dell'immagine raw suddivisa, creando un file virtuale /mnt/raw/nps-2009-

[32] http://afflib.org/

domexusers.001.raw. Tale file contiene virtualmente, nonostante il nome inglobi l'estensione del primo dei file componenti l'immagine, l'intera immagine raw, sulla quale si potrà eseguire il mount come di consueto. Si ricorda che anche questo file non esiste nel filesystem locale, è infatti un file virtuale, convertito on-the-fly a partire dai singoli file componenti l'immagine.

Per smontare l'immagine virtuale, una volta terminato il lavoro, si può utilizzare come di consueto il comando "umount /mnt/raw" oppure, se tale comando non dovesse avere successo, il comando della libreria FUSE "fusermount –u /mnt/raw".

Arrivati a questo punto siamo in grado di poter lavorare su di un'immagine raw dd monolitica, sia essa reale o virtuale tramite affuse/xmount, pur avendo a disposizione diversi formati (ewf, aff o persino split raw). Vediamo ora come montare le partizioni sulle quali lanciare il generatore di supertimeline log2timeline, ipotizzando di avere utilizzato xmount per il montaggio dell'immagine, dato che sembra essere più stabile e performante dello script mount_ewf.py.

Innanzitutto, tramite il comando "mml" della suite TSK analizziamo la tabella delle partizioni, attraverso la visualizzazione dei i confini e delle dimensioni in settori.

```
# mmls /mnt/raw/nps-2009-domexusers.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000000   0000000062   0000000063   Unallocated
02:  00:00   0000000063   0083859299   0083859237   NTFS (0x07)
03:  -----   0083859300   0083886079   0000026780   Unallocated
```

Notiamo che nell'immagine è presente una sola partizione NTFS, che inizia al settore 63. Dato che i settori dell'immagine hanno una dimensione standard di 512 byte, dopo aver creato la directory /mnt/c, all'interno della quale verrà eseguito il mount, possiamo procedere a montare la partizione come segue:

```
mount -o ro,loop,show_sys_files,offset=$((512*63)) /mnt/raw/nps-2009-domexusers.dd /mnt/c
```

Il parametro -o indica le opzioni con le quali verrà eseguito il mount. L'opzione "ro" è superflua se si utilizza affuse, mount_ewf.py o xmount, ma

per abitudine conviene utilizzarla comunque. L'opzione "show_sys_files" serve per poter visualizzare, oltre ai file "tradizionali" presenti nella partizione, anche quelli riservati ai metadati del filesystem in uso. Nel caso del filesystem NTFS, ad esempio, noteremo che all'interno di /mnt/c avremo alcuni file inusuali: $BadClus, $Bitmap, $Boot, $Extend, $LogFile, $MFTMirr, $Secure, $UpCase e $Volume.

5.10 Estrazione della tabella MFT

Anche se non lo visualizziamo nel listing, abbiamo la possibilità di accedere alla tabella MFT contenuta nel file $MFT, che ci servirà in seguito per integrare i timestamp del filesystem all'interno della timeline: provate a digitare "**od –c /mnt/c/\$MFT | head**" e noterete i vari FILE0 che introducono i numerosi record MFT (attivi e non, attuali e delle vecchie formattazioni) del filesystem. L'alternativa al mount con il parametro show_sys_files, per ottenere la tabella MFT, è quella di utilizzare il tool "icat -o 63 /mnt/raw/nps-2009-domexusers.dd -0 | od -c | head" della suite TSK tramite il comando "icat ", oppure avvalersi dei diversi tool gratuiti e commerciali.

```
# icat -o 63 /mnt/raw/nps-2009-domexusers.dd 0 | od -c | head
(oppure)
# od -c /mnt/c/\$MFT | head
0000000 F I L E 0 \0 003 \0 8 213 * \n \0 \0 \0 \0
0000020 001 \0 001 \0 8 \0 001 \0 240 001 \0 \0 \0 004 \0 \0
0000040 \0 \0 \0 \0 \0 \0 \0 \0 006 \0 \0 \0 \0 \0 \0
0000060 Q 001 \0 \0 \0 \0 \0 020 \0 \0 \0 ` \0 \0 \0
0000100 \0 \0 030 \0 \0 \0 \0 \0 H \0 \0 \0 030 \0 \0 \0
0000120 352 ) 240 312 277 2 311 001 352 ) 240 312 277 2 311 001
*
0000160 006 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000200 \0 \0 \0 \0 \0 001 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000220 \0 \0 \0 \0 \0 \0 \0 \0 0 \0 \0 \0 h \0 \0 \0
```

Pur esulando dall'argomento della trattazione si precisa che, in realtà, i file riservati NTFS visibili tramite l'opzione show_sys_files sono in numero maggiore. La potenza dei parametri del comando mount può spingersi molto oltre, permettendoci persino di accedere direttamente al Change Journal del filesystem NTFS, conosciuto anche con il nome di USN Journal[33]. Tale

[33] http://en.wikipedia.org/wiki/USN_Journal

registro, quando abilitato, mantiene un log dettagliatissimo di tutte le operazioni eseguite su filesystem con data di creazione, data di ogni modifica e persino la tanto agognata data di cancellazione. Paradossalmente, software commerciali blasonati come FTK[34], compresa l'ultima versione 3.4.1 dell'estate 2010, non sono in grado né di accedere a tale metafile, né tantomeno di parsificarlo.

Poiché la parte del registro che ci interessa è contenuta nell'Alternate Data Stream "$J", sfrutteremo un'opzione piuttosto sconosciuta del comando "mount", chiamata "streams_interface", fornendole il valore "Windows", poiché stiamo analizzando il filesystem NTFS. Dovremo quindi avere l'accortezza di montare il filesystem con il seguente comando:

```
# mount -o ro,loop,show_sys_files,streams_interface=windows,offset=$((512*63))
/mnt/raw/img.dd /mnt/c
```

A questo punto, in maniera piuttosto semplice, potremo accedere al Change Journal con la seguente operazione:

```
# od -c /mnt/c/\$Extend/\$UsnJrnl:\$J | head
```

Come per il caso del metafile $MFT, anche l'alternate data strema, contenuto nel metafile $UsnJrnl, non viene mostrato nel risultato di un listing sulla directory.

```
# ls -al /mnt/c/\$Extend
total 4
drwxrwxrwx 1 root root 0 2008-10-20 14:26 .
drwxrwxrwx 1 root root 4096 2008-10-30 03:17 ..
---------- 1 root root 0 1970-01-01 00:00 $ObjId
---------- 1 root root 0 1970-01-01 00:00 $Quota
---------- 1 root root 0 1970-01-01 00:00 $Reparse
```

Come si nota, infatti, la cartella $Extend contiene soltanto i tre metafile $ObjId, $Quota e $Reparse, manca il $UsnJrnl: accedendovi però direttamente ne potremo copiare o utilizzare il contenuto.

Ovviamente, come negli esempi precedenti, piuttosto che mostrare a schermo l'intestazione del file acceduto, passandolo in pipe al comando "head", potremo mandarlo in output su un qualsiasi file utilizzando il comando di redirezione ">". Il comando precedente diventerebbe, quindi:

[34] http://accessdata.com

```
# od -c /mnt/c/\$Extend/\$UsnJrnl:\$J > usnjrnl.bin
```

Negli esempi relative al Change Journal non è stata utilizzata l'immagine nps-2009-domexusers, bensì una generica /mnt/raw/img.dd, poiché nell'acquisizione della Digital Corpora il journalling NTFS non era attivo.

5.11 Generazione della supertimeline

Dopo a
ver montato la partizione (o le partizioni) da analizzare, e divagato sulle potenzialità della suite TSK, possiamo ora procedere con l'esecuzione del tool log2timeline.

Come operazione preliminare facoltativa possiamo, innanzitutto, ricavare due informazioni che potranno risultare utili per impostare correttamente i parametri del parser log2timeline o per verificare che siano corretti quelli rilevati automaticamente.

Cominciamo col verificare il tipo di sistema operativo presente sulla macchina in analisi. La struttura delle directory può già fornire un'idea sulla versione del sistema operativo installato, la presenza[35] nella root di una cartella "Users" ci fa ipotizzare Vista o Seven, "Documents and Settings" riconduce a versioni precedenti tipo XP. Il folder di sistema "Winnt" è proprio di NT o 2000 (anche se rinominabile in fase di installazione), "Windows" degli OS successivi. Nella cartella degli utenti, la presenza di un folder "Local Settings" identifica XP, mentre un "AppData" Vista o Seven. Microsoft fornisce alcune tabelle di conversione tra folder dei suoi diversi OS, è comunque meglio[36] attenersi alle informazioni contenute nel registro nella chiave HKLM\Software\Microsoft\Windows NT\CurrentVersion o della sezione di risorse PE del file %WinDir%\system32\ntoskrnl.exe per avere maggiore attendibilità.

Se abbiamo a disposizione[37] il tool Regripper, di Harlan Carvey, possiamo utilizzare lo script fornito nella suite per acquisire informazioni circa l'OS estraendole dal registro:

[35] http://windows.microsoft.com/en-US/windows7/Where-are-my-files-and-folders-after-upgrading-from-Windows-XP-or-Windows-Vista

[36] http://www.forensicswiki.org/wiki/Determining_OS_version_from_an_evidence_image

[37] Nel caso in cui manchi nella distribuzione che stiamo utilizzando possiamo tranquillamente scaricarlo da http://regripper.wordpress.com e installarlo, anche su Live CD.

```
# rip.pl -r /mnt/c/WINDOWS/system32/config/software -p winver
Launching winver v.20081210
ProductName = Microsoft Windows XP
CSDVersion = Service Pack 3
InstallDate = Mon Oct 20 21:43:18 2008
```

Scopriamo, quindi, che sulla macchina acquisita è installato Windows XP con SP3, con data di installazione risalente al 20 ottobre 2008.

Vediamo ora di verificare il fuso orario impostato sul sistema operativo memorizzato sulla partizione dell'immagine disco in analisi:

```
# rip.pl -r /mnt/c/WINDOWS/system32/config/system -p timezone2
Launching timezone v.20101219
TimeZoneInformation key
ControlSet001\Control\TimeZoneInformation
LastWrite Time Tue Oct 28 16:31:28 2008 (UTC)
 DaylightName -> Pacific Daylight Time
 StandardName -> Pacific Standard Time
 Bias -> 480 (8 hours)
 ActiveTimeBias -> 480 (8 hours)
 TimeZoneKeyName -> N/A
```

Osserviamo che la timezone area relativa all'immagine acquisita è "Pacific Standard Time", corrispondente a PST8PDT nella notazione utilizzata da TSK e log2timeline. In genere, nei casi con materiale italiano, avremo come fuso orario "DaylightName -> ora legale Europa occidentale" e "StandardName -> ora solare Europa occidentale", che corrisponde nella notazione "Europe/Rome".

L'ultima versione di log2timeline, la 0.60, è in grado di ricavare autonomamente tali informazioni tramite il preprocessing (opzione "-p"), ma non è male verificare di persona che tali dati siano stati ricavati correttamente, dato che condizionano l'intera parsificazione, che verrà lanciata con il comando seguente:

```
# log2timeline -p -r -z PST8PDT /mnt/c/ -w c-log2t.csv
Start processing file/dir [/mnt/c/] ...
Starting to parse using input modules(s): [all]
Loading output file: csv
[PreProcessing] The default browser of user administrator according to registry is: (FIREFOX.EXE)
[PreProcessing] Unable to determine the default browser for user default user
[PreProcessing] Unable to determine the default browser for user networkservice
[PreProcessing] The default browser of user domex1 according to registry is: (FIREFOX.EXE)
[PreProcessing] Unable to determine the default browser for user localservice
[PreProcessing] The default browser of user domex2 according to registry is: (FIREFOX.EXE)
[PreProcessing] Hostname is set to REALISTIC_XP
[PreProcessing] The timezone according to registry is: (PST) Pacific Standard Time
[PreProcessing] The timezone settings are NOT overwritten so the settings might have to be adjusted.
[PreProcessing] The default system browser is: : IEXPLORE.EXE ("C:\Program Files\Internet Explorer\iexplore.exe" -nohome)
Unable to open /mnt/c//$Extend/$ObjId
Unable to open /mnt/c//$Extend/$Quota
Unable to open /mnt/c//$Extend/$Reparse
Unable to open /mnt/c//$Secure
[…]
Run time of the tool: 2928 seconds
```

Terminato il preprocessing, l'output su schermo si fermerà dando l'impressione che il tool si sia bloccato. Se apriamo un'altra finestra di terminale, ci posizioniamo nella stessa cartella dalla quale abbiamo lanciato il comando e digitiamo **"tail -f c-log2t.csv**", ci accorgeremo, invece, che l'estrazione dei timestamp dagli artefatti sta procedendo a velocità sostenuta.

Vediamo una breve descrizione dei parametri da linea di comando passati al tool log2timeline:

-p: abilita il preprocessing dei file e permette di rilevare alcune informazioni quali l'hostname della macchina, gli utenti, i browser di default degli utenti, etc...

-r: abilita lo scanning ricorsivo della directory passata in input (in genere si usa sulla root del filesystem da analizzare)

-f: indica il subset di plugin (o il singolo plugin) da utilizzare nella scansione. Conviene limitare i plugin a quelli attinenti al sistema in questione, per velocizzare l'output ed evitare errori dovuti all'applicazione di plugin per dati diversi da quelli in analisi.

-z: indica la timezone di riferimento per i file presenti sul sistema in analisi

-w: contiene il nome del file in cui scrivere l'output dell'elaborazione, di default in formato csv

Terminata la parsificazione di tutti i timestamp presenti nei file non cancellati, che risiedono sul disco C:, procediamo utilizzando una delle più recenti innovazioni nella suite log2timeline, che ci permette di parsificare la tabella MFT estraendo sia il campo $STDinfo sia il campo $FILEname. Dal confronto dei timestamp contenuti in questi due campi sarà possibile poter rilevare eventuali attacchi di anti-forensics (tramite tool quali ad esempio il sopracitato "timestomp") o azioni di virus/trojan che hanno modificato il valore di $STDinfo senza riuscire, invece, a modificare il campo $FILEname.

Per poter parsificare il record MFT abbiamo due possibilità: lo estraiamo e lo passiamo come parametro a log2timeline, oppure usiamo come parametro l'accesso diretto eseguito grazie alla visibilità sull'MFT fornita dal mount con il parametro show_sys_files. Per estrarre l'MFT, come visto in precedenza, possiamo fare uso del comando icat della suite TSK, oppure possiamo copiarlo nella directory di lavoro tramite un accesso diretto al metafile $MFT nella directory in cui è stato montato il filesystem (a patto che il mount sia avvenuto tramite il parametro show_sys_files). Dato che al momento non ci interessa salvare una copia dell'MFT nella directory di lavoro, vi accediamo direttamente utilizzando il metafile $MFT come parametro del tool log2timeline.

```
# log2timeline -f mft /mnt/c/\$MFT -z PST8PDT -m C: -w c-mft.csv
Start processing file/dir [/mnt/c/$MFT] ...
Starting to parse using input modules(s): [mft]
Loading output file: csv
```

Nelle versioni precedenti la 0.60, questa operazione di estrazione dei file presenti e cancellati dal filesystem veniva effettuata tramite il tool fls della suite TSK, con la differenza che in quel caso non era possibile rilevare potenziali tentativi di compromissione dei timestamp, poiché venivano analizzato soltanto i record $StdInfo.

Come accennato in precedenza, non ci limiteremo ad analizzare i metadati presenti nei file rilevati sui settori allocati del filesystem, ma eseguiremo un carving che ci permetterà di ricavare anche un buon numero di timestamp prelevati da materiale ricuperato in zone non allocate. Di particolare interesse, come vedremo, saranno le informazioni temporali ricavate dalle varie copie del registro di sistema, dai .lnk Windows, dai registri eventi e da tutti i documenti ricuperati.

Per effettuare il carving possiamo utilizzare il tool che preferiamo: in genere

Photorec, Foremost e Scalpel si sono sempre dimostrati degli ottimi strumenti. Nel nostro esempio utilizzeremo Photorec[38] con il comando seguente, di cui mostriamo direttamente l'output di conclusione del ricupero.

```
# photorec /mnt/raw/nps-2009-domexusers.dd

PhotoRec 6.11, Data Recovery Utility, April 2009
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /mnt/raw/nps-2009-domexusers.dd - 42 GB / 40 GiB (RO)
 Partition Start End Size in sectors
 1 * HPFS - NTFS 0 1 1 5219 254 63 83859237

992 files saved in /mnt/hgfs/forensics/memberbook/carving/recup_dir directory.
Recovery completed.
txt: 625 recovered
exe: 237 recovered
tx?: 46 recovered
gif: 22 recovered
gz: 21 recovered
cab: 9 recovered
jpg: 7 recovered
png: 7 recovered
lnk: 6 recovered
edb: 4 recovered
others: 8 recovered
[ Quit ]
```

Durante l'esecuzione, confermiamo l'analisi dell'immagine passata come parametro, impostiamo come partition table type "Intel", selezioniamo HPTF-NTFS (e non "whole disk") come area da elaborare e contemporaneamente, nella finestra "File Opt" verifichiamo che tutti i tipi di file siano impostati secondo la configurazione di default (volendo possiamo rimuovere quelli che sappiamo non contenere metadati o di cui non ci importano i timestamp, come gli exe, i cab, i txt, etc...). Si consiglia di rimuovere dal carving gli eseguibili ("exe") in quanto in genere piuttosto numerosi nei sistemi e ricchi di timestamp inutili quali data di linking e compilazione. A questo punto selezioniamo "Search", confermiamo che il filesystem type è "Other" (FAT/NTFS/HFS+/ReiserFS/...) e abilitiamo il carving soltanto sullo spazio non allocato, selezionando "Free" nella finestra in cui viene richiesta la tipologia di carving. Confermiamo ancora,

[38] www.cgsecurity.org/wiki/PhotoRec

premendo -y', la directory in cui salvare i file ricuperati ed avviamo così il processo di estrazione.

Una volta terminata la fase di carving, tramite Photorec, ci ritroveremo diverse directory con nomi poco significativi, quali "recup_dir.1", "recup_dir.2" e così via. Tramite un comodo script Python, chiamato "recovery.py", presente nella distribuzione DEFT, possiamo riorganizzare le directory spostando i file in base alla loro estensione, ritrovandoci quindi con directory quali "doc", "reg", "jpg", etc... con l'ovvio contenuto.

L'immagine domexusers non contiene un numero di artefatti sufficiente per poter mostrare l'utilità dell'operazione di timelining su file ricuperati tramite carving. Nei casi reali, durante un'analisi forense di un PC con mesi o anni di utilizzo, è possibile ricavare diverse informazioni indispensabili con i relativi timestamp inquadrati in una timeline globale, quali ad esempio:

- versioni precedenti dei file del registro (system, ntuser.dat, software, sam) che riportano inserimento di periferiche USB, installazione software, userassist, etc...
- eventi di sistema
- informazioni sull'esecuzione dei programmi ricavati dai prefetch
- informazioni ricavate dai dati exif delle fotografie digitali in formato jpg
- numerosi link .LNK con riferimenti a dispositivi connessi, percorsi di directory e file aperti tramite Office o tramite riferimento/link ai programmi installati
- documenti PDF, Word, Excel, OpenOffice con date di creazione, modifica, stampa (la data di creazione viene persa se il file viene cancellato dall'MFT dopo la rimozione... rimane come metadato interno al file insieme a diversi altri timestamp)
- history dei principali browser con le URL delle pagine visitate e i timestamp delle visite, anche se l'utente di tanto in tanto si è premurato di rimuovere la cronologia
- frammenti di conversazioni Skype estratti da database sqlite

La peculiarità di questa analisi è che avviene su parti di sistema ormai scomparse, permettendoci di inserire nella timeline anche timestamp relativi a dati cancellati. Per estrarre i timeline dai dati ricavati tramite carving, eseguiamo come abbiamo appreso sopra il comando log2timeline:

```
# log2timeline -r -z PST8PDT ./carving/ -w c-log2t-carving.csv
Start processing file/dir [./carving/] ...
Starting to parse using input modules(s): [all]
Loading output file: csv
Error while parsing (./carving//recup_dir.2/f2778160.msf):
[LOG2T] Initialization of the input module failed, unable to parse.
Run time of the tool: 22 seconds
```

Non essendo un sistema di produzione, come detto sopra, l'immagine domexusers non contiene molto materiale nelle aree non allocate. Nonostante questo, diverse entry della supertimeline risultano di notevole interesse, come si può osservare scorrendo il file "c-log2t-carving.csv".

Per esempio, veniamo a sapere che il 18 settembre è stato creato il LNK all'applicazione Picasa3, modificato per l'ultima volta il 21 ottobre:

09/18/2008,09:44:24,PST8PDT,..A..,LNK,Shortcut LNK,Access,-,-,C:/Program Files/Google/Picasa3/Picasa3.exe […]
09/18/2008,09:44:24,PST8PDT,..CB,LNK,Shortcut LNK,Created,-,-,C:/Program Files/Google/Picasa3/Picasa3.exe […]
10/21/2008,08:11:48,PST8PDT,M...,LNK,Shortcut LNK,Modified,-,-,C:/Program Files/Google/Picasa3/Picasa3.exe […]

Abbiamo date di creazione, modifica, autore e informazioni varie ricavate dai metadati Open XML dei documenti MS Office presenti sul sistema:

10/29/2008,09:15:00,PST8PDT,MACB,OXML,Open XML Metadata,created,domex1,-,, - Application: Microsoft Office Word - AppVersion: 12.0000,2 […]
10/29/2008,09:15:00,PST8PDT,MACB,OXML,Open XML Metadata,modified,domex1,-,, - Application: Microsoft Office Word - AppVersion: 12.0000,2 […]

Abbiamo timestamp ricavati dai dati exif di diverse immagini in formato jpeg:

09/27/2007,15:45:31,PST8PDT,MACB,EXIF,EXIF metadata,XMP/ModifyDate (1),-,-,ModifyDate […]
09/21/2007,15:45:14,PST8PDT,MACB,EXIF,EXIF metadata,XMP/CreateDate,-,-,CreateDate […]

Ricaviamo diverse URL con timestamp relative al momento in cui sono state aperte sul browser, che rivedremo nei paragrafi successivi durante l'analisi dettagliata dei risultati:

10/20/2008,15:42:24,PST8PDT,..ACB,WEBHIST,Internet Explorer,time1,.Administra

tor,-,visited http://www.google.com/search?hl=en&q=pidgin&aq=f [...]
10/20/2008,15:42:55,PST8PDT,M...,WEBHIST,Internet Explorer,time2,Administrat
or,-,visited http://sourceforge.net/project/downloading.php?groupname=pidgin&fil
ename=pidgin-2.5.2.exe&use_mirror=internap [...]

Notiamo, tra l'altro, che all'interno delle URL è possibile visualizzare le query passata al motore di ricerca, identificando così anche il tipo di ricerche eseguite dal proprietario del PC.

Le versioni precedenti alla 0.60 di log2timeline non erano in grado di parsificare il registro di sistema ed estrarre i timestamp; era quindi necessario utilizzare lo script regtime.pl[39], di Harlan Carvey, e scegliere come output "mactime" e successivamente mixare la timeline risultante con quella generata dal filesystem tramite fls della suite TSK e quella globale generata da log2timeline. Pur potendo eseguire tutte le analisi con il solo strumento log2timeline, può comunque essere utile conoscere l'esistenza sia del comando fls della suite TSK, sia dello script regtime.pl, il cui utilizzo prevede il passaggio dal formato mactime a quello CSV. Per completezza, anche se non verrà utilizzato nella timeline perché i valori del registro sono già forniti direttamente da log2timeline, riportiamo i comandi di esecuzione dello script regtime.pl che avremmo utilizzato per una timeline del registro:

```
# regtime.pl -m HKLM-SAM/ -r /mnt/c/WINDOWS/system32/config/SAM >>
regtime.body
# regtime.pl -m HKLM-SECURITY/ -r /mnt/c/WINDOWS/system32/config/SE-
CURITY >> regtime.body
# regtime.pl -m HKLM-SOFTWARE/ -r /mnt/c/WINDOWS/system32/config/
software >> regtime.body
# regtime.pl -m HKCU-DOMEX1/ -r /mnt/c/Documents\ and\ Settings/domex1/
NTUSER.DAT >> regtime.body
# regtime.pl -m HKCU-DOMEX2/ -r /mnt/c/Documents\ and\ Settings/domex2/
NTUSER.DAT >> regtime.body
# mactime -y -m -d -z PST8PDT -b regtime.body > regtime.csv
```

Pur non potendolo eseguire in pratica nel case study in analisi, aggiungiamo che oltre alle aree non allocate esiste un altra fonte di dati dalla quale può risultare decisivo estrarre timestamp, in questo caso principalmente tramite il tool fls della suite The Sleuth Kit e l'ausilio di comandi quali vssadmin e

[39] Contenuto nel DVDROM allegato ai libri Windows Forensics Analysis and Incident Response. Per chi non avesse acquistato il libro, l'autore mette a disposizione i suoi script open source all'indirizzo http://code.google.com/p/winforensicaanalysis

dd.exe. Utilizzando un sistema operativo della serie Windows Vista, 7 o 2008 server è possibile, infatti, avere accesso alle singole shadow copy[40] passate e parsificare il contenuto senza troppa difficoltà per quanto riguarda la timeline del filesystem. Con un po' di lavoro in più potremo anche estrarre le diverse shadow copy (o averne comunque accesso logico) per potervi utilizzare sopra il tool log2timeline. Rispetto all'analisi tradizionale, l'integrazione delle shadow copy ci fornisce una sorta di "fotografia" di com'era il filesystem e di com'erano i file in precisi momenti del passato, ci fornisce informazioni circa gli artefatti che nello stato attuale del sistema non sono più accessibili o elaborabili correttamente. Allo stato attuale, l'analisi delle shadow copy, mediante strumenti open source o free, risulta comunque piuttosto laboriosa e complessa.

Ricapitolando, abbiamo generato tre file necessari per la supertimeline:

- c-log2t.csv: parsing tramite log2timeline della partizione C:
- c-log2t-carving.csv: parsing tramite log2timeline delle aree non allocate della partizione C:
- c-mft.csv: parsing della tabella MFT della partizione C:

È giunto ora il momento di unificare i risultati ottenuti per poter analizzare i dati e creare una supertimeline globale che, successivamente, scremeremo per ridurne le dimensioni:

```
# cat *.csv > supertimeline-unsorted.csv
```

5.12 Analisi dei risultati

Siamo arrivati, tramite un processo elaborato ma non complesso, ad avere un file, in formato CSV, contenente tutti i timestamp estratti dal disco, dalle aree allocate e da quelle non allocate. Come noterete, il file supertimeline-unsorted.csv contiene quasi 2519230 timestamp ed occupa circa un Gbyte. Non c'è da spaventarsi perché molte entry sono duplicate e verranno rimosse durante il passo successivo.

La prima cosa che faremo, infatti, sarà quella di eliminare i record doppi, ordinarli e, eventualmente, filtrarli in base al periodo temporale di interesse oltre che tramite le keyword note. La grossa mole di dati ottenuta richiede un'attenta ricerca di tutti i filtri che possono ridurre il carico di elaborazione successivo. Come consiglio, nei casi in cui le chiavi di registro sembra non

[40] http://en.wikipedia.org/wiki/Shadow_Copy

fornire informazioni rilevanti (e l'immagine in analisi sembra uno di quelli), si possono tranquillamente filtrare, tramite whitelist, i timestamp relativi al Windows Registry, oppure utilizzare il parametro "-f" di log2timeline, selezionando "win7_no_reg" oppure "winxp_no_reg" come gruppo di input modules da utilizzare.

A questo punto utilizziamo lo script l2t_process per ordinare i record ottenuti dall'unificazione dei file risultante dall'operazione precedente, eliminare i duplicati ed eventualmente analizzare potenziali tentativi di timestomping.

Tramite l2t_process è possibile parsificare un file, in formato .CSV, contenente numerosi timestamp generati da log2timeline o timescanner, ordinare i record, eliminare i duplicati, ma soprattutto eseguire, facoltativamente, le seguenti operazioni:

- verificare le possibili attività di timestomping eseguite sui record MFT;
- filtrare i risultati inserendo soltanto quelli contenenti keyword rilevanti (blacklist);
- filtrare i risultati scartando quelli contenenti keyword innocue (whitelist);
- analizzare, tramite rappresentazione grafica in assi cartesiani, i file presenti nella directory "WINDOWS/System32", e mettere in relazione il numero della entry MFT (in genere assegnato in maniera ordinale ai file) con il timestamp di creazione del file. Gli elementi che presentano anomalie (risiedono agli estremi oppure interrompono un disegno regolare di timestamp/numero entry) possono essere ricollegabili ad attività di trojan/virus;
- Filtrare i risultati in base ad un intervallo temporale prescelto.

Eseguiamo lo script l2t_process con i parametri seguenti:

```
e# l2t_process -b supertimeline-unsorted.csv > supertimeline.csv

Total number of events that fit into the filter (got printed) = 2519229
Total number of duplicate entries removed = 2133232
Total number of events skipped due to whitelisting = 0
Total number of events skipped due to keyword filtering = 0
Total number of processed entries = 2519229
Run time of the tool: 189 sec
```

Otterremo, in questa maniera, un file CSV ordinato, senza duplicati, contenente oltre ai timestamp relativi agli artefatti, anche delle indicazioni

circa l'eventuale attacco di timestomping perpetrato sul sistema. Non abbiamo filtrato in base a whitelist o keyword. In particolare il file CSV sarà formattato secondo la struttura presentata nella tabella seguente:

Field	Description
Date	The date of the event, in the format of MM/DD/YYYY
Time	The time of day, expressed in a 24h format, HH:MM:SS
Timezone	the timezone that was used to call the tool with.
MACB	The MACB meaning of the fields, mostly for compatibility with the mactime format.
Source	The short name for the source. All web browser history is for instance WEBHIST, registry entries are REG, simple log files are LOG, etc.
Sourcetype	A slightly more comprehensive description of the source, "Internet Explorer" instead of WEBHIST, "NTUSER.DAT" instead of REG, etc.
Type	The type of the timestamp itself, such as "Last Accessed", "Last Written" or "Last modified", etc.
User	The username associated with the entry, if one is available.
Host	The hostname associated with the entry, if one is available.
Short	A short description of the entry, usually contains less text than the full description field.
Desc	The description field, this is where most of the information is stored, the actual parsed description of the entry.
Version	The version number of the timestamp object.
Filename	The filename with the full path of the filename that contained the entry
Inode	The inode number of the file being parsed.
Notes	Some input modules insert additional information in the form of a note, which comes here. Or it can be used during the review by the investigator.
Format	The name of the input module that was used to parse the file.
Extra	Some additional information parsed is joined together and put here.

Pur avendo analizzato un'immagine di partenza piuttosto esigua, notiamo che, nonostante la rimozione di duplicati, la timeline occupa ancora circa 400 MBytes e contiene 385993 timestamp: troppi per una analisi! Le potenzialità della supertimeline è enorme, ma la quantità di dati prodotti rende estremamente difficile far risaltare le informazioni davvero utili. Si rivela, quindi, decisamente strategica l'attività di filtraggio dei risultati, in base a parametri che via via riterremo più adeguati: un intervallo temporale, delle keyword, degli artefatti o quanto a nostra disposizione.

L'immagine di esempio non ci consente di effettuare grandi anticipazioni sul tipo di analisi che andremo a fare, non essendoci stato fornito un arco temporale di interesse o informazioni su cosa cercare: sappiamo soltanto che i due utenti del PC utilizzano IM ed email per comunicare fra di loro. Ovviamente nei casi reali le supertimeline diventano decisive quando si possono avere a disposizione maggiori informazioni sul tipo di reato commesso o di attività che si vuole verificare.

Se volessimo simulare, comunque, una ricerca di potenziali comportamenti anomali, potremo basarci sulla quantità di file utilizzati (creati, modificati, acceduti) giornalmente. Per ottenere questa utilissima informazione ci viene

incontro il tool fls, parte della suite TSK, menzionato più volte nei paragrafi precedenti e che verrà utilizzato per la generazione di una tradizionale timeline del filesystem:

```
# fls -o 63 -r -m C: /mnt/raw/nps-2009-domexusers.dd > c-timeline.body
# mactime -y -m -d -i day c-timeline-daily.csv -z PST8PDT -b c-timeline.body >
c-timeline.csv
# mactime -y -m -d -i hour c-timeline-hourly.csv -z PST8PDT -b c-timeline.body
> c-timeline.csv
```

Ci ritroveremo così non solo una tipica timeline del FS (come quella generata da Autopsy, per intenderci), ma anche due summary giornalieri (con granularità sui giorni e sulle ore) riferiti al numero di file utilizzati. Tale informazione, che può anche essere elaborata tramite foglio di calcolo per effettuare una visualizzazione grafica o per ordinare le voci, ci permette di capire, a grandi linee, in quali periodi il PC è stato più utilizzato. Un altro vantaggio, che deriva dall'utilizzo di fls, è che può fornire un double check sulla correttezza delle operazioni svolte sinora, tramite il parsing dell'MFT svolto dal tool log2timeline.

Se scorriamo con un editor di testo il file "c-timeline-hourly.csv", ad esempio, ci rendiamo conto che il periodo in cui il PC è stato soggetto a più operazioni sul filesystem è Lunedì 20 ottobre 2008, in particolare intorno alle ore 17, ed un altro picco di anomalie Mercoledì 4 agosto 2004 intorno alle ore 5. I giorni che vanno dal 20 al 30 ottobre 2008, infine, risultano quelli in cui c'è stata maggiore attività.

```
# cat c-timeline-hourly.csv
[...]
Wed 08 04 2004 05:00:00, 13029
[...]
Mon 10 20 2008 07:00:00, 5420
Mon 10 20 2008 14:00:00, 8427
Mon 10 20 2008 15:00:00, 2190
Mon 10 20 2008 16:00:00, 5035
Mon 10 20 2008 17:00:00, 14009
Mon 10 20 2008 18:00:00, 4893
Mon 10 20 2008 20:00:00, 119
Mon 10 20 2008 21:00:00, 2553
[...]
```

Se ricordate, prima di operare tramite lo script log2timeline, abbiamo verificato le caratteristiche del sistema operativo installato tramite gli script Regripper: oltre all'informazione sulla presenza di Windows XP, avevamo scoperto che l'installazione era avvenuta proprio il... 20 ottobre 2008. I file precedenti quindi, presumibilmente, risalgono ad un utilizzo antecedente l'installazione di ottobre, la quale avrà sovrascritto quanto già esistente.

Sceglieremo allora, come data di inizio del periodo di interesse, proprio il 20 ottobre 2008, lanciando nuovamente l2t_process con il seguente comando:

```
# l2t_process -b supertimeline-unsorted.csv -y 2008-10-20 > supertimeline-20081020.csv
There are 121 that fall outside the scope of the date range, yet show sign of possible timestomping.
Would you like to include them in the output? [Y/n] Y

Total number of events that fit into the filter (got printed) = 2450488
Total number of duplicate entries removed = 2116273
Total number of events skipped due to whitelisting = 0
Total number of events skipped due to keyword filtering = 0
Total number of processed entries = 2519229
Run time of the tool: 319 sec
```

Come si nota, lo script avverte che 121 linee hanno date (basate sui metadati SI o FN del filesystem NTFS) che ricadono al di fuori del periodo in analisi, chiedendo se includerle nel report. Le includeremo, anche se il tipo di analisi basato sul discostamento tra SI e FN contiene quasi sempre falsi positivi e richiede tempo per l'analisi.

Dall'analisi del file prodotto possiamo ricavare numerose informazioni, a partire dalle date e dagli orari in cui il PC è stato acceso e spento. Basterà infatti ricercare gli eventi[41] con ID 6009 (boot), 6005 (event log started), 6006 (clean shutdown) e 6008 (dirty shutdown).

Rileviamo quindi che il 20 ottobre il PC è stato avviato alle 07:30:17:

*10/20/2008,07:30:17,PST8PDT,MACB,EVT,Event Log,Time generated/written,-,MACHINENAME,**EventLog/6009** […]*
*10/20/2008,07:30:17,PST8PDT,MACB,EVT,Event Log,Time generated/written,-,MACHINENAME,**EventLog/6005;** […]*

Il PC è stato spento alle 15:00 per essere riavviato un minuto dopo,

[41] http://support.microsoft.com/kb/196452/

probabilmente per un reboot:

*10/20/2008,15:00:52,PST8PDT,MACB,EVT,Event Log,Time generated/written,-,USER-7B73421FE5,**EventLog/6006**; […]*
*10/20/2008,15:01:58,PST8PDT,MACB,EVT,Event Log,Time generated/written,-,USER-7B73421FE5,**EventLog/6009**; […]*
*10/20/2008,15:01:58,PST8PDT,MACB,EVT,Event Log,Time generated/written,-,USER-7B73421FE5,**EventLog/6005**; […]*

Dai log, che non riportiamo per brevità, il PC risulta essere stato nuovamente spento alle 15:10 per essere riavviato alle 15:23, poi ancora spento alle 18:29 per essere riavviato alle 18:33.

Procedendo oltre con la ricerca, vediamo anche che alle ore 15:42, sempre del 20 ottobre, l'utente Administrator ha cercato su Google il programma di Instant Messaging "pidgin":

10/20/2008,15:42:23,PST8PDT,M...,WEBHIST,Internet Explorer,Content viewed,-,REALISTIC_XP,visited http://clients1.google.com/complete/search?hl=en&q=pidgin […]

È , quindi, passato sul sito www.pidgin.im che lo ha portato su sourceforge, dal quale poteva scaricare il software.

10/20/2008,15:42:39,PST8PDT,M...,WEBHIST,Internet Explorer,Content viewed,-,REALISTIC_XP,visited http://www.pidgin.im/download/windows/ […]
10/20/2008,15:42:43,PST8PDT,.ACB,WEBHIST,Internet Explorer,time1,Administrator,-,visited http://downloads.sourceforge.net/pidgin/pidgin-2.5.2.exe[...]

Il software è stato effettivamente scaricato e salvato su PC nella cartella "C:/Documents and Settings/All Users/Documents/":

10/20/2008,15:43:20,PST8PDT,M..B,FILE,NTFS $MFT,$FN [M..B] time,-,-,C:/Documents and Settings/All Users/Documents/pidgin-2.5.2.exe[…]

Il giorno dopo, l'eseguibile scaricato, è stato acceduto, insieme all'installer del noto programma di posta elettronica Thunderbird, mentre in seguito l'utente domex1 si è registrato presso il sito AOL:

10/21/2008,11:13:04,PST8PDT,.A..,FILE,NTFS $MFT,$FN [.A..] time,-,-,C:/Documents and Settings/All Users/Documents/pidgin-2.5.2.exe […]
10/21/2008,11:04:08,PST8PDT,.A..,FILE,NTFS $MFT,$FN [.A..] time,-,-,C:/Documents and Settings/All Users/Documents/Thunderbird Setup 2.0.0.17.exe […]
10/21/2008,12:19:17,PST8PDT,M...,WEBHIST,Internet Explorer,Last Access,domex1,REALISTIC_XP,visited https://reg.my.screenname.aol.com/_cqr/

registration/initRegistration.psp [...]

Senza addentrarci troppo nei dettagli, passiamo al 29 ottobre e vediamo che l'utente domex1 ha creato un documento "This is a word document by domex user 1.docx" in Ms Word, che ha lasciato traccia tra i Recent sotto forma di link .LNK:

10/29/2008,09:14:00,,MACB,OXML,Open XML Metadata,created,domex1,REAL ISTIC_XP,, - Application: Microsoft Office Word - AppVersion: 12.0000,2,/mnt/c// Documents and Settings/domex1/My Documents/This is a word document by domex user 1.docx [...]
10/29/2008,09:14:52,PST8PDT,..A..,LNK,Shortcut LNK,Access,-,REALISTIC_XP, C:/Documents and Settings/domex1/My Documents/This is a word document by domex user 1.docx <-/mnt/c//Documents and Settings/domex1/Recent/This is a word document by domex user 1.lnk- which is stored on a local vol type - [...]
10/29/2008,09:14:52,PST8PDT,MACB,FILE,NTFS $MFT,$SI [MACB] time,-,-,C:/ Documents and Settings/domex1/My Documents/This is a word document by domex user 1.docx, [...]

Allo stesso modo, analizzando la supertimeline, possiamo notare che domex1 ha creato anche il documento "This is a word document sent by domex user 1.docx" e il foglio di calcolo "This is a spreadsheet by domex user 1.xlsx", li abbia aperti e modificati, si sia connesso agli account creati tramite Gmail e Live.com, abbia utilizzato la posta elettronica, Pidgin e diverse altre azioni potenzialmente rilevanti.

Come accennato, alcuni paragrafi addietro, la particolarità della supertimeline non è quella di riuscire ad estrarre le informazioni temporali relative al registro, alla navigazione Internet, ai file aperti e modificati, ai link creati, ecc., ma di poterli organizzare in una struttura organica in modo da avere una visione globale e sequenziale.

Nei casi reali capita molto spesso di poter rilevare accessi a webmail o Facebook (con userid in chiaro, pagine dei profili, etc...), seguiti da inserimento di periferiche USB, copia di file, nuove query su motori di ricerca, inserimento di URL nei bookmark, esecuzione di programmi, accesso alla posta.

*10/29/2008,19:44:34,,MACB,REG,UserAssist key,Time of Launch,domex2,REALISTIC_XP,UEME_RUNPATH:C:/Program Files/Mozilla Thunderbird/**thunderbird.exe**, [Count: 2] [...]*
*10/30/2008,00:50:43,PST8PDT,MACB,PRE,XP Prefetch,Last run,-,REALISTIC_ XP,AIM6.EXE-34DC5725.pf: **AIM6.EXE** was executed,AIM6.EXE-34DC5725.pf - [AIM6.EXE] was executed - run count [5] [...]*
10/30/2008,00:51:52,PST8PDT,MA..,WEBHIST,Firefox 3 history,dateAdded,dom ex1,REALISTIC_XP, Bookmarked Obama reaches out with historic TV ad (http:// news.bbc.co.uk/go/rss/-/2/hi/americas/us_elections_2008/7699058.stm) [...]

Spesso capita, persino, di rilevare l'utilizzo di programmi di anti-forensics nella stessa timeline, potendo vedere chiaramente il download dal web, l'installazione e l'esecuzione.

Le informazioni ricavabili, ricapitolando, sono numerose e in genere devono essere integrate da un'analisi dettagliata e focalizzata su quanto emerso durante l'analisi. Il presente case study non pretende di essere esaustivo o di portare alla risoluzione di un caso che, data l'immagine di esempio, non è risolvibile perché non esiste. Ciò che si è cercato di fare è di mostrare, al lettore, le diverse potenzialità degli strumenti che portano alla generazione di una supertimeline ed illustrare il tipo di informazioni estratte e le possibilità di utilizzo.

Chi volesse esaminare in dettaglio i 2.7 GB di log prodotti dai diversi strumenti utilizzati per l'analisi del case study sull'immagine di prova, senza dover tirare su un ambiente e generarseli da solo, può scaricarli dal link www.forensicator.com/files/memberbook.7z.

5.13 Strumenti per raffinare l'analisi

L'analisi manuale, portata avanti pocanzi sul file di testo contenente i log della supertimeline, può essere supportata da alcuni strumenti testuali e grafici. Possono essere usate diverse soluzioni per filtrare i risultati, a cominciare dallo script l2t_process stesso che, come si è visto, è in grado di scremare la timeline in base ad un arco temporale, a keyword positive e a keyword negative. Nel caso in cui si sia interessati soltanto a particolari artefatti (browser history, registry, lnk, prefetch, etc...) è possibile utilizzare il comando Linux (disponibile anche in Windows) "grep", che parsifica il file sorgente e produce un file di destinazione filtrato in base alla query ricevuta come parametro, che dovrà contenere il TAG relativo all'artefatto di nostro interesse.

In Windows, alternativamente, oltre a poter utilizzare il porting di grep[42] suggeriamo, come supporto alla visualizzazione e al filtro della timeline, l'utilissimo Highlighter[43] della Mandiant. Tale visualizzatore possiede un'ottima capacità di filtraggio, come una sorta di grep in tempo reale, del file di cui si sta visionando il contenuto in base alle keyword di volta in volta prescelte. Un feedback grafico nell'area destra dello schermo ci informa circa la diffusione delle keyword selezionate. Tramite menù contestuale o shortcuts è possibile nascondere le righe che non contengono le parole di nostro interesse, evidenziare le keyword rilevanti e scremare via via, lasciando visibile soltanto

[42] http://gnuwin32.sourceforge.net/packages/grep.htm
[43] http://www.mandiant.com/products/free_software/highlighter/download

la parte di nostro interesse. Così, ad esempio, se siamo interessati soltanto ai link LNK, elaborati da log2timeline, selezioniamo la stringa "Shortcut LNK" e optiamo per la voce "Show only" nel menù contestuale. Allo stesso modo, se siamo interessati ai timestamp relativi ad un file, ad una pagina web (alle visite ad essa riferite) o a una chiave di registro non dobbiamo fare altro che rimuovere dalla visualizzazione ciò che non ci interessa. Per cercare quindi i timestamp relativi, ad esempio, ad attività di inserimento di periferiche USB nel PC, possiamo evidenziare il termine "USBSTOR", oppure mantenere le righe con il termine e filtrare il resto. Le possibilità sono numerose, durante l'uso pratico si impareranno via via le scorciatoie per "arrivare al sodo", tralasciando le decine di migliaia di timestamp superflui.

Come riferimento, le fonti da cui vengono estratti i timestamp, che possono essere poi utilizzate per filtrare in fase di analisi ed orientarsi soltanto su specifiche aree di azione, sono riportate nella tabella seguente.

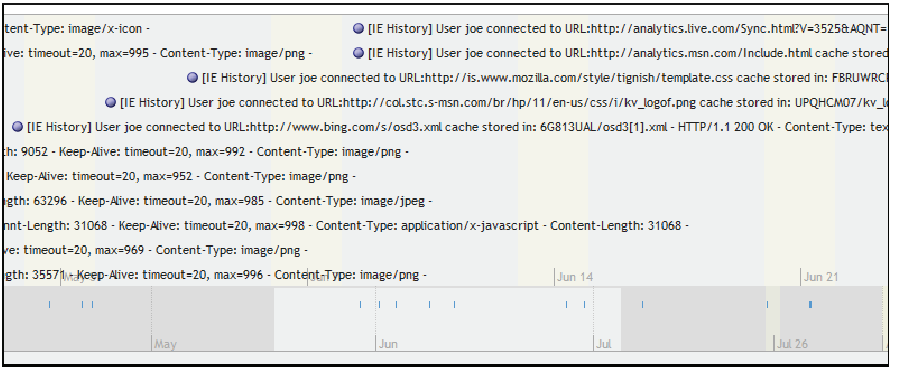
Apache2 Access Log File	NTFS $MFT	PDF Metadata
Apache2 Error Log File	MSSQL ErrorLog	$st . Prefetch
Chrome History	NTUSER key	$Recycle.bin
Encase Imager FolderPath	$ktype .	$Recycle.bin
Event Log	key	Restore Point
$sys{channel}	$type . key	Safari history
EXIF metadata	FileExts key	SAM key
Firefox	NTUSER key	SECURITY key
Firefox 2	OpenSaveMRU key	SetupAPI Log
Firefox 3 history	Map Network Drive MRU key	Skype History
FTK Imager FolderPath	MountPoints2 key	SOFTWARE key
Generic Linux Log	RecentDocs key	Flash Cookie
Internet Explorer	RunMRU key	Squid access log
IIS Log File	RegEdit key	Linux Syslog Log File
ISA text export	UserAssist key	SYSTEM key
NTFS Change Log	Opera	Shortcut LNK
MACTIME	Open XML Metadata	WMIprov Log file
McAfee AV Log	PCAP file	XP Firewall Log

Un'altra modalità di visualizzazione e filtraggio consiste nell'importare in Excel (o l'equivalente "Calc" in Open Office o "Numbers" su Mac OS) ed elaborare il file tramite formule/script. Come nel caso del "grep", una delle funzioni più utili sarà quella di filtrare in base al periodo temporale e al tipo di artefatto descritto dai numerosissimi timestamp acquisiti.

Per chi vollesse spingersi ancora oltre, è possibile visualizzare graficamente le timeline prodotte tramite log2timeline, impostando l'output nei due seguenti formati:

- XML: per strumenti quali Smile[44] timeline widget, Webscavator[45] o il prototipo CyberForensics TimeLab
- TDF: per BeeDocs[46] su Mac OS

Vediamo come esempio, qui di seguito, un estratto di supertimeline visualizzato tramite Smile a partire da un output XML del tool log2timeline.



Si raccomanda, nel caso in cui si desideri visualizzare graficamente una supertimeline, di filtrare tutti i timestamp superflui, mantenendo soltanto un esiguo numero di linee rilevanti, per non sovraffollare l'area grafica della timeline che altrimenti risulterebbe troppo pesante da leggere.

5.14 Note pratiche sulla sincronizzazione temporale

Per concludere il case study presentiamo alcune note pratiche sulla

[44] http://simile.mit.edu/
[45] http://webscavator.org
[46] http://www.beedocs.com/

problematica della sincronizzazione temporale dei reperti acquisiti. Il problema diventa importante quando si comincia ad analizzare i timestamp in essi contenuti.

Proprio per l'arbitrarietà, già descritta, dell'impostazione degli orologi dei sistemi di cui si effettua l'analisi (sia essa avvenuta per manipolazione, errato fuso orario, clock skew, etc...), può essere utile, quando possibile, verificare l'orario del sistema e confrontarlo, in fase di acquisizione, con uno attendibile. Nel caso in cui l'orario di sistema del dispositivo, che si sta acquisendo, differisca da quello che consideriamo attendibile, è sufficiente tenere traccia della differenza temporale in modo da poterla utilizzare come "delta" costante di correzione in fase di analisi.

Non sempre questa precauzione è fattibile per i motivi più svariati. Può non essere possibile avere accesso all'orario impostato sui dispositivi[47], anche avendo rimosso tutte le unità disco dal sistema, oppure i reperti su cui andiamo a eseguire copia forense possono essere rimasti troppo tempo inattivi ed aver perso la sincronizzazione temporale. Ancora, possiamo aver ricevuto l'incarico di analizzare copie forensi eseguite da terzi durante l'acquisizione delle quali non è stata tenuta traccia del disallineamento temporale.

Dando per scontato che rimarranno comunque, nei casi in cui non si è certi dell'allineamento temporale del sistema in analisi, delle riserve, è possibile adottare alcuni accorgimenti che permettono non di avere una certezza, ma di aumentarne quanto meno l'attendibilità.

Una possibile verifica può essere effettuata attraverso il confronto tra il timestamp exif delle fotografie e quello di eventuali orologi (sveglie, orologi da parete, campanili, etc...) inquadrati nelle fotografie stesse. Non si otterrà, ovviamente, una granularità al secondo, e neanche al minuto, ma si garantisce che in più di un'occasione è stato possibile, tramite questo "espediente", appurare che non fossero presenti sul sistema macro-errori temporali.

Per quanto riguarda la corretta impostazione del fuso orario e della conseguente ora legale/solare, la verifica si può fare durante la fase di analisi parificando, come vedremo in seguito, la chiave di registro contenente le informazioni sulle impostazioni temporali di Windows.

Ancora, come accennato in precedenza, si possono utilizzare gli header delle email, impostati dai server di rete, per valutare i macroerrori nell'impostazione locale.

Un'ottima e originale idea per verificare che la data e l'ora di sistema siano

[47] È possibile rilevare l'orario di sistema dei dispositivi, senza in genere compromettere la prova, accedendo al BIOS del sistema (quando non protetto da password) oppure durante la copia forense stessa, se ci si sta avvalendo di Live CD quali DEFT o Caine.

corrette, senza avere a disposizione il sistema originale, viene dal podcast Forensic4cast[48] di Lee Whitfield. Lee ha notato che, quando ci si trova all'interno di Google+, il click su di un link esterno alla piattaforma, condiviso da qualcuno della nostra cerchia, non porta il browser ad accedere direttamente alla pagina linkata, ma causa un redirect ad un'URL generata dinamicamente. Tale URL, che compare nella barra di navigazione Internet per così pochi istanti da passare in genere inosservata, contiene tra i parametri un valore numerico particolare: il timestamp UNIX in millisecondi e riferito al GMT del momento in cui si è cliccato sul link. La particolarità di tale timestamp risiede nel fatto che viene generato dinamicamente dal server G+ e non dal client: questo significa che è attendibile, anche indipendente dall'impostazione dell'ora sul PC. Come si sarà intuito, l'idea è quella di utilizzare il timestamp UNIX GMT inserito automaticamente da Google nei link Google+, per confrontarlo con quello presente nei timestamp della navigazione del browser del sistema in analisi, rilevando e quantificando così le incongruenze temporali.

Ad esempio, cliccando sul link al blog www.forensicator.com, condiviso da qualche contatto presente in una vostra cerchia, noterete che il browser visualizzerà per pochi istanti un indirizzo simile al seguente[49] sulla barra di navigazione.

http://plus.google.com/url?q=http://www.forensicator.com&ei=9AJhTuTGA4bu-gbEjM3_Dw&sa=X&ust=1314983156400057

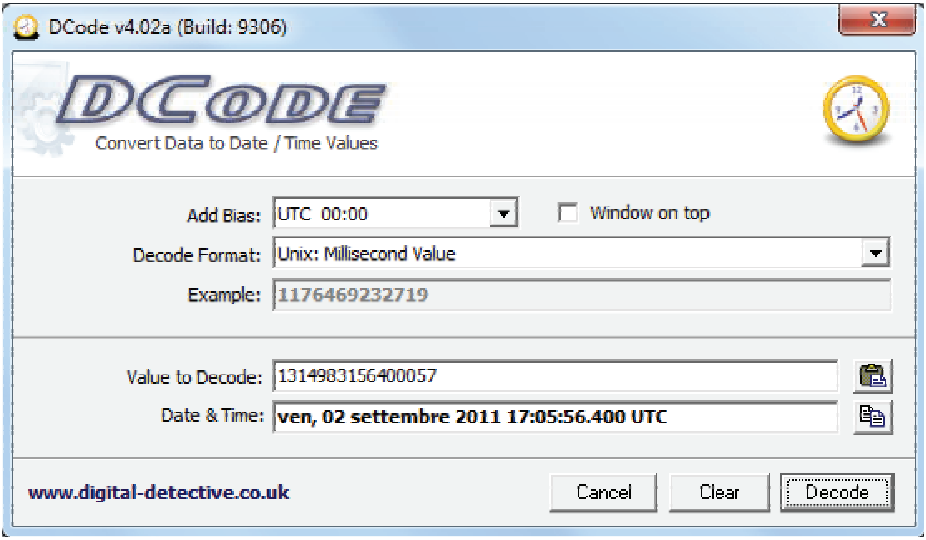
La redirezione sarà così veloce che vi ritroverete direttamente sul sito www.forensicator.com, il cui indirizzo comparirà anche nella barra di navigazione al posto del redirect mostrato sopra (che sarà stato però ormai memorizzato nella history del browser...).

Bene, il parametro in fondo, "1314983156400057", inserito dinamicamente dal server Google nel momento in cui si clicca sul link uscente dalla piattaforma Google+, non è altro che l'orario GMT UNIX in millisecondi del momento in cui è avvenuto il click, come si può vedere dallo screenshot seguente del tool di conversione temporale Dcode[50].

[48] http://www.forensic4cast.com/2011/07/flashpost-google-plus-artefacts-url-forwarding/

[49] Sembra che in alcuni casi venga utilizzato il dominio www.google.com, come prova il fatto che sostituendo plus.google.com con www.google.com nelle URL generate dinamicamente il redirect continua a funzionare.

[50] http://www.digital-detective.co.uk/freetools/decode.asp



Ai lettori, che volessero provare a copiare il link sopra riportato sul loro browser, anticipiamo che il risultato sarà leggermente diverso. Questo non perché il meccanismo riportato non sia funzionante, bensì perché il Google+ effettua la redirezione diretta soltanto se il link viene richiesto dal browser su cui è presente il cookie con dominio "google.com" ricevuto nel momento il cui la redirezione dinamica è stata generata. Sostanzialmente, il link provocherà redirezione diretta soltanto se viene cliccato direttamente dall'ambiente Google+ o dallo stesso browser sui cui si è acceduto all'ambiente. Se il link viene acceduto altrove (ad esempio se viene spedito via email o riportato in un articolo del Memberbook e copiato dai lettori sul browser) genererà una pagina intermedia, intitolata "Redirect Notice". In tale pagina Google avverte che il link sta generando una redirezione verso l'indirizzo destinazione (riportato chiaramente nella pagina) fornendo all'utente la scelta di proseguire e visualizzare la pagina linkata oppure annullare e tornare indietro. Probabilmente tale precauzione è dovuta al fatto che se la redirezione fosse sempre diretta il link potrebbe essere usato per attacchi di phishing o creare comunque disorientamento nell'utente. Per chi fosse interessato, uno dei prossimi post del blog www.forensicator.com tratterà proprio dell'analisi di questa curiosa caratteristica della piattaforma Google+.

L'aspetto notevole, ribadiamo, è che il codice temporale viene inserito nel link direttamente dal server Google+, indipendentemente dall'orario impostato sul proprio client. Se si modificasse l'ora di sistema e si cliccasse sul link, si noterebbe una discrepanza fra il codice inserito dai server Google (che si presume corretto) e l'orario di sistema. Se G+ dovesse avere diffusione pari anche soltanto a 1/10 di quella ottenuta da Facebook, sarebbe un bell'aiuto

per le indagini informatiche, così come ora lo sono tutti gli artifacts lasciati dall'utilizzo quotidiano della piattaforma di Mark Zuckerberg.

6. VALORE PROBATORIO

Parlare della prova o delle fonti di prova, nell'ambito di un processo penale, significa aprire uno dei capitoli più ampli e dibattuti atteso che, intorno alle stesse, ruota una copiosa dottrina e giurisprudenza da non poter esaurire il discorso in pochi spunti, quali sono e devono intendersi quelli seguenti.

Questo paragrafo, pertanto, è stato redatto a corredo dell'egregio studio sopra esposto e focalizzato su un aspetto molto peculiare, quello della timeline nell'ambito delle indagini di computer forensics. Si tenterà pertanto di fornire qualche breve indicazione di natura giuridico processuale che possa essere utile per il consulente tecnico di parte o il perito del Tribunale.

Secondo il codice di rito (così da intendersi nel gergo il codice di procedura penale), se all'organo giudicante (Tribunale in veste monocratica o collegiale) è riservato il compito di prendere delle decisioni (ordinanze o sentenze), alle parti: pubblico ministero, parte civile, persona offesa, imputato (e suoi difensori) è affidato il compito, a volte gravoso, di ricercare le fonti di prova, di richiederne l'ammissione in giudizio, e di attendere che, nel contraddittorio (dibattimento o incidente probatorio), si "formi" la prova.

Il giudice pertanto, nell'accertare se è avvenuto o meno un fatto storico e se l'imputato possa essere giudicato responsabile dello stesso, esamina l'insieme degli indizi e delle prove emerse nel corso del dibattimento e valuta se, il fatto, così come ricostruito dal processo penale, sia conforme al "fatto tipico" che è quello previsto e punito dalle vigenti leggi penali. Dall'analisi compartiva di tutte le circostanze (*latu sensu*) emerse nel corso del giudizio, l'organo giudicante emetterà una sentenza che sarà motivata a seconda del convincimento che lo stesso organo giudicante ha maturato nel corso del giudizio; maturazione che è frutto dell'analisi comparata e ragionata degli elementi emersi nel corso dello stesso.

La decisione (o sentenza) che il magistrato è chiamato per suo ufficio ad emettere quindi, si fonda sulle prove che sono state acquisite nel corso del processo e che sono emerse durante il dibattimento (escussione testimoniale, esame dell'imputato, prove documentali, perizie, etc.).

La prova quindi, potrebbe essere definita come quel procedimento logico in base al quale, da un fatto noto, si deduce l'esistenza di un fatto ignoto, e le modalità con le quali lo stesso si è verificato.

Il risultato di una prova deve però anche essere contestualizzato ed analizzato comparandolo con i risultati delle altre prove emerse nel corso del processo; si parla infatti di "quadro probatorio", intendendo con questa denfinizione, l'ampia gamma di materiale posto all'attenzione del giudice e di cui lo stesso dovrà tener conto nella motivazione della sentenza che andrà ad emettere. Se dall'intero quadro, emerge una contraddizione, questa deve essere in qualche modo risolta nel ragionamento e nella motivazione della sentenza emessa dal Tribunale.

Al fine di evitare confusioni terminologiche, oggi occorre distinguere tra:

1) **fonte di prova**, che è tutto ciò che è idoneo a fornire un elemento di prova (una persona o una cosa);

2) **mezzo di prova**, che è invece lo strumento con il quale si acquisisce all'interno del processo un elemento utile a fondare la decisione (ad esempio, un mezzo di prova è la testimonianza);

3) **elemento di prova**, che è l'informazione che si ricava dalla fonte di prova;

Il giudice, come dicevamo, valuta la credibilità della fonte e l'attendibilità dell'elemento ottenuto, ricavando dal suo ragionamento un **risultato probatorio che sarà riportato nel corpo della sentenza, nella sua *argomentatio***.

L'**indizio** invece è quel procedimento mediante il quale, partendo da un fatto provato (una circostanza indiziante), si ricava, attraverso massime di esperienza o leggi scientifiche, l'esistenza di un fatto storico da provare.

L'indizio non è una prova minore ma è una prova che merita la necessità di una verifica; esso è idoneo ad accertare l'esistenza di un fatto storico di reato solo quando sono presenti altre prove che escludono una diversa ricostruzione dell'accaduto.

Secondo l'art. 192 comma 2 del codice di procedura penale "L'esistenza di un fatto non può essere desunta da indizi a meno che questi siano gravi, precisi e concordanti."

Occorre allora capire, affinché un indizio abbia una sua rilevanza probatoria,, come si devono intendere queste tre caratteristiche intrinseche ed imprescindibili della gravità, precisione e concordanza. L'analisi deve essere compiuta attraverso lo studio della giurisprudenza e della dottrina maggioritaria, la quali ci permettono oggi di asserire che:

- la gravità degli indizi riguarda il grado di convincimento: è grave l'indizio resistente alle obiezioni;
- la precisione degli indizi si ha quando gli stessi non sono suscettibili di differenti interpretazioni;
- infine, sono concordanti gli indizi, quando convergono tutti verso la

medesima conclusione.

Gli indizi quindi, non si contano, ma si valutano, all'interno del processo.

Se l'oggetto della prova è un fatto incompatibile con la ricostruzione del fatto storico operata nell'imputazione, è sufficiente anche un solo indizio per escludere la responsabilità o colpevolezza dell'imputato: questo elemento è comunemente definito alibi.

L'alibi allora è quella prova logica che dimostra che l'imputato non poteva essere a quell'ora sul luogo del delitto perché, in quel medesimo momento, era invece in un altro luogo.[51] In tal caso può avvenire che un solo indizio sia idoneo a dimostrare che il fatto non si è verificato secondo il costrutto accusatorio avanzato dal pubblico ministero. Naturalmente la circostanza indiziante sulla quale si fonda l'alibi (un testimone, una prova documentale, una prova informatica), come ogni altro elemento di prova, deve essere sottoposto al vaglio di attendibilità del Tribunale.

Una lettura attenta del file di log, degli orari incrociati dei server e dei computer locali, dell'analisi dei vari elementi di natura informatica o digitale, adeguatamente e correttamente inserite in un arco temporale, come egregiamente descritto nei paragrafi precedenti di questo capitolo, consentirà di poter escludere, o comprovare, la presenza di Tizio, Caio o Sempronio, in un determinato luogo ad una determinata ora. Come ad esempio, potrebbe essere utile verificare se l'imputato potesse o meno trovarsi davanti al computer di casa o di ufficio il giorno x o y.

Non dimenticandosi che, in questi contesti, non è soltanto utile verificare se il computer fosse acceso o spento o se fosse utilizzato in quel determinato momento, ma soprattutto, se l'imputato o l'indagato, fosse l'unica persona ad avere utilizzo ed uso del computer oppure se l'uso del terminale fosse comunque promiscuo. Una buona perizia informatica non può non tenere conto della necessità di incrociare le risultanze digitali con quelle della vita reale.

In merito all'alibi, occorre anche differenziare tra l'alibi cosiddetto "fallito" (e cioè è fallito il tentativo di fornire un alibi in quanto non si è riusciti a provare un determinato fatto) che è irrilevante ai fini di un giudizio di colpevolezza, e l'alibi cosiddetto "falso", che invece assume rilevanza in merito alla credibilità dell'imputato stesso.

La Corte di Cassazione[52] a Sezioni unite, nel 1992 ha evidenziato, con un'importante pronuncia, che, l'aver appurato la falsità dell'alibi, non può di per sé determinare un'inversione dell'onere della prova, "costituendo prova della

[51] Paolo Tonini, "Manuale di Procedura Penale".

[52] Corte di Cassazione, Sezioni Unite, 21 ottobre 1992, Bompressi, in Foro It. 1993, II, 309.

verità del fatto dedotto dall'accusa" e quindi esonerando la pubblica accusa dal dover comunque provare positivamente il suo assunto o teorema. La dottrina prevalente infatti predilige l'irrilevanza probatoria di entrambe le situazioni sopra indicate e fa discendere tale assunto dalla sempre valente presunzione di innocenza e dal diritto alla difesa.[53]

7. CONCLUSIONI

Oggi l'informatica gioca un ruolo sempre più determinante all'interno dei processi, abbiamo visto come, nel processo di Duisburg ed in altri casi di omicidio verificatosi recentemente in tutta la penisola, l'utilizzo delle tecnologie (ad esempio per l'alterazione di una data di una prova documentale tecnologica, o per la prova del fatto che l'imputato all'ora del delitto fosse davanti al proprio computer a lavorare, o per la prova del fatto che la telefonata x fosse partita da un cellulare che al momento della stessa non era nella disponibilità del titolare) ci insegna come è impossibile oggi pensare di poter affrontare un processo penale senza doversi confrontare, sempre di più, con un'analisi attenta e peculiare delle informazioni digitalizzate.

Di conseguenza, il primo obiettivo di una attenta analisi di computer forensics deve essere indirizzato a rimettere in ordine gli eventi connessi all'uso di quella specifica tecnologia, secondo il loro ordine naturale. Ricostruire il puzzle per dare ad ogni pezzo il proprio posto di appartenenza.

Riuscire a fare questo significa già consentire agli investigatori la possibilità di operare su un piano di maggiore attendibilità e certezza in merito a tutte le altre risultanze investigative che emergeranno, poiché tutte riconducibili ad una comune timeline e quindi ad una precisa ricostruzione degli avvenimenti.

Si auspica sempre che l'operato di chi agisca in sede investigativa non sia mirato ad esaltare le proprie capacità e competenze professionali, ma animato sempre dal tentativo di portare luce dove è buio, di rischiarare gli avvenimenti del passato, per consentire, a chi opera nel presente e deve giudicare per il futuro, una chiara e serena interpretazione dei fatti, avvenuti ieri ma necessari per costruire il domani.

[53] V. Grevi: "Nemo tenetur se detegere. Interrogatio dell'imputato e diritto al silenzio nel processo penale italiano", Milano, 1972, 54.

8. BIBLIOGRAFIA

Florian Buchholz, Brett Tjaden: A brief study of time

Weil Michael C.: Dynamic time & date stamp analysis

Stevens Malcolm W.: Unification of relative time frames for digital forensics

Schatz Bradley, Mohay George, Clark Andrew: A correlation method for establishing provenance of timestamps in digital evidence

Paxson V: On calibrationg measurements of packet transmit times

Renico Koen, Martin S. Olivier: The Use of File Timestamps in Digital Forensics

Svein Yngvar Willassen: Finding Evidence of Antedating in Digital Investigations

Liu Naiqi, Wang Zhongshan, Hao Yujie, QinKe: Computer Forensics Research and Implementation Based on NTFS File System

Jewan Bang, Byeongyeong Yoo, Jongsung Kim, and Sangjin Lee: Analysis of Time Information for Digital Investigation

Simson L. Garfinkel: Automating Disk Forensic Processing with SleuthKit, XML and Python

Neil C. Rowe and Simson L. Garfinkel: Global analysis of drive file times
K.P. Chow, Frank Y.W. Law, Michael Y.K. Kwan, Pierre K.Y. Lai: The Rules of Time on NTFS File System

Liu Zhi jun, Zhang Huan guo: Time Bounding Event Reasoning in Computer Forensic

E. Earl Eiland: Time Line Analysis in Digital Forensics

Chen Lin, Li Zhitang, Gao Cuixia: Automated Analysis of Multi-source Logs for Network Forensics

Michael D. Kelly and Sean J. Geoghegan: FIRST: Forensic Internet Replay Sequencing Tool

Jorge Herrerias, Roberto Gomez: Log Analysis towards an Automated Forensic Diagnosis System

Joshua Ojo Nehinbe: Log Analyzer for Network Forensics and Incident Reporting

Ali Reza Arasteh, Mourad Debbabi*, Assaad Sakha, Mohamed Saleh: Analyzing multiple logs for forensic evidence

Rich Murphey: Automated Windows event log forensics

Mahbub Ahmed, Yang Xiang, Shawkat Ali: Above the Trust and Security in Cloud Computing: A Notion towards Innovation

A. Arona, D. Bruschi, E. Rosti: Adding Availability to Log Services of Untrusted Machines

Zhongli Liu, Yinjie Chen, Wei Yu and Xinwen Fu: Generic Network Forensic Data Acquisition from Household and Small Business Wireless Routers

Zhenhua Tang, Hong Ding, Ming Xu, Jian Xu: Carving the Windows registry files based on the internal structure

Paolo Tonini, Manuale di Procedura Penale, Giuffrè

V. Grevi: Nemo tenetur se detegere. Interrogatio dell'imputato e diritto al silenzio nel processo penale italiano, Milano 1972, 54

Corte di Cassazione, Sezioni Unite, 21 ottobre 1992, Bompressi, in Foro It. 1993, II, 309

Mark Hallman: Time line creation and analysis using Log2Timeline