Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il Management

Una proposta di logiche di correlazione per artefatti forensi utilizzabili nell'ambito dell'analisi live.

Relatrice:
Chiar.ma Prof.ssa
Raffaella Brighi

Presentata da:
Riccardo Mioli

Sessione II
Anno Accademico 2019/2020

Introduzione

L'avvento delle nuove tecnologie, la loro capillarità e flessibilità hanno contribuito, negli ultimi cinquant'anni, a quella che potremmo identificare come una generale tendenza all'informatizzazione della società in ogni suo settore.

L'informazione ha subito profondi mutamenti per quanto riguarda le modalità di produzione, acquisizione, trasmissione, conservazione e fruizione; ciò è stato permesso da una evoluzione tecnica che definire celere sarebbe a dir poco riduttivo.

Le tecnologie informatiche sono divenute sempre più accessibili e semplicemente utilizzabili; al contempo sono cresciute anche la dipendenza da esse e, mai come oggi, la quotidiana necessità di fruire costantemente di dispositivi, siano essi computer portatili, workstation, smartphone o tablet, che permettano di accedere a contenuti digitali nelle loro forme più varie.

Era dunque solo questione di tempo, poco per la verità, prima che i calcolatori venissero usati per compiere reati o divenissero fonti di prova degli stessi.

A seguito delle prime indagini ad oggetto informatico sorse quindi la necessità di creare un *corpus* di norme giuridiche e di buone pratiche che garantissero la scientificità di un campo in costante evoluzione, nel quale la tecnica non poteva essere lasciata incontrollata e in cui il diritto necessitava di strumenti adatti al dominio tecnologico.

La nascita di quella che oggi assume il nome di informatica forense è stata dunque la conseguenza di questo sodalizio.

Con l'aumento della capienza delle memorie di massa e dell'utilizzo dei calcolatori, uno degli aspetti più problematici delle indagini è divenuta la fase di analisi dei dati.

La complessità dei moderni sistemi e la mole di dati che un esaminatore si trova a dover analizzare richiedono tempi di lavoro che mal si sposano con la rapidità richiesta

dall'ambiente in cui esso si trova ad operare.

Per rispondere a tale necessità sono stati effettuati tentativi spazianti dalla proposta di modelli di indagine allo sviluppo di soluzioni software per automatizzare la fase di analisi. Tuttavia gli applicativi disponibili ad oggi offrono una scarsa possibilità di personalizzazione e flessibilità; ciò mina di fatto la riutilizzabilità di tali strumenti in differenti contesti di indagine.

L'obiettivo del lavoro di tesi sarà quindi quello di sviluppare alcune logiche di correlazione, riutilizzabili, semplicemente adattabili ed espandibili, allo scopo di fornire un valido strumento a supporto delle indagini.

A tal fine, il presente elaborato si articola in tre capitoli.

Il primo capitolo intende presentare una puntuale analisi di quelle che sono le cornici giuridiche e tecniche in cui l'informatico forense muove il suo operato; tale inquadramento risulta necessario al fine di comprendere i requisiti minimi necessari per un corretto trattamento delle fonti di prova e i rischi che si possono correre qualora le c.d. *best practices* di settore non fossero rispettate.

Nel secondo capitolo sarà trattato l'attuale stato dell'arte inerente all'analisi automatizzata; saranno prese in esame alcune proposte provenienti dal mondo accademico e altre di natura commerciale.

Successivamente verrà presentato Velociraptor, una potente e innovativa piattaforma *open source* per il monitoraggio degli *endpoint* e per le indagini forensi, tramite cui verranno programmate, e puntualmente illustrate, le logiche di correlazione da lanciare sui calcolatori in analisi.

Nel terzo capitolo, infine, quanto progettato e sviluppato verrà testato su alcuni malware reali e in un caso di "movimento laterale"; ciò consentirà di valutare l'effettiva utilizzabilità, praticità e funzionalità delle soluzioni codificate mediante Velociraptor.

Indice

Introduzione **i**

1 L'informatica forense **1**

1.1 Un mondo che cambia 1

1.1.1 Il fenomeno della informatizzazione 1

1.1.2 *Computer crimes* 2

1.1.3 La disciplina e le sue ramificazioni 3

1.2 Principi e obiettivi dell'informatica forense 7

1.2.1 Sulla modificabilità del dato 7

1.2.2 Evitare una cattiva pratica e una cattiva scienza 9

1.2.3 Norme del diritto 10

1.2.4 Standard tecnici 15

1.3 Metodi dell'informatica forense 20

1.3.1 Copia forense: la giurisprudenza 20

1.3.2 Acquisizioni *post mortem* e acquisizioni *live* 21

2 Logiche di correlazione **25**

2.1 La fase di analisi: il problema dei dati 25

2.1.1 Il triage 26

2.1.2 Stato dell'arte dell'analisi automatizzata 28

2.1.3 Velociraptor 32

2.2 Progettazione e sviluppo 34

2.2.1 BasicProgramInspection 35

2.2.2 ImprovedLateralMovement 42

3 Test delle logiche di correlazione **47**

3.1 Setup dell'ambiente 47

3.2 Test BasicProgramInspection 49

3.2.1 EcV01.04.R.exe 49

3.2.2 figg.exe 53

3.3 Test ImprovedLateralMovement 55

3.3.1 Infection Monkey 56

3.3.2 APTSimulator 60

Conclusioni **61**

Bibliografia **65**

Capitolo 1

L'informatica forense

1.1 Un mondo che cambia

In questa sezione saranno affrontati il contesto di nascita dell'informatica forense ed alcuni aspetti di informatica giuridica attinenti ai reati informatici; in tal modo il lettore potrà inquadrare chiaramente gli ambiti di applicabilità della disciplina.

In seguito si fornirà una definizione della materia e si descriveranno le branche che la compongono, indicando alcune tipiche situazioni di intervento per ciascuna di esse.

1.1.1 Il fenomeno della informatizzazione

A partire dalla seconda metà del novecento la società ha assistito a un importante cambiamento socio-economico: si è infatti passati da un'economia di stampo industriale ad un'economia della conoscenza.

In questo nuovo paradigma l'informazione diviene oggetto privilegiato di scambio nonchè fondamentale risorsa strategica. Allo stesso tempo si assiste a una forte crescita del settore terziario e alla nascita di nuove tecnologie informatiche, che rivoluzioneranno per sempre il modo di interfacciarsi dell'uomo con il mondo che lo circonda.

Discutere se la rapidissima evoluzione della scienza informatica sia stata frutto delle necessità di un mondo che stava cambiando l'oggetto della sua economia o se questo mutamento sia stato dovuto alla nascita e alla diffusione di prodotti come i personal computer, e di tecnologie internet, non è obiettivo del presente lavoro. In questa sede ci

limitiamo semplicemente a prendere atto di come l'*information technology* abbia assunto un ruolo sempre più centrale da sessant'anni a questa parte.

Le tecnologie informatiche, con la loro flessibilità e scalabilità, sono state rapidamente adottate da una società sempre più orientata alla produzione e allo scambio di conoscenza. Questo ha comportato che il calcolatore divenisse il mezzo privilegiato per lo svolgimento di quasi ogni attività umana, dalla contabilità al controllo di altre macchine.

1.1.2 *Computer crimes*

Con l'utilizzo sempre più massiccio delle nuove tecnologie si è anche assistito all'avvento dei primi reati connessi all'informatica.

Possiamo effettuare una loro suddivisione in tre categorie:

- Reati informatici e telematici.
- Reati commessi a mezzo informatico.
- Reati comuni di cui è possibile trovare tracce su dispositivi digitali.

Per meglio inquadrare la differenza, e la caratterizzazione, dei tre tipi di reati appena elencati si procederà con alcune definizioni ed esempi.

Il primo caso, quello dei reati informatici e telematici, venne introdotto con la legge 23 dicembre 1993, n. 547 sui *computer crimes*. Prima di allora, nell'ordinamento italiano, non era presente alcuna disposizione specifica su questo tipo di crimini. Si definiscono quindi reati informatici i reati con elementi di "tipizzazione della tecnologia informatica, vale a dire implicanti, connessi o relativi a procedimenti di elaborazione automatizzata di dati, secondo programmi informatici"[1]. Il più classico degli esempi, nonchè reato introdotto con la legge poc'anzi citata, risulta essere l'accesso abusivo a sistema informatico, ossia l'ingresso di un soggetto terzo in un tale sistema senza che egli abbia il diritto ad esercitare tale privilegio a prescindere che ne conosca le credenziali[2].

[1]La definizione è di L. Picotti, *La nozione di criminalità informatica e la sua rilevanza per le competenze penali europee*, in *Riv. Trim. dir. Pen. Ec.*, Vol. 4, 2011, p. 827 ss.

[2]Per l'esattezza, l'art. 4 della citata legge introduce nel Codice Penale un nuovo articolo, il 615-ter, intitolato "Accesso abusivo a un sistema informatico o telematico", che così lo definisce: "chiunque

Invece, i reati del secondo caso, quelli commessi a mezzo informatico, rappresentano una categoria particolare. Essi infatti comprendono tutti gli illeciti, anche tradizionali, che possono ora essere compiuti con il supporto delle nuove tecnologie. Un esempio lampante può essere quello dello *stalking*, oggi sempre più spesso praticato sotto forma di *cyberstalking*, o l'estorsione, declinata in *sextorsion*, realizzata tramite foto e video ripresi via smartphone, sistemi di videosorveglianza o webcam.

Infine, la terza categoria di reati, ossia i reati comuni di cui è possibile trovare tracce su dispositivi informatici, è piuttosto autoesplicativa. Alcuni esempi possono essere l'evasione fiscale, di cui si potrebbero trovare prove sotto forma di fogli di calcolo su un personal computer; l'omicidio volontario, con indizi contenuti nelle registrazioni audio effettuate da un assistente vocale[3]; l'omicidio colposo, con dati relativi a un incidente stradale estrapolati dal computer di bordo di un veicolo.

1.1.3 La disciplina e le sue ramificazioni

Come si è evidenziato nella precedente sezione, i dispositivi elettronici, siano essi personal computer, assistenti vocali, computer di bordo, router, server o microcontrollori, sono oggi sempre più presenti attorno a noi e possono contenere svariati dati riguardanti un ipotetico reato.

Risulta quindi evidente il motivo della nascita dell'informatica forense e l'importanza sempre crescente che la materia si troverà ad assumere negli anni a venire a causa della diffusione massiva dei dispositivi elettronici alla quale stiamo assistendo.

Giunti a questo punto possiamo quindi definire l'informatica forense, anche se forse sarebbe più corretto parlare di *Digital Forensics* vista l'estensione della materia, come la disciplina "che applica tecniche scientifiche e analitiche alle reti, ai dispositivi digitali e ai file per individuare, estrarre, elaborare, conservare dati digitali che possano essere

abusivamente si introduce in un sistema informatico o telematico protetto da misure di sicurezza ovvero vi si mantiene contro la volontà espressa o tacita di chi ha il diritto di escluderlo [...]".

[3]Per approfondimenti sulla fattispecie vedi `https://web.archive.org/web/20200208105432/https://www.ansa.it/sito/notizie/tecnologia/hitech/2019/11/07/omicidio-in-florida-alexa-testimone_8ab45007-2f20-440a-8e5e-cb531a1c40da.html`.

valutati come prova nel procedimento"[4], oppure, secondo un'efficace sintesi, che "studia le norme giuridiche e le tecniche informatiche per il trattamento dei dati digitali a fini processuali"[5].

Questa branca della scienza forense, che si colloca nell'ambito dell'informatica giuridica, ha per oggetto di studio sia le norme di diritto sostanziale e processuale che riguardano l'utilizzo di sistemi informatci e telematici nonchè i dati digitali, sia le tecniche informatiche in senso lato purchè finalizzate al trattamento dei dati digitali per trarne informazioni utili alla ricostruzione dei fatti processualmente rilevanti e con modalità conformi alle norme giuridiche, sia gli standard internazionali pubblicati dalla ISO/IEC[6]. Inoltre ricopre svariati ambiti tecnici che vanno dalla *Disk Forensics* alla *Embedded Forensics* passando per scenari di alta complessità come quelli della *Mobile Forensics*, *Network Forensics* o *Cloud Forensics*.

Disk Forensics

La *Disk Forensics* è stato forse il primo settore della *Digital Forensics* a essersi sviluppato e ha come obiettivo l'analisi delle memorie di massa (nastri magnetici, HDD, SSD, chiavette USB, SD card, etc.) al fine di estrarre i dati memorizzati in esse.

Grazie all'analisi dei supporti di memorizzazione secondaria è possibile recuperare non solo i file attualmente presenti nello spazio allocato del sistema, ma, a meno che non

[4]R. Brighi, *Una governance integrata per nuovi modelli dell'informatica forense*, in *Journal of Law, Cognitive Science and Artificial Intelligence (i-lex)*, 2017, pp. 47-70. Un'altra definizione è quella di C. Maioli: "la disciplina che studia l'insieme delle attività che sono rivolte all'analisi e alla soluzione dei casi legati alla criminalità informatica, comprendendo tra questi i crimini realizzati con l'uso di un computer, diretti a un computer o in cui il computer può rappresentare una fonte di prova"(C. Maioli in *Dar voce alle prove: elementi di informatica forense*, 2004, p. 1). Propendono per una definizione più ampia G. Fagioli e A. Ghirardini per i quali l'informatica forense consiste nella "scienza che disciplina le metodologie per la preservazione, l'identificazione e lo studio delle informazioni contenute nei computer o nei sistemi informativi in generale, al fine di evidenziare l'esistenza di prove utili allo svolgimento dell'attività investigativa" (G. Fagioli, A. Ghirardini, *Digital forensics*, Apogeo, 2013).

[5]A. Gammarota, *Informatica forense e processo penale: la prova digitale tra innovazione normativa e incertezze giurisprudenziali*, Tesi di Dottorato, Alma Mater Studiorum - Università di Bologna, 2016, pp. 26-27.

[6]Cfr. ibidem.

siano state messe in atto tecniche di "anti-forense" o non sia trascorso troppo tempo, anche quelli nello spazio non allocato, ossia file eliminati o parzialmente sovrascritti. In questa seconda ipotesi si procede tramite l'analisi degli *slack space*[7].

Embedded Forensics

L'*Embedded Forensics* si occupa dell'analisi dei sistemi integrati, ovvero sistemi non riprogrammabili studiati per svolgere una specifica funzione.

Alcuni esempi classici possono essere le scatole nere montate su aerei o i computer di bordo che, sempre più spesso, vengono installati sulle automobili per svolgere funzionalità di *infotainment*.

È altamente probabile che questo settore dell'informatica forense conoscerà una poderosa crescita in futuro per via del sempre maggiore utilizzo che viene fatto di sistemi integrati nella vita quotidiana: si pensi ai dispositivi IoT, alle macchine a controllo numerico, console per videogiochi, ASIC[8], e così via.

Mobile Forensics

La *Mobile Forensics* si occupa dell'estrazione e della conservazione di dati da dispositivi mobili quali gli smartphone.

Questo ramo dell'informatica forense presenta alcune difficoltà date dai sistemi di sicurezza adottati sui moderni telefoni cellulari. Emblematico, e fra i più noti, è il caso di

[7]Quando un *file* viene eliminato lo spazio da esso occupato sulla memoria di massa viene contrassegnato come libero. I dati rimangono memorizzati, ma non vengono "visualizzati" dal *file system*. Nel momento in cui nuove informazioni vengono scritte sul settore del disco contrassegnato come disponibile queste potrebbero richiedere solo una frazione dello spazio di tale settore; qualora ciò accadesse il rimanente spazio conterrebbe ancora le informazioni relative al file precedentemente memorizzato. Per approfondimenti si veda M. Ferrazzano, *Indagini forensi in tema di scambio di file pedopornografici mediante software di file sharing a mezzo peer-to-peer*, Tesi di Dottorato, Alma Mater Studiorum - Università di Bologna, 2014, p. 34.

[8]Gli *Application Specific Integrated Circuit* sono circuiti non riprogrammabili studiati per svolgere una specifica funzione. Un esempio possono essere i *controller* contenuti nei lettori DVD.

San Bernardino in cui l'FBI ha dovuto richiedere ad Apple aiuto, negato poi dall'azienda, per accedere ai dati presenti su un dispositivo appartenuto ad un terrorista[9].

Ove si consideri il grande numero di dati e informazioni personali presenti su dispositivi che ci seguono costantemente, appare chiaro il motivo per cui in questo segmento si stiano concentrando sforzi e ricerche sul piano tecnico.

Network e *Cloud Forensics*

La *Network Forensics* si occupa dell'analisi del traffico sulle reti informatiche, alle quali sono oggi connessi i più svariati tipi di dispositivi.

Il settore presenta sfide derivanti dal fatto che buona parte dei dati che transitano sono volatili, salvo ad esempio alcune tracce contenute in file di *log* gestiti da router, firewall o direttamente dai calcolatori o dagli applicativi interessati.

La *Network Forensics* cresce di pari passo con l'evoluzione di internet. È infatti recente la nascita della *Cloud Forensics*, che si occupa di acquisizioni da infrastrutture *cloud*, sempre più diffuse in ambito aziendale e non solo, essendo ormai comune l'archiviazione di dati personali e familiari sulla "nuvola" tramite servizi di *hosting* anche gratuiti.

Live Forensics

La *Live Forensics* è il campo della *Digital Forensics* che si occupa di estrarre dati da un sistema attivo; in tal modo è possibile ottenere informazioni attualmente non consolidate sulle memorie di massa e presenti solamente in quella primaria.

Questo settore presenta alcune problematicità che si originano dall'interazione con calcolatori attivi; le azioni dell'informatico forense possono infatti causare modifiche, che dovranno essere attentamente documentate e giustificate, ai dati presenti sul sistema.

Prospettive future

Come ribadito più volte, la *Digital Forensics* si evolve seguendo gli sviluppi della tecnica; è quindi complesso, se non impossibile, delineare con certezza quelle che saranno

[9]La vicenda è ampiamente descritta in `https://web.archive.org/web/20200719200744/https://en.wikipedia.org/wiki/FBI%E2%80%93Apple_encryption_dispute`.

le traiettorie future di sviluppo della materia, sebbene possano già individuarsene alcune. Si pensi ad esempio alle possibilità di strumenti di indagine offerti dall'intelligenza artificiale o a tutto il nuovo campo della *blockchain*.

1.2 Principi e obiettivi dell'informatica forense

Definita la materia, e inquadrate le sue diramazioni, passiamo ora ad esaminare i principi e gli obiettivi che guidano l'informatica forense.

Con riguardo in particolare a questi ultimi, essi sono stati individuati nel "conservare, identificare, acquisire, documentare e interpretare i dati presenti su un computer. A livello generale si tratta di individuare le modalità migliori per: acquisire le prove senza alterare o modificare il sistema informatico su cui si trovano, garantire che le prove acquisite su altro supporto siano identiche a quelle originarie, analizzare i dati senza alterarli. In sintesi, di 'dare voce alle prove'"[10].

Nella enunciazione delle suddette modalità si possono rinvenire anche quelli che sono i principi che devono governare la materia e che, come vedremo a breve, emergono dalle normative specifiche, tanto giuridiche quanto tecniche: in particolare, il principio basilare che potremmo definire di "non alterazione" nelle fasi di acquisizione e analisi dei dati.

L'osservanza di questo principio scaturisce da una peculiarità tipica del dato informatico: la sua modificabilità.

1.2.1 Sulla modificabilità del dato

Il dato informatico è estremamente mutevole e può essere modificato da azioni non direttamente svolte dall'utente. Ad esempio, la sola accensione di un calcolatore che è stato spento implica modifiche potenzialmente irreversibili a migliaia di file, scritture sui *log* di sistema ed altri effetti collaterali.

Allo stesso modo non è possibile visionare, in modo tradizionale, dei file pensando che l'azione non produca mutamenti sul sistema in analisi. Prendiamo, ad esempio, l'apertura di un documento testuale. Senza contare ipotetiche modifiche al contenuto,

[10]La definizione è di C. Maioli, *Dar voce...*, pp. 1-2.

Figura 1.1: Un esempio di modifiche ai metadati.

che potrebbero derivare da pressioni accidentali di tasti, spesso durante le indagini ciò che più interessa di un file sono i metadati ad esso associati.

I metadati sono dati riferiti ai file che registrano informazioni aggiuntive quali ad esempio:

- Data di creazione.
- Data di ultimo accesso.
- Data di ultima modifica.
- A seconda del sistema operativo utilizzato, altri dati comprendenti percorso, grandezza del file, creatore, permessi, etc.

Vediamo ora un esempio pratico di modifiche ai metadati visibile nella figura 1.1.

Nell'ordine: viene creato un file di nome `text.txt` e se ne stampano a schermo alcune informazioni con il comando `ls -lu`. I parametri `l` e `u`, se usati insieme, mostrano i file ordinati per nome e l'ultimo accesso che, in questo caso, risale alle 8:44.

Successivamente il file viene aperto con l'editor `nano`, semplicemente visualizzandolo senza apportare alcuna modifica, e si ripete quindi l'operazione di stampa di alcuni dei metadati: come si può notare, la data di ultimo accesso risale adesso alle 8:45, ossia al momento dell'apertura con `nano`.

1.2.2 Evitare una cattiva pratica e una cattiva scienza

Non tenere conto della modificabilità del dato informatico può condurre a conseguenze disastrose. Un esempio ne è il c.d. Caso Garlasco[11].

Durante le indagini preliminari il computer di Alberto Stasi, sul quale l'indagato affermava di stare scrivendo la tesi al momento dell'omicidio della sua fidanzata, venne analizzato dalla polizia giudiziaria che, esplorando i file contenuti sul disco, pregiudicò irrimediabilmente l'integrità dei dati in esso contenuti. Come abbiamo visto nell'esempio pratico precedente, la sola visualizzazione di un file ne avrebbe modificato i metadati relativi all'ultima apertura, rendendo di fatto non verificabile l'alibi di Stasi, così compromesso.

I motivi del tradimento nelle aule giudiziarie dei principi alla base di questa disciplina appaiono molteplici e sono da ricercarsi in una scarsa conoscenza di ciò che accade al calcolatore durante il suo utilizzo, nell'erronea e fiduciosa presunzione che tutto ciò che emerge dai dispositivi digitali sia veritiero e, infine, nella mancanza di adeguati mezzi tecnici a supporto dell'indagine. In realtà l'unica certezza che si può ragionevolmente coltivare in campo informatico è che da un dispositivo elettronico possono fuoriuscire soltanto due cose: l'effettivo contenuto delle sue memorie, le quali vanno acquisite seguendo specifiche procedure che salvaguardino l'integrità dei dati, oppure un insieme di informazioni inquinate da comportamenti maldestri.

Proprio dal pericolo di *malpractices* ha origine il quadro normativo - giuridico e tecnico - che a breve andremo ad analizzare e che, oggi, evita che si possa parlare di cattiva scienza per l'informatica forense. D'altronde essa, nata in America attorno al 1970 (all'epoca si parlava di *Computer Forensics*), ha conosciuto, soprattutto a partire dagli anni 1990, un rapido sviluppo non sempre accompagnato da chiare linee guida su quali fossero le migliori pratiche da seguire durante l'acquisizione delle fonti di prova.

Tuttavia ai nostri giorni il quadro è decisamente cambiato ed è possibile reperire agevolmente i principi tecnici da applicare e le norme giuridiche da attuare per il corretto trattamento dei dati digitali a fini processuali.

Va detto che se le norme giuridiche forniscono la guida, le norme tecniche ne rappresentano l'implementazione.

[11] Su questo caso, vedasi in particolare A. Gammarota, *Informatica forense...*, pp. 120-137.

Come vedremo, il diritto insiste frequentemente sulla necessità di adottare misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione, ma poi il legislatore non indica un modo migliore di altri *(best practices)* per l'individuazione, l'acquisizione e la salvaguardia delle prove digitali; a questo pensano gli standard tecnici. Si tratta di una soluzione che, considerata la maggiore rapidità di aggiornamento delle normative tecniche rispetto a quelle del diritto, offre agli operatori vie pratiche al passo con i tempi per evitare un trattamento scorretto dei dati informatici[12].

Gli obiettivi dell'informatica forense di garantire integrità, accuratezza e affidabilità delle fonti di prova originali - ossia di ottenere una fonte di prova di qualità - risultano così assicurati, in un mondo tecnologico in costante e rapida evoluzione.

1.2.3 Norme del diritto

La Convenzione sul Cybercrime

A livello di norme del diritto, un testo di riferimento dell'informatica forense è costituito dalla Convenzione sulla criminalità informatica del Consiglio d'Europa, aperta alla firma il 23 novembre 2001 ed entrata in vigore il 1 luglio 2004[13].

Sebbene essa risenta in certo modo del tempo trascorso, resta però di fondamentale importanza in quanto "primo strumento internazionale vincolante per affrontare in modo globale il problema, che cerca di armonizzare le leggi sui reati informatici dei vari Stati aderenti, migliorare la capacità e le modalità di indagine e accrescere la cooperazione investigativa internazionale"[14]. Infatti, i suoi obiettivi espressamente dichiarati sono:

- "Criminalizzare le infrazioni contro la riservatezza, l'integrità e la disponibilità di dati e sistemi informatici, le infrazioni associate all'informatica, le infrazioni

[12]È di questa opinione A. Gammarota, *Informatica forense...*, p. 78, per il quale appare condivisibile, alla luce del continuo progresso tecnologico, la scelta del legislatore italiano di non tipizzare specifiche tecniche di trattamento dei dati a fini processuali.

[13]La lista dei paesi che l'hanno sottoscritta, ratificata e fatta entrare in vigore è reperibile all'url `https://www.coe.int/en/web/conventions/full-list/-/conventions/treaty/185/signatures?p_auth=h4WrCTGG`.

[14]R. Brighi, C. Maioli, *Un cambio di paradigma nelle scienze forensi. Dall'armonizzazione tecnico-giuridica a una nuova cornice epistemologica.*, in *Informatica e diritto*, Vol. 24, 2015, p. 219.

associate ai contenuti (ovvero pedopornografia, razzismo e xenofobia) e le infrazioni legate alla violazione del copyright e dei diritti correlati.

- Stabilire procedure per aumentare l'efficienza delle indagini.
- Fornire una base giuridica per la cooperazione internazionale tra gli Stati parti della Convenzione, compresi gli scambi di informazioni su base spontanea, l'estradizione e l'assistenza reciproca a livello internazionale e punti di contatto disponibili 24 ore su 24, 7 giorni su 7"[15].

La "Convenzione di Budapest", com'è per lo più conosciuta, è divisa in diverse sezioni nelle quali affronta alcuni aspetti definitori di base, fissa i principi in materia di giurisdizione internazionale, regola aspetti di diritto sostanziale e procedurale[16]. In particolare, "prevedeva a carico dei Paesi aderenti l'obbligo di dotarsi di un corpus normativo che comprendesse le principali figure di reato, gli strumenti processuali per il loro accertamento, gli organi e le procedure di coordinamento e cooperazione transnazionale tra organi investigativi statali per rendere più efficace l'attività di contrasto"[17] al *cybercrime*.

Essa è di particolare interesse perchè le sue disposizioni si estendono a qualunque reato nel quale sia necessario raccogliere elementi probatori in formato elettronico[18].

Per quanto riguarda i profili più rilevanti ai fini del presente lavoro, la Convenzione "stabilisce la condotta anzichè la tecnologia, garantendo che le norme e le procedure rimangano valide con l'evolvere della tecnologia"[19].

Nella sostanza, "la Convenzione di Budapest, presupponendo le finalità da perseguire, ovvero la conservazione dell'integrità dei dati originari e la loro duplicazione a fini forensi, ha formalizzato i principi minimi, le modalità di massima per realizzarle, ma lasciando opportunamente fuori dall'ambito dell'intervento normativo tutto l'ambito tecnico-scientifico del corretto trattamento dei dati ritenuto un mero presupposto idoneo

[15]Come si legge sulla pagina web dedicata alla stessa: `http://web.archive.org/web/20191101155241/https://www.coe.int/it/web/portal/coe-action-against-cybercrime`

[16]Il testo in lingua italiana è reperibile in `http://web.archive.org/web/20201001191740/https://rm.coe.int/16802f423d`

[17]A. Gammarota, *Informatica forense...*, p. 31.

[18]Così R. Brighi, C. Maioli, *Un cambio di paradigma...*, p. 219.

[19]Sempre secondo quanto indicato in `http://web.archive.org/web/20191101155241/https://www.coe.int/it/web/portal/coe-action-against-cybercrime`

a garantire l'automatica applicazione delle tecniche che realizzano le finalità codificate"[20]. In altre parole, "la Convenzione di Budapest ha imposto ai Paesi aderenti l'adozione delle tecniche per il trattamento dei dati oggetto di investigazione e procedimento senza imporre [...] specifiche tecniche, purchè fosse raggiunto il fine ultimo costituito dalla corretta gestione del dato"[21], con ciò venendo peraltro a costituire il primo riconoscimento normativo della necessità del corretto trattamento dei dati digitali a fini di indagini[22].

Emblematico, in proposito, l'art. 19 della Convenzione, il quale prevede, in materia di perquisizione e sequestro di dati informatici, che ogni paese debba adottare misure legislative e di altra natura - si dà per scontato, tecnica - che dovessero essere necessarie per consentire alle proprie autorità competenti di sequestrare o acquisire i dati informatici. Queste misure - in base all'articolo sopra citato - devono includere, fra i vari poteri quello "di mantenere l'integrità dei relativi dati informatici immagazzinati", senza però che la norma offra ulteriori specificazioni in ordine alle metodologie tecniche per mantenere tale integrità, che peraltro costituisce nel campo della *digital evidence* uno dei principi fondamentali di cui abbiamo detto in precedenza.

Legge n. 48/2008

In Italia la Convenzione di Budapest venne ratificata con la legge 18 marzo 2008, n. 48, attraverso la quale numerosi articoli del Codice Penale e del Codice di Procedura Penale subirono una revisione, optando così il legislatore italiano per l'impianto di nuove norme sulle vecchie per rispondere all'istanza di aggiornamento.

Nello specifico la legge di ratifica aggiunge alcune figure di reato, come ad esempio il danneggiamento di informazioni, dati e programmi informatici, assenti nella legge n. 547/1993[23], e soprattutto aggiorna in modo sostanziale gli articoli del c.p.p riguardanti i mezzi di ricerca della prova, "ma senza alcun intervento sui mezzi di prova a contenuto informatico nella direzione dell'implementazione della *digital evidence*"[24]. Se pertanto,

[20] A. Gammarota, *Informatica forense...*, p. 162.

[21] Ivi, p. 74

[22] Cfr. ivi, p. 108.

[23] Per una puntuale ricostruzione di tutti gli articoli del Codice Penale italiano interessati da tale legge, si rinvia a A. Gammarota, *Informatica forense...*, p. 73, nota 173.

[24] Così A. Gammarota, *Informatica forense...*, p. 73.

sotto quest'ultimo aspetto, vi è stato chi ha visto in questa legge un'occasione mancata o addirittura un salto all'indietro, non le si può negare il merito "di aver acceso una luce sul problema del trattamento dei dati digitali a fini processuali e di aver sensibilizzato gli operatori forensi sulla necessità di adeguamento delle restanti norme alle esigenze imposte dal progresso tecnologico in campo informatico"[25].

Nel modificare le norme del codice di rito penale con le previsioni relative al trattamento dei dati informatici a fini processuali, il legislatore italiano è intervenuto nell'ambito degli atti a iniziativa della polizia giudiziaria e in quello delle indagini, in particolare con riguardo ai mezzi di ricerca della prova.

Senza addentrarci in un'approfondita disamina delle modifiche[26], qui vogliamo limitarci a riportare quelle nelle quali si esplicita in modo particolarmente evidente l'affermazione (che riportiamo in corsivo) di quel principio di "non alterazione", nelle sue molteplici articolazioni, di cui abbiamo parlato. Si tratta, in particolare, degli articoli 244 (Ispezione), 247 (Perquisizione), 254 bis (Sequestro di dati informatici presso fornitori di servizi informatici, telematici, e di telecomunicazioni), 260 (Sigillo elettronico o informatico e copia dei dati), 352 (Perquisizioni), 354 (Accertamenti urgenti sui luoghi, sulle cose e sulle persone. Sequestro).

244 c.p.p. - [...] L'autorità giudiziaria può disporre [...] *ogni altra operazione tecnica, anche in relazione a sistemi informatici o telematici, adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione.*

247 c.p.p. - 1-bis. Quando vi è fondato motivo di ritenere che dati, informazioni, programmi informatici o tracce comunque pertinenti al reato si trovino in un sistema informatico o telematico, ancorchè protetto da misure di sicurezza, ne è disposta la perquisizione, *adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione.*

254-bis. c.p.p. - 1. L'autorità giudiziaria, quando dispone il sequestro, presso i fornitori di servizi informatici, telematici o di telecomunicazioni, dei dati da questi detenuti, compresi quelli di traffico o di ubicazione, può stabilire, per esigenze legate alla regolare fornitura dei medesimi servizi, che la loro *acquisizione avvenga mediante copia di essi su*

[25]Ivi, pp. 72-73.

[26]Per la quale si rinvia al più volte citato lavoro di A. Gammarota, *Informatica forense...*, pp. 78-97.

adeguato supporto, con una procedura che assicuri la conformità dei dati acquisiti a quelli originali e la loro immodificabilità. In questo caso è, comunque, ordinato al fornitore dei servizi di conservare e proteggere adeguatamente i dati originali.

260 c.p.p. - L'autorità giudiziaria fa estrarre copia dei documenti e fa eseguire fotografie o altre riproduzioni delle cose sequestrate che possono alterarsi o che sono di difficile custodia, le unisce agli atti e fa custodire in cancelleria o segreteria gli originali dei documenti, disponendo, quanto alle cose, in conformità dell'art. 259. *Quando si tratta di dati, di informazioni o di programmi informatici, la copia deve essere realizzata su adeguati supporti, mediante procedura che assicuri la conformità della copia all'originale e la sua immodificabilità*; in tali casi, la custodia degli originali può essere disposta anche in luoghi diversi dalla cancelleria o dalla segreteria.

352 c.p.p. - 1-bis. Nella flagranza del reato, ovvero nei casi di cui al comma 2 quando sussistono i presupposti e le altre condizioni ivi previsti, gli ufficiali di polizia giudiziaria, *adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione*, procedono altresì alla perquisizione di sistemi informatici o telematici, ancorchè protetti da misure di sicurezza, quando hanno fondato motivo di ritenere che in questi si trovino occultati dati, informazioni, programmi informatici o tracce comunque pertinenti al reato che possono essere cancellati o dispersi.

354. - 1. Gli ufficiali e gli agenti di polizia giudiziaria curano che le tracce e le cose pertinenti al reato siano conservate e che lo stato dei luoghi e delle cose non venga mutato prima dell'intervento del pubblico ministero. In relazione ai dati, alle informazioni e ai programmi informatici o ai sistemi informatici o telematici, gli ufficiali della polizia giudiziaria *adottano, altresì, le misure tecniche o impartiscono le prescrizioni necessarie ad assicurarne la conservazione e ad impedirne l'alterazione e l'accesso e provvedono, ove possibile, alla loro immediata duplicazione su adeguati supporti, mediante una procedura che assicuri la conformità della copia all'originale e la sua immodificabilità.*

Si nota dunque come sia sistematicamente richiamata dal legislatore l'attenzione sull'importanza della conformità del dato all'originale e della sua immodificabilità nonchè sull'importanza della sua conservazione e non alterazione - in altri termini, sull'importanza della qualità e dell'integrità del dato - da attuare mediante misure tecniche adeguate. Il legislatore però non si esprime su quali siano le procedure più adatte da seguire per

preservare i dati[27] e quindi, di fatto rinvia ad una tecno-regolazione.

Ciò da un lato garantisce una flessibilità nelle pratiche adottabili e nell'evoluzione degli standard tecnici, dall'altro lascia un vuoto normativo piuttosto insidioso considerando che, in caso di *malpractices*, non viene prevista l'inutilizzabilità della fonte di prova in dibattimento, che rimane quindi liberamente valutabile dal giudice[28].

Ecco allora che l'adozione di pratiche corrette durante il trattamento dei dati si pone anche come il baluardo contro una cattiva scienza. Per questo motivo è "strategica la definizione di protocolli operativi per gli accertamenti informatici e la documentazione delle prove"[29].

1.2.4 Standard tecnici

Come abbiamo detto, se la legge 48/2008 fissa i principi al fine di garantire l'attendibilità dei dati oggetto di trattamento a fini processuali, non fissa però le specifiche tecniche per realizzarli. Fino all'ottobre del 2012 le metodologie erano definite in alcune *best practices* del settore[30].

Forze di polizia come l'FBI o istituti come il NIST (National Institute of Standards and Tecnology) sono stati i primi a realizzare documenti che indicavano quali fossero le migliori pratiche da adottare[31].

[27]Osserva M. Ferrazzano, *Indagini forensi...*, p. 25: "Nel disciplinare il modus operandi delle operazioni può osservarsi come l'attenzione del legislatore si sia focalizzata, giustamente, più sul risultato che deve essere ottenuto piuttosto che sul metodo da seguirsi: la canonizzazione all'interno di norme giuridiche di procedure tecniche a livello informatico più che rappresentare una garanzia, avrebbe portato, alla lunga, ad effetti contrari e distorsivi rappresentati dall'evoluzione costante della disciplina e dalle peculiarità proprie di ciascun caso".

[28]La mancata espressa previsione di inutilizzabilità o, meglio, di nullità dei mezzi di prova acquisiti in violazione delle procedure previste è una delle carenze imputate dalla dottrina alla L. 48/08; per un loro esame complessivo, si veda A. Gammarota, *Informatica forense...*, pp. 100-104.

[29]R. Brighi, C. Maioli, *Un cambio di paradigma...*, p. 232.

[30]Sui limiti del ricorso a *best practices* e, più in generale, per un loro inquadramento nell'ambito dell'informatica forense si veda A. Gammarota, *Informatica forense...*, p. 97 e p. 105 ss.; per il secondo aspetto, si veda anche M. Ferrazzano, *Indagini forensi...*, pp. 25-26.

[31]È interessante notare come già nel 2004 il NIST si occupasse di stabilire le metodologie da adottare su dispositivi mobili: `https://web.archive.org/web/20200509115250/https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-72.pdf`.

Successivamente l'ISO (Organizzazione Internazionale per la Standardizzazione) e l'IEC (Commissione Elettrotecnica Internazionale), che formano il sistema specializzato per la standardizzazione a livello mondiale, hanno presentato alcuni documenti riguardanti l'informatica forense che si candidano a norme tecniche di riferimento riconosciute a livello internazionale per il trattamento della *digital evidence*, alle quali si può e deve ricorrere per attuare le norme di metodo previste dal codice di procedura penale, ma non solo[32].

In particolare, per i profili di nostro interesse, si tratta di ISO/IEC 27037:2012 e ISO/IEC 27042:2015, una valida base di condivisibili procedure operative e modalità di indagini per gli informatici forensi.

ISO/IEC 27037:2012

Lo standard ISO/IEC 27037:2012[33] ha per oggetto le "linee guida per l'identificazione, raccolta, acquisizione e conservazione delle prove digitali".

Dallo standard, che peraltro non suggerisce quali possano essere i programmi da utilizzare durante le varie fasi prese in considerazione, emerge che il trattamento della prova digitale deve seguire alcuni requisiti fondamentali:

- Rilevanza: devono sussistere buone ragioni per acquisire del materiale, che deve essere pertinente e utile all'indagine in corso.
- Affidabilità: le azioni svolte devono poter essere documentate e possibilmente riproducibili.

[32]Osservano R. Brighi, C. Maioli, *Un cambio di paradigma...*, p. 223, "Si forma, dunque, un corpus coerente che può costituire un riferimento per la conduzione di investigazioni digitali in tutti gli ambiti, quindi non solo nei processi penali ma anche in quelli civili e nelle indagini condotte internamente nelle varie organizzazioni pubbliche o private che possono anche non essere mai portati davanti a un tribunale".

[33]Reperibile in banca dati *British Standards Online* all'indirizzo `https://bsol-bsigroup-com.ezproxy.unibo.it/PdfViewer/Viewer?pid=000000000030333314`. Per una sua dettagliata analisi vedi M. Ferrazzano, *Indagini forensi...*, pp. 27-38. Lo standard era stato preceduto da una serie di altri orientati in generale al tema della sicurezza informatica: per un loro *excursus* si rimanda a O. M. Fal', *Standardization in Information Technology Security*, in *Cybernetics and System Analysis*, Vol. 53, 2011, pp. 78-82.

- Sufficienza: è importante decidere quanto materiale vada raccolto a seconda delle esigenze del caso.
- Verificabilità: utilizzando la documentazione prodotta deve essere possibile verificare le procedure svolte e valutare appropriatezza del metodo seguito.
- Ripetibilità e riproducibilità: un altro operatore, seguendo la documentazione prodotta, dovrebbe poter essere in grado di ripetere le operazioni svolte e raggiungere gli stessi risultati.
- Giustificabilità: è necessario essere sempre in grado di motivare le azioni intraprese durante il trattamento delle prove digitali.

Vediamo ora le varie fasi del trattamento della prova identificate dallo standard.

1. L'*identificazione* rappresenta il primo passo. In questo momento il *Digital Evidence First Responder*, da qui in poi abbreviato in *DEFR*, procede a identificare i dispositivi digitali che possono contenere prove. Ne sono un esempio stampanti, chiavette usb, personal computer, SD card, computer di bordo di autovetture, impianti di sorveglianza, etc.

   È di primaria importanza definire una priorità di acquisizione fra i vari dispositivi, dal momento che per alcuni i dati potrebbero essere ottenuti solo se accesi, o non essere acquisibili al passare di un determinato intervallo di tempo (e.g. impianti di sorveglianza che sovrascrivono le immagini).

   Oltre ad individuare i dispositivi digitali, potrebbe essere necessario trovare anche cavi di alimentazione proprietari e imballaggi contenenti codici di sblocco di dispositivi, come ad esempio nel caso delle SIM.

2. Una volta terminata l'identificazione segue la fase della *raccolta*, durante la quale i dispositivi vengono etichettati con un codice univoco e schedati in modo tale da documentare data, ora, nome dell'operatore che ha effettuato il prelievo e altre informazioni rilevanti.

   Il *DEFR* potrebbe anche saltare la raccolta qualora il reperto non fosse trasportabile nel laboratorio forense. È, ad esempio, il caso di server che offrono servizi non

interrompibili perchè di pubblica utilità o di macchinari sanitari che non possono essere spenti.

3. Se ci si trova in questa circostanza si procede in loco con l'*acquisizione*, ossia la la fase successiva alla copia, con l'utilizzo di specifici strumenti e programmi, del contenuto delle memorie.

   Gli applicativi e la strumentazione utilizzata dovrebbero essere il meno intrusivi possibile in modo da causare pochi o, preferibilmente, nessun cambiamento sul target dell'acquisizione. In caso le modifiche non siano evitabili, bisogna documentare adeguatamente i processi svolti, ad esempio mediante l'utilizzo di video.

   La copia ottenuta e la sorgente vanno poi verificate tramite una funzione; generalmente viene utilizzato l'hash MD5 o SHA-1, al fine di provarne l'uguaglianza e garantire l'immodificabilità della copia[34]. Qualora ciò non sia possibile, ad esempio nel caso di alcune operazioni eseguite su un sistema attivo, il *DEFR* dovrà documentare e giustificare i passi seguiti.

4. L'ultima fase, quella della *conservazione e trasporto*, consiste nel custodire le prove digitali in luoghi appositamente adibiti che tengano conto delle caratteristiche fisiche dei reperti e ne impediscano il deterioramento, ad esempio a causa di alte temperature o campi magnetici.

   Durante la conservazione e il trattamento dei reperti viene utilizzato il paradigma della *catena di custodia*. Ogni interazione è appuntata su uno specifico documento in modo tale che possa essere ricostruito chiaramente chi ha avuto accesso ai dispositivi, quando e dove ciò è accaduto, che operazioni sono state svolte e altre informazioni che possono dipendere dal modello utilizzato.

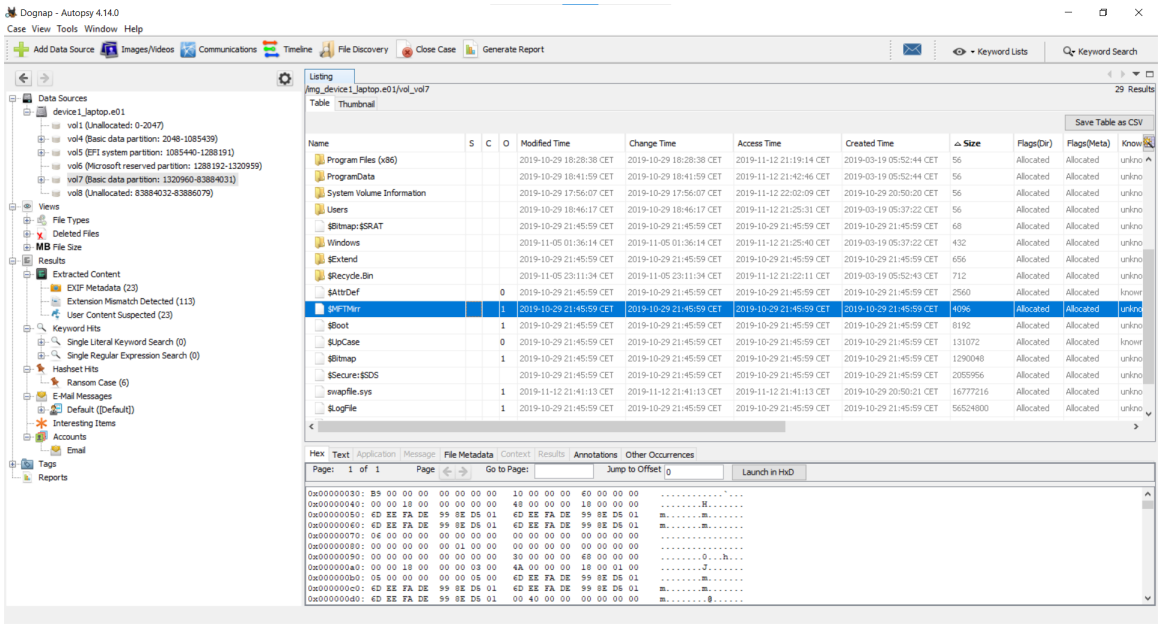

Figura 1.2: Un esempio di analisi statica con l'utilizzo di Autopsy

ISO/IEC 27042:2015

Lo standard ISO/IEC 27042:2015[35] si occupa di definire, come specificato nel titolo, le "linee guida per la analisi e l'interpretazione delle prove digitali", ossia per lo stadio immediatamente successivo a quello oggetto dello standard precedentemente illustrato.

1. L'*analisi* rappresenta la prima fase del processo: in questo momento il contenuto della memoria viene esplorato alla ricerca di dati rilevanti. Il documento distingue due principali metodologie: quella statica, condotta con l'ausilio di programmi su una copia forense acquisita in precedenza, come ad esempio Autopsy (visibile in figura 1.2), e quella *live*, che può essere svolta direttamente sui calcolatori in esame oppure utilizzando un'immagine realizzata in precedenza e mandata in esecuzione su hardware simile o emulato.

[34]Come sottolinea C. Maioli in *Dar voce...*, p. 3: "le prove stesse vanno autenticate e verificate temporalmente con opportuni programmi di utilità in modo da poter facilmente dimostrare in sede di giudizio che la operazione di riproduzione delle prove è stata eseguita nei modi e nei tempi indicati".

[35]Reperibile in `https://bsol-bsigroup-com.ezproxy.unibo.it/PdfViewer/Viewer?pid=000000000030333330`.

2. Dopo aver individuato artefatti di interesse, è necessario procedere con l'*interpretazione*. In questo momento l'operatore elabora i dati recuperati effettuando correlazioni con l'obiettivo di sviluppare una spiegazione su quanto accaduto.

3. Infine si procede con la *presentazione*. Il report finale dovrebbe contenere informazioni riguardo a dati analizzati, metodologia utilizzata, durata delle indagini, interpretazioni delle informazioni ottenute e conclusioni.

In conclusione, si può condivere la considerazione che "i dati assunti al di fuori di tali criteri, sotto il profilo giuridico prestano il fianco al rischio di inutilizzabilità, mentre sotto il profilo del merito legittimano informazioni erronee o inattendibili, in ogni caso inidonee a costituire la base di sentenze capaci di superare il limite minimo 'al di là di ogni ragionevole dubbio'"[36].

1.3 Metodi dell'informatica forense

Il contesto in cui l'informatico forense si trova ad operare presenta diverse peculiarità determinate dalle caratteristiche architetturali dei dispositivi digitali e dal rapido sviluppo della tecnologia, che imprime forte dinamismo all'evoluzione della materia.

L'informatica forense, a differenza di altre scienze forensi, si caratterizza anche per l'ampiezza e la complessità che le indagini ad oggetto informatico possono raggiungere. Si pensi, ad esempio, a scenari, sempre più frequenti, nei quali vi è ampio utilizzo di crittografia, oppure sia necessario acquisire dischi di grandi dimensioni: situazioni queste che richiedono scelte specifiche che analizzeremo nelle sezioni seguenti.

1.3.1 Copia forense: la giurisprudenza

L'acquisizione, la duplicazione e l'analisi dei supporti di memoria di un dispositivo elettronico rappresentano un momento di fondamentale importanza durante un'indagine preliminare di un procedimento penale.

Nonostante non manchino in dottrina opinioni di diverso avviso, l'attuale orientamento maggioritario della Cassazione tende ad escludere che gli accertamenti tecnici ad

[36] A. Gammarota, *Indagini forensi e processo penale...*, p. 201

oggetto informatico rientrino nella categoria degli accertamenti tecnici non ripetibili, i quali offrono una serie di garanzie processuali estremamente importanti[37].

In particolare, la giurisprudenza maggioritaria della Cassazione non considera l'attività di copia un *accertamento tecnico irripetibile ex* art. 360 c.p.p.[38]. Come si legge in una recente sentenza, l'operazione, di per sè, "non comporta alcuna attività di carattere valutativo su base tecnico-scientifica né determina alcuna alterazione dello stato delle cose, tale da recare pregiudizio alla genuinità del contributo conoscitivo nella prospettiva dibattimentale, essendo sempre comunque assicurata la riproducibilità di informazioni identiche a quelle contenute nell'originale"[39].

In verità, la tesi non pare del tutto condivisibile, in quanto l'attività di estrazione di copia di un file può determinare alterazione dello "stato delle cose". Bisogna infatti tenere conto che alcune operazioni, come l'acquisizione della memoria volatile, modificano in realtà la sorgente rendendo la riproducibilità non garantita. Inoltre non si considerano casi limite di indagini durante le quali non vengano applicati gli standard di settore, con conseguenze sul contenuto informativo dei dati estratti, in relazione ai quali "verrebbe meno la loro capacità rappresentativa e con essa tutto il patrimonio informativo che dagli stessi può essere tratto ai fini dell'indagine o della difesa"[40].

1.3.2 Acquisizioni *post mortem* e acquisizioni *live*

L'attività di acquisizione è riconducibile sostanzialmente a due tipologie: quella c.d. *post mortem* e quella *live*.

L'acquisizione *post mortem* consiste nella realizzazione di una copia delle memorie secondarie di un dispositivo dopo che esso è stato spento.

L'operazione può essere svolta principalmente in tre modi, ovvero utilizzando:

[37] Vedi A. Gammarota *Informatica forense ...*, pp. 141-143.

[38] Si vedano in questo senso, sulla base di argomenti diversificati, Cass., sez. I, sent. 26 febbraio 2009 - 18 marzo 2009, n. 11863; Cass., sez. I, sent. 5 marzo 2009 - 2 aprile 2009, n. 14511; Cass., sez. II, sent. 4-16 giugno 2015, n. 24998; Cass., sez. II, sent. 8 luglio 2015, n. 29061.

[39] Cass., sez. pen. VI, 20 dicembre 2018, n. 15838, reperibile in `https://web.archive.org/web/20200831181135/http://www.italgiure.giustizia.it/xway/application/nif/clean/hc.dll?verbo=attach&db=snpen&id=.%2F20190410%2Fsnpen%40s60%40a2019%40n15838%40tS.clean.pdf`.

[40] A. Gammarota, *Informatica forense ...*, p. 158.

- un *write blocker*, ossia un dispositivo fisico che blocca tutti i comandi di scrittura da parte del sistema operativo della workstation su cui è in esecuzione il programma di copia;
- specifiche distribuzioni Linux, avviabili in modalità *live*, che montano i dischi in modalità read only (e.g. Caine, DEFT, Tsurugi, etc.);
- un copiatore forense che, collegando due dischi, esegue direttamente la procedura di copia richiedendo un minimo sforzo da parte dell'operatore.

Una volta che è stata inibita la possibilità di scrivere sui supporti di memorizzazione si procede con l'operazione di copia vera e propria. A seconda degli strumenti a disposizione, e delle necessità, è possibile scegliere tra diverse strategie appropriate.

Le acquisizioni *live* differiscono dall'approccio *post mortem* in quanto si svolgono sul computer in stato di attività e per questo motivo possono essere lasciate tracce su *log* di sistema e nella memoria primaria. Ciò rende necessaria un'attenta documentazione di tutte le operazioni svolte.

L'approccio *live* si rivela necessario quando si devono acquisire informazioni che non vengono consolidate sul disco (e.g. alcuni dati relativi alla navigazione internet o il contenuto della memoria volatile) oppure qualora le memorie di massa siano crittografate e si sia privi delle password di decrittazione.

Copia *bitstream*

Se vi è necessità di acquisire tutti i settori del disco, compresi gli spazi non allocati, occorre realizzare una copia *bitstream*. Questo metodo copia bit per bit la memoria sorgente sul supporto di destinazione e permette di svolgere operazioni di *carving*, cioè l'estrazione dei dati eliminati.

La procedura comporta alcuni svantaggi. Nel caso della realizzazione di un clone è richiesto che il supporto di destinazione sia quantomeno della stessa dimensione dell'originale, cosa non sempre possibile per memorie di massa di grandi dimensioni. Qualora si decida di optare per la realizzazione della copia su un file immagine sarà possibile scegliere rapporti di compressione per occupare meno spazio sul dispositivo di destinazione.

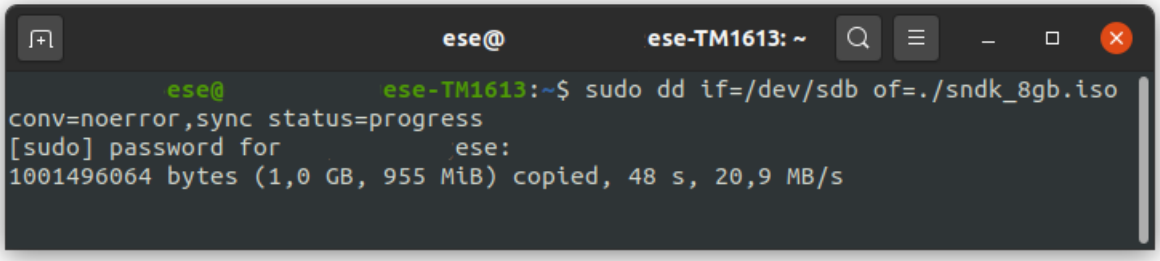

Figura 1.3: Un esempio di copia *bitstream* mediante `dd`

Copia logica

La copia logica si differenzia dalla copia *bitstream* in quanto non viene realizzato un clone della memoria sorgente, ma vengono copiati solamente gli spazi allocati di determinate partizioni ed è di particolare utilità qualora si necessiti solo di alcuni specifici file; può anche essere svolta in *live* tramite applicativi come FTK.

Questa modalità di acquisizione consente di occupare sensibilmente meno spazio rispetto alla copia bit per bit, dato che non viene considerato lo spazio non allocato; è inoltre più veloce, dal momento che permette di salvare solo i dati strettamente necessari scelti dall'operatore.

Di contro l'utilizzo di questa metodologia non consente di acquisire contenuti eliminati dall'utente e gli *slack space*; ciò comporta una perdita di informazioni, potenzialmente anche molto rilevanti, che va considerata attentamente.

Questo ha però risvolti positivi sulla riservatezza qualora sui computer in esame siano presenti informazioni che esulano dalle indagini in atto. Si pensi ad un caso ipotetico in cui un'azienda subisca un accesso abusivo e nella copia forense *bitstream* vengano inclusi anche documenti contenenti informazioni su nuovi prodotti in fase di progettazione, quando sarebbe stato sufficiente acquisire solo alcuni *log* di sistema (con conseguente violazione di uno dei principi dello standard ISO/IEC 27037:2012).

Acquisizione della memoria volatile

La memoria volatile, o memoria primaria, è una componente fondamentale per i calcolatori: in essa sono contenuti tutti i programmi in esecuzione insieme ai loro dati e a quelli necessari al sistema operativo.

Il contenuto della RAM deve essere salvato prima dello spegnimento di un computer in quanto, da questo momento in poi, l'alimentazione agli slot di memoria viene sospesa con la conseguente perdita graduale della carica elettrica e delle informazioni contenute; inoltre il costante mutamento dei dati, ad opera dei programmi in esecuzione, rende spesso urgente la procedura di clonazione.

La copia di questo tipo di supporto rappresenta un'operazione irripetibile in quanto il solo spostamento in RAM del programma per effettuare la procedura può comportare la sovrascrittura di determinate zone di memoria con alterazione del dato originale. Tuttavia il pericolo appare compensato dai vantaggi di tale approccio di acquisizione, che consentirebbe ad esempio di ottenere dati relativi ad applicativi *web*, sempre più diffusi, altrimenti non ricavabili da un'analisi delle memorie di massa, oppure reperire password relative ad aree crittografate delle stesse[41]: il che rende talvolta non solo opportuno, ma addirittura necessario il ricorso a questo tipo di operazione. Uno dei programmi più utilizzati a tal fine è rappresentato da Volatility, che dispone di plugin modulari per il recupero di specifici contenuti.

[41] Per un esempio sulla procedura vedasi `https://web.archive.org/web/20191222072540/https://volatility-labs.blogspot.com/2014/01/truecrypt-master-key-extraction-and.html`.

Capitolo 2

Logiche di correlazione

2.1 La fase di analisi: il problema dei dati

La fase di analisi segue l'acquisizione delle prove digitali ed è quel momento in cui l'informatico forense studia i dati in suo possesso al fine di formulare ipotesi su cosa sia accaduto all'interno di un determinato sistema informatico.

Durante l'analisi il *know-how* di un operatore si pone come base fondamentale del processo; tuttavia anche l'esaminatore più esperto si trova di fronte a un problema che negli anni a venire tenderà ad aggravarsi sempre maggiormente, ossia la mole di dati da analizzare.

Confrontando il mercato delle memorie di massa degli anni '80 con quello odierno si è assistito ad una forte crescita dello spazio a disposizione degli utenti - come visibile in figura 2.1 - a fronte di una altrettanto consistente diminuzione dei prezzi; allo stesso tempo la quantità di dati generata è aumentata, rendendo l'analisi un processo potenzialmente molto lungo. Sottolineano V. Roussev *et al.: "The average amount of data per case, as experienced by FBI's 15 Regional Computer Forensic Laboratories, has grown 6.65 times (from 84 GB to 559GB) in eight years (2003-2011)"*[1].

[1] V. Roussev, C. Quates, R. Martell, *Real-time digital forensics and triage.*, in *Digital Investigation*, Vol. 10, 2013, p. 158.

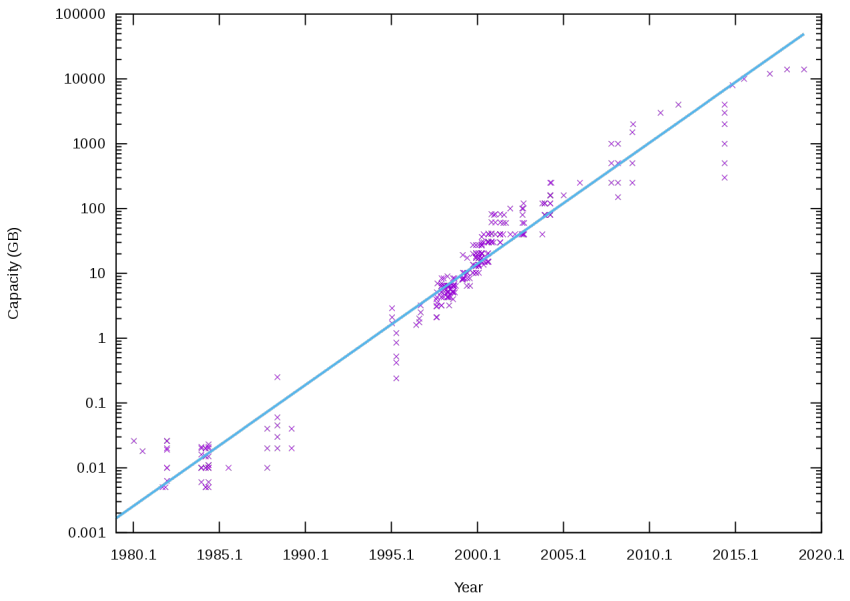

Figura 2.1: L'immagine, tratta da Wikipedia, mette in relazione la capacità delle memorie di massa commercializzate da diversi produttori con l'anno di presentazione.

2.1.1 Il triage

Il tentativo di velocizzazione del processo investigativo è avvenuto attraverso lo sviluppo di nuove metodologie operative: il primo esempio è certamente rappresentato dal *Computer Forensics Field Triage Process Model*[2].

Il CFFTPM si pone come una fase di pre-analisi; il processo infatti viene svolto fuori da un laboratorio forense prima di quanto descritto dagli standard ISO/IEC 27037:2012 e 27042:2015. Tuttavia ciò non impedisce che, in un secondo momento, si possa proseguire con i passi individuati dai due documenti precedentemente citati.

Gli ideatori del CFFTPM prevedono che i dispositivi digitali vengano individuati e se ne stabilisca una priorità per poi analizzarli rivolgendo l'attenzione su alcuni elementi di interesse indicati dal modello. Alcuni esempi sono la cartella home[3] dell'utente, artefatti

[2]Il nome della metodologia coincide con il titolo dell'articolo in cui viene presentata; per approfondimenti si faccia riferimento a M. K. Rogers, J. Goldman, R. Mislan, T. Wedge, S. Debrota, *Computer Forensics Field Triage Process Model*, in *Journal of Digital Forensics, Security and Law*, Vol. 1, 2006, pp. 19-38.

[3]A causa dell'elevato numero di termini in inglese, da qui in avanti non si utilizzerà più il corsivo

relativi ai browser, registri di sistema, etc. Durante le operazioni l'integrità dei dati viene garantita connettendo i device ad una workstation mediante l'utilizzo di un write blocker per poi procedere all'analisi mediante piattaforme di analisi statica.

Gli autori stessi però riconoscono come la durata dell'analisi dipenda fortemente dagli strumenti a disposizione, dalla conoscenza dell'esaminatore e dalla sua esperienza[4]. Viene inoltre sottolineato come alcune ricerche abbiano costi proibitivi in termini di tempo e come sia quindi fondamentale che l'operatore abbia una chiara idea di cosa stia cercando[5]; ciò rende il CFFTPM una non-soluzione. Infatti, come affermano altri autori: *"Digital triage has to be fast if it is to be useful, and requiring the user to have large amounts of expertise to use the tool limits its usefulness"*[6].

Altri articoli, successivi alla formalizzazione del primo modello di triage, effettuano una suddivisione in live triage, svolto su calcolatori attivi, e post-mortem triage, su computer spenti[7]. V. Jusas *et al.* confermano come entrambe le modalità si collochino prima della metodologia forense standard e che ciò possa essere utile a valutare se una fonte di prova meriti di essere analizzata approfonditamente o meno[8].

Tuttavia diversi autori sollevano perplessità sulla possibilità di considerare questo processo come facente parte dell'informatica forense, specialmente nel suo svolgimento live[9] a causa della potenziale invasività delle operazioni.

Una visione alternativa è quella che identifica il triage come un procedimento il cui obiettivo è produrre *intelligence*, e non dati ammissibili a processo, utilizzabile nelle successive operazioni forensi[10].

Il dibattito, di fatto, risulta ancora aperto, evidenziando la novità di questo specifico settore; ad ogni modo risulta evidente come l'accelerazione del processo di indagine sia

per i tecnicismi informatici.

[4]M. K. Rogers, J. Goldman, R. Mislan, T. Wedge, S. Debrota, *Computer Forensics Field...*, p. 28.

[5]Ivi, p. 33.

[6]G. Cantell, D. Dampier, Y. S. Dandass, N. Niu, C. Bogen, *Research toward a Partially-Automated, and Crime Specific Digital Triage Process Model*, in *Computer and Information Science*, Vol. 5, 2012, p. 31.

[7]Si fa riferimento a V. Jusas, D. Birvinskas, E. Gahrmanov, *Methods and Tools of Digital Triage in Forensic Context: Survey and Future Directions*, in *Symmetry*, Vol. 9, 2017, p. 2.

[8]Ivi, p. 1.

[9]È questa la posizione di V. Jusas *et al.*, *Methods and Tools...*, p. 2.

[10]Vedi G. Cantrell *et al.*, *Research toward...*, p. 30.

diventato un requisito prioritario per l'informatica forense.

2.1.2 Stato dell'arte dell'analisi automatizzata

Se l'esecuzione di quelli che potremmo quasi definire modelli agili di indagine trova il proprio rallentamento principale nella fase di analisi dei dati, la soluzione è certamente rappresentata dalla selezione automatizzata dei dati di maggiore rilevanza; in tal modo sarebbe possibile restituire all'operatore un sottoinsieme di informazioni di più rapida interpretabilità.

Piattaforme per l'analisi statica, come Autopsy, permettono di importare copie forensi da più fonti e restituiscono una visualizzazione agevolata, fornendo, ad esempio, la possibilità di effettuare ricerche utilizzando espressioni regolari o mettendo in evidenza file la cui estensione non coincida con la file signature[11].

Gli utenti possono anche creare moduli aggiuntivi, in Java o Python, per processare dati secondo le loro necessità. L'espandibilità e la combinazione di tali moduli aggiuntivi risulta però non immediata a causa dell'architettura sottostante.

Strumenti come Cyber Triage hanno il pregio di automatizzare alcune procedure: ad esempio è possibile lanciare un'analisi su più host e, secondo alcune logiche interne, l'utilizzatore verrà informato di quali file ed eventi meritano una maggiore attenzione. È importante notare che tale soluzione, per il momento, non consente a un operatore di definire ulteriori elaborate regole di correlazione e pertanto risulta limitata a quanto codificato dagli sviluppatori.

In ambito accademico sono stati effettuati sforzi in molteplici direzioni; ad esempio tentando di ricondurre il problema della correlazione dei log in termini algebrici[12] o proponendo framework per l'analisi dei metadati sulla base di tecnologie tipiche del semantic web[13].

[11] La file signature è una sequenza di byte all'inizio di ogni file. Differenze tra la estensione e la firma del file possono identificare la volontà di nascondere determinati contenuti.

[12] Si veda A. R. Arasteh, M. Debbabi, A. Sakha, M. Saleh, *Analyzing multiple logs for forensic evidence*, in *Digital Investigation*, Vol. 4S, 2007, pp. 82-91.

[13] Cfr. H. Mohammed, N. Clarke, F. Li, *An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data*, in *Journal of Digital Forensics, Security and Law*, Vol. 11, 2016, pp. 137-152.

A livello progettuale sono stati realizzati tentativi spazianti dalla creazione di database contenenti log, ridotti a una forma canonica, estratti da svariati applicativi e interrogabili dall'utente[14], allo sviluppo di piattaforme che, partendo da un'analisi della memoria, indagano i collegamenti tra processi, network e file su disco[15].

Tuttavia i progetti precedentemente citati presentano delle limitazioni, infatti:

- non sono pensati per essere eseguiti in parallelo su più host e sfruttare i risultati anche a livello di insieme;
- la possibilità di personalizzazione delle correlazioni, ove presente, è generalmente scarsa o difficile da implementare;
- non sembrerebbe che sia presente una modularità tale che consenta di sviluppare singoli "plugin" riutilizzabili da più logiche di correlazione;
- spesso si limita l'analisi ai file di log, non considerando altri artefatti di rilevanza forense. Ciò restituisce una visione limitata vista la specificità di alcune informazioni che i secondi possono contenere e sulle quali l'investigatore può fare affidamento.

Racoon

Durante l'attività di tirocinio è stato sviluppato un prototipo, scritto in Python, denominato Racoon. L'obiettivo era quello di sopperire ad alcune delle mancanze degli altri progetti in termini di specificità degli artefatti forensi analizzati e di aggregazione dei dati provenienti da più calcolatori.

L'applicativo, ideato per essere eseguito su una workstation appartenente a un esaminatore, riceveva in input un numero variabile di cartelle contenenti file CSV output di parser di terze parti. Una volta indicizzati i file da analizzare, Racoon ricercava in essi elementi comuni per correlare tra loro informazioni provenienti da fonti differenti; ad

[14]È la soluzione sviluppata da K. Chen, A. Clark, O. De Vel, G. Mohay, *Event Correlation for Forensics*, 2003.

[15]Per maggiori informazioni si veda A. Case, A. Cristina, L. Marziale, G. G. Richard, V. Rousseu, *FACE: Automated digital evidence discovery and correlation*, in *Digital Investigation*, Vol. 5S, 2008, pp. 65-75 e anche X. Fu, X. Du, B. Luo, *Data correlation-based analysis methods for automatic memory forensic*, in *Security and Communication Networks*, Vol. 8, 2015, pp. 4213-4226.

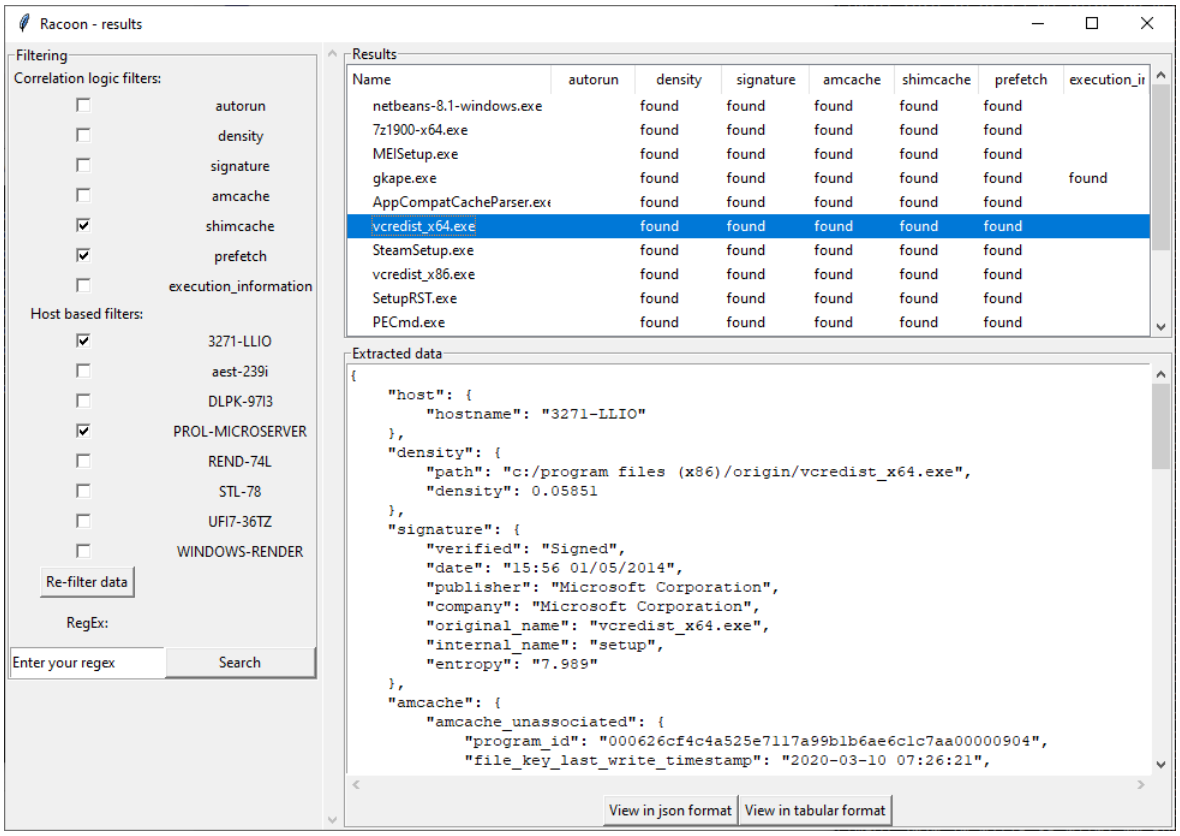

Figura 2.2: Lo screenshot ritrae la schermata di presentazione dei risultati di Racoon. L'operatore sta filtrando gli elementi provenienti da due specifici host e di cui è stata trovata presenza in Shimcache e tra i file di Prefetch.

esempio, dai processi in esecuzione veniva ricavata la posizione sul disco dell'eseguibile e questa veniva sfruttata per ottenere le informazioni relative alla firma di tale programma.

Terminate le associazioni, i risultati venivano restituiti come visibile nella figura 2.2. L'operatore poteva quindi filtrare quanto ottenuto utilizzando espressioni regolari, a seconda dell'host da cui i dati erano stati estratti e sulla base della presenza in uno o più artefatti forensi.

Il progetto risultava però inadatto dal punto di vista dell'espandibilità, dal momento che, per ogni logica di correlazione che si fosse voluta codificare, si sarebbe dovuta progettare una nuova classe. La flessibilità garantita era sostanzialmente assoluta, ma al prezzo di dover scrivere intere nuove sezioni di codice e di conoscere Python.

Considerazioni sull'analisi live

Un aspetto che merita particolare attenzione è quello riguardante l'esecuzione di applicativi per l'analisi dei dati. Gli autori dei progetti precedentemente citati non sembrano soffermarsi sugli ambienti nei quali questi programmi sono pensati per essere eseguiti; tuttavia questo rappresenta un aspetto di non marginale importanza dato che, come dimostrato in precedenza, qualsiasi azione svolta su un calcolatore produce un mutamento nelle informazioni in esso contenute.

Ad una prima analisi si potrebbe pensare che l'esecuzione di programmi su un computer in analisi non rappresenti un eccessivo problema in quanto per le acquisizioni live ciò è permesso; è però importante evidenziare come il caso appena citato si ponga su un differente livello operativo, nel quale la finalità ultima è preservare alcuni dati, come quelli relativi alla memoria volatile, non ottenibili diversamente.

L'utilizzo di applicativi ai fini di analisi è normata dallo standard ISO/IEC 27042:2015 di cui si riporteranno ora i casi trattati.

Qualora non fosse possibile effettuare una copia bitstream, ad esempio a causa dell'impossibilità di spegnere un determinato sistema, l'esecuzione di applicativi ai fini di analisi direttamente sul calcolatore stesso sarebbe permessa, saltando, di fatto, tutte le fasi descritte dallo standard ISO/IEC 27037:2012. In questo caso risulta comunque necessario prendere precauzioni per minimizzare il rischio di danneggiamento di potenziali prove, documentando attentamente il processo, ad esempio effettuando riprese, ed essendo in grado di giustificare qualsiasi alterqaione dei dati che possa essere stata prodotta da quanto svolto[16].

Qualora fosse invece possibile effettuare un clone del sistema, una volta realizzato, esso andrebbe eseguito in un ambiente virtuale o, meglio ancora, sull'hardware stesso[17]; in tal caso non vi sarebbero problematiche nell'utilizzo di applicativi che altererebbero lo stato del sistema in quanto si starebbe operando su una copia e non sui dati originali.

Alla luce di quanto emerge dallo standard risulterebbe quindi legittima l'esecuzione di applicativi su un clone del calcolatore o, in specifici casi, direttamente su esso stesso.

[16]Per approfondimenti si rimanda a *Guidelines for the analysis and interpretation of digital evidence (ISO/IEC 27042:2015)*, p. 9.

[17]Ibidem.

2.1.3 Velociraptor

Stabiliti i limiti procedurali ai quali è necessario sottostare, passiamo ora alla disamina di un possibile applicativo a supporto della fase di analisi.

Velociraptor è un progetto open source che consiste in una "piattaforma per il monitoraggio degli endpoint, Digital Forensics e Cyber Response"[18] tramite cui è possibile lanciare specifiche interrogazioni su uno o più calcolatori. L'applicativo supporta i tre principali sistemi operativi sul mercato (Windows, MacOS e GNU/Linux), rendendolo adatto alla maggior parte dei contesti di indagine.

Gli artefatti

L'attività di analisi è realizzata grazie a specifici file chiamati artefatti[19], che sono scritti secondo la sintassi YAML e possono essere raggruppati in due tipologie: semplici e composti[20]. I primi si occupano di singoli compiti, come ad esempio individuare gli utenti di un sistema o tutte le porte aperte su un calcolatore, mentre i secondi sono formati dalla combinazione dei precedenti. Questo approccio modulare consente quindi di sfruttare la libreria di artefatti con cui Velociraptor viene fornito come base per la creazione di complesse logiche di correlazione.

L'esecuzione di un artefatto produce una o più tabelle che sono il risultato delle query specificate nel file YAML. Una volta ritornato il prodotto dell'esecuzione, è anche possibile creare un report sulla base di regole codificate dal creatore dell'artefatto.

Le funzionalità di reporting consentono di aggregare i risultati provenienti da più dispositivi attraverso la scrittura di specifiche interrogazioni che prendano in considerazione i dati nel loro complesso.

È anche possibile creare report grafici mediante l'utilizzo di funzionalità integrate in Velociraptor; ciò può essere utile qualora si desideri una rappresentazione visuale di quanto ottenuto.

[18] Per maggiori informazioni: `http://web.archive.org/web/20201001174326/https://www.velocidex.com/about/`.

[19] È importante notare che gli artefatti di Velociraptor non coincidono con i c.d. artefatti di rilevanza forense: questi ultimi infatti rappresentano le fonti da cui ottenere dati utili per l'indagine.

[20] La divisione appena effettuata non si riscontra nella documentazione ufficiale ed è qui introdotta solamente al fine di chiarire alcuni aspetti.

Velociraptor Query Language

Il *Velociraptor Query Language* è il linguaggio di interrogazione utilizzato da Velociraptor per ottenere dati dagli host. La sintassi di una richiesta è sostanzialmente identica a quella di SQL, consentendo a un investigatore di scrivere potenti ed espressive interrogazioni con poco sforzo e una bassa curva di apprendimento iniziale.

`SELECT` *Attributi*
`FROM` *Artefatto*
`[WHERE` *Condizione*
`GROUP BY` *Attributo*
`ORDER BY` *Attributo*`]`

Le righe delle tabelle ritornate non contengono necessariamente valori semplici, come stringhe o numeri, ma possono avere al loro interno anche liste, dizionari o oggetti JSON.

Un'ulteriore differenza sostanziale di VQL rispetto ad altri linguaggi è che i soggetti della clausola `FROM` sono altri artefatti che possono ricevere in input parametri i quali verranno utilizzati per filtrare i risultati della tabella che verrà restituita. Inoltre non sono supportate operazioni di join tra le tabelle; per ottenerne un surrogato è necessario procedere nel seguente modo:

`SELECT * FROM foreach(row =` {`SELECT` *Colonna1, Colonna2* `FROM` *Artefatto1()*}`,`
`query =` {`SELECT` *Colonna1, Colonna2, Colonna3* `FROM` *Artefatto2()* `WHERE` *Colonna3 = Colonna1*}`)`

Il plugin `foreach` riceve in input una tabella e, per ogni riga, viene eseguito quanto specificato nel campo `query`. Nel caso dell'esempio vengono selezionati gli elementi della *Colonna1*, *Colonna2* (appartenenti alla prima tabella) e *Colonna3* (appartenente alla seconda) qualora il valore della terza colonna fosse uguale a quello della prima.

Modalità di deploy

Velociraptor permette di essere eseguito principalmente in due modalità: secondo un'architettura client-server e in triage mode.

Nel primo caso è presente un server, che può essere rappresentato dal computer dell'esaminatore, dal quale vengono create specifiche sessioni chiamate hunt.

Una caccia consiste nell'esecuzione di una serie di artefatti scelti tra quelli a disposizione; al momento della selezione è possibile fornire alcuni parametri, dipendenti dall'artefatto, sulla base dei quali saranno filtrati i risultati.

Configurata e avviata la sessione, tutti i client in ascolto ricevono i comandi da eseguire e restituiscono i risultati al server. L'operatore può quindi scaricare un archivio in formato `.zip` contenente l'esito del processo.

Gli endpoint possono lanciare Velociraptor utilizzando un eseguibile portabile[21]. Nel caso di Windows è anche possibile installare la piattaforma come servizio o avviarla tramite un network share ed una group policy[22]. L'ultima opzione citata può risultare particolarmente utile nel caso di indagini "at scale" durante le quali si necessiti di ottenere dati da un gran numero di calcolatori e dove avviare manualmente ogni singolo client rappresenterebbe un'operazione tediosa.

La modalità triage consiste nell'esecuzione di un determinato artefatto senza che venga creata l'infrastruttura client-server discussa precedentemente; in tal caso l'operatore si limita ad avviare il programma tramite riga di comando, a specificare quale artefatto eseguire e in che destinazione desidera venga salvato il risultato.

2.2 Progettazione e sviluppo

L'architettura e la flessibilità di Velociraptor lo rendono un eccellente candidato come base per lo sviluppo di logiche di correlazione per artefatti forensi.

Nella presente sezione si procederà a illustrare il lavoro di progettazione e sviluppo degli artefatti di Velociraptor contenenti la codifica di alcune correlazioni che un operatore

[21] Che quindi non richiede l'installazione sul sistema.

[22] Qualora l'eseguibile di Velociraptor fosse disponibile su un percorso condiviso, sarebbe possibile far eseguire automaticamente tale applicativo a tutti i computer appartenenti a un determinato dominio. Per maggiori informazioni sulla procedura: `https://web.archive.org/web/20201002111537/https://www.velocidex.com/docs/getting-started/deploying_clients/`.

si troverebbe altrimenti a dover svolgere manualmente durante la fase di analisi[23].

Sulla base del sottoinsieme di dati estratti si potrà poi articolare la fase di interpretazione, secondo quanto descritto nello standard ISO/IEC 27042:2015.

2.2.1 BasicProgramInspection

`BasicProgramInspection` è il nome della logica di correlazione sviluppata durante l'attività di tirocinio. Essa verrà trasposta come artefatto di Velociraptor.

L'obiettivo principale delle associazioni effettuate è quello di individuare file e programmi sospetti al fine di ottenere informazioni aggiuntive sulle esecuzioni precedenti.

Autoruns

La prima necessità di un malware, a seguito della sua installazione, è quella di poter essere eseguito in automatico all'avvio del calcolatore. Il raggiungimento della persistenza sarebbe infatti inutile se un applicativo malevolo dovesse essere avviato manualmente dall'utente al termine di ogni boot del sistema.

La tecnica più utilizzata è quella di registrare il programma tra quelli che richiedono l'avvio automatico, i cosiddetti autorunnabili.

Autoruns è un artefatto creato per sfruttare l'omonimo applicativo, distribuito da Microsoft, che ritorna informazioni relative ai programmi avviati in automatico da un calcolatore.

Ai fini dell'analisi verranno quindi selezionati tutti i programmi che risultano non firmati e che utilizzano la funzionalità precedentemente descritta; per ottenerne la lista viene utilizzato l'artefatto `Autoruns` nativamente incluso in Velociraptor.

```
- name: Autoruns
  queries:
     - LET autorunsData <= SELECT EntryLocation, Entry, Enabled,
       Category, Profile, Description, Signer, Company, ImagePath,
       Version, LaunchString, MD5, SHA-1, SHA-256, FROM
```

[23] Verrà preso in considerazione solamente il sistema operativo Windows; la scelta è stata dettata dal market share che quest'ultimo ha in ambito Desktop.

```
      Artifact.Windows.Sysinternals.Autoruns(AutorunArgs=
      '-nobanner -accepteula -t -s -h -a * -c *') WHERE Signer=""
      and ImagePath
```

Processi in esecuzione

Il secondo aspetto tenuto in considerazione è quello dei processi attualmente in esecuzione; tutti i programmi verranno aggiunti alla lista di quelli da scansionare.

Anche in questo caso si fa affidamento su un artefatto già disponibile in Velociraptor.

```
- name: PsList
  queries:
     - LET programsInExecution <= SELECT * FROM Artifact.Windows.
       System.Pslist() WHERE Authenticode.Filename
```

Densityscout

L'individuazione di applicativi malevoli può risultare semplice qualora questi siano già conosciuti e la loro signature[24] sia stata salvata all'interno di database sfruttati dai software antivirus. Al contrario, se il file è sconosciuto e ha subito una compressione unita a tecniche di offuscamento tramite crittografia[25], la loro identificazione potrebbe risultare complessa.

Densityscout è un applicativo che nasce per individuare malware potenzialmente sconosciuti; per ottenere tale risultato il programma sfrutta la frequenza con la quale compaiono sequenze di byte all'interno di un file[26] al fine di calcolare un indice, definito densità.

Velociraptor non viene fornito con alcun artefatto che consenta di sfruttare Densityscout e per questo motivo ne è stato creato uno.

[24]La signature di un malware, similmente a quella di un normale file, è una sequenza di byte tipici di una famiglia di software malevoli.

[25]Quanto descritto è tipico dei malware moderni e viene definito packing.

[26]File compressi e crittografati avranno una distribuzione estremamente varia dei byte; per maggiori informazioni è possibile consultare: `https://web.archive.org/web/20200728151224/https://www.cert.at/en/downloads/software/software-bytehist`.

```
name: Custom.Windows.Forensics.DensityScout
description: |
    Uses DensityScout to scan files.

tools:
  - name: DensityScout
    url: https://www.cert.at/media/files/downloads/software/densityscout
    /files/densityscout_build_45_windows.zip

precondition: SELECT OS From info() where OS = 'windows'

required_permissions:
  - EXECVE

parameters:
  - name: DensityScoutArgs
    description: |
      A list of args for DensityScout.
    default: " -pe -p 0.1 -l 0.1 -d -r "
  - name: ScanPath
    description: |
      The path to scan.
    default: ``C:\\Windows\\System32"

sources:
  - query: |

      // Get the path of DensityScout and a tempdir
     LET zip <= SELECT FullPath, tempdir() AS TempDir
         FROM Artifact.Generic.Utils.FetchBinary(ToolName=
```

```
        "DensityScout")

    // "Unzips" the executable.
   LET exec <= SELECT copy(filename=url(path=zip[0].FullPath,
       fragment="win64/densityscout.exe").String,
       dest=tempfile(extension='.exe'), accessor='zip') AS Executable
       FROM scope()

    // Executes the program and writes the output in the temp folder
   LET result <= SELECT * FROM Artifact.Windows.System.PowerShell(
       Command=exec[0].Executable+DensityScoutArgs+ScanPath+" -o "
       + zip[0].TempDir+"\\density.txt")

    // Creates the final table with density and file fullpath
   SELECT regex_replace(source=Path, re="[\r\n]", replace="") as Path,
   Density FROM split_records(filenames= [zip[0].TempDir+
   "\\density.txt"],regex='\\|', columns=['Density', 'Path'],
   first_row_is_headers=false)
```

Quando Velociraptor esegue il file YAML scarica l'archivio compresso contenente il programma, specificato nella sezione `tools`; successivamente questo viene estratto ed eseguito con alcuni parametri specificati nelle variabili `DensityScoutArgs`, riguardante i settaggi che Densityscout deve utilizzare per la scansione, e `ScanPath`, relativa al percorso che si vuole esplorare ricorsivamente.

L'output dell'esecuzione viene scritto in un file che, una volta letto, sarà restituito come tabella finale.

L'artefatto creato può essere utilizzato singolarmente o all'interno di logiche più complesse, come nel caso della `BasicProgramInspection`.

```
- name: DensityScout
  queries:
      - LET densityData <= SELECT * FROM chain(a={SELECT * FROM
```

```
        Artifact.Custom.Windows.Forensics.DensityScout(ScanPath=
        DensityScanPath)}, b={SELECT * FROM foreach(row={SELECT Path
        FROM paths}, query={SELECT * FROM Artifact.Custom.Windows.
        Forensics.DensityScout(ScanPath=Path)})})
```

Nel codice riportato vengono scansionati il percorso scelto dall'utente, che può coincidere anche con l'intera memoria di massa, e gli eseguibili individuati nei passi precedenti da `Autoruns` e `Pslist`.

Sigcheck

Individuati i possibili file di interesse, si vogliono ottenere dettagli relativi alla firma digitale con cui sono stati distribuiti e al loro riconoscimento da parte di piattaforme per l'analisi di malware, come VirusTotal.

Tali informazioni possono essere utili per filtrare programmi che hanno subito una compressione "legittima" - come nel caso di comuni installer che probabilmente risulteranno regolarmente firmati e non malevoli - da applicativi non firmati per i quali sono stati individuati comportamenti sospetti ai quali potrebbe essere opportuno prestare maggiore attenzione.

Per ottenere tali dati verrà sfruttato Sigcheck. La creazione dell'artefatto non sarà riportata in quanto estremamente simile al precedente, al contrario si presterà attenzione solo al suo richiamo all'interno di `BasicProgramInspection`.

```
- name: Sigcheck
  queries:
      - LET sigcheckData <= SELECT * FROM foreach(row={SELECT Path FROM
        paths}, query={SELECT * FROM Artifact.Custom.Windows.Forensics.
        Sigcheck(SigcheckArgs=' -a -c -h -vt -v -accepteula -nobanner ',
        ScanPath="'"+Path+"'") WHERE VT_detection=~"^[1-9].*" and
        Verified != "Signed"})
```

Per ogni percorso dei file individuati precedentemente, viene richiamata l'esecuzione di Sigcheck. La tabella restituita, contenente i dati relativi a programmi non firmati dal

produttore e segnalati da almeno uno degli engine sfruttati da VirusTotal, viene salvata nella variabile `sigcheckData`.

Amcache e Shimcache

Filtrati i programmi riconosciuti come dannosi, se ne estraggono le informazioni di esecuzione presenti sul sistema.

L'Amcache è uno dei registri di Windows e memorizza due dati di importanza primaria relative ai programmi eseguiti su un determinato calcolatore: il percorso di esecuzione, che sarà usato per individuare i programmi di interesse, e l'ultima modifica della voce sul registro, coincidente con la prima esecuzione. Oltre ai campi elencati ne sono disponibili altri che immagazzinano, ad esempio, nome del programma, versione dell'eseguibile, grandezza del file, etc.

La Shimcache contiene un elenco di programmi eseguiti utilizzando gli strumenti per la compatibilità di Windows. Questo comportamento è spesso sfruttato dai malware in diversi modi; un esempio può essere il mantenimento della persistenza sul sistema[27].

La lista è limitata agli ultimi 1024 programmi eseguiti ed è ordinata come uno stack in cui gli applicativi usati recentemente si trovano nelle posizioni più alte.

Entrambi gli artefatti sono già interrogabili nativamente tramite Velociraptor e pertanto se ne filtreranno i risultati sulla base dei percorsi individuati da Densityscout.

```
- name: Amcache
  queries:
     - SELECT * FROM Artifact.Windows.System.Amcache() WHERE
       lowcase(string=Binary) in lowcase(string=sigcheckData.Path)

- name: ShimCache
  queries:
```

[27] Per un'approfondita analisi sull'utilizzo dello shim engine da parte di programmi malevoli si rimanda a `https://web.archive.org/web/20200716012904/https://www.blackhat.com/docs/eu-15/materials/eu-15-Pierce-Defending-Against-Malicious-Application-Compatibility-Shims-wp.pdf`.

```
    - SELECT * FROM Artifact.Windows.Registry.AppCompatCache() WHERE
      lowcase(string=Name) in lowcase(string=sigcheckData.Path)
```

File di Prefetch

I file di Prefetch contengono dati utilizzati per velocizzare l'apertura dei programmi. Similmente a quanto accade per la Amcache, le informazioni di maggiore rilevanza che si ottengono da questo artefatto sono quelle relative alle esecuzioni precedenti. Tuttavia i file di Prefetch non solo memorizzano la data e l'ora di creazione della voce nel registro di sistema, coincidente al primo avvio di un determinato applicativo, ma anche quelle delle ultime sette esecuzioni del programma e il numero totale delle aperture del programma. Tali dati possono essere particolarmente utili qualora si stesse cercando di ricostruire una timeline di eventi accaduti sul sistema.

In aggiunta ai dati precedentemente elencati, i file di Prefetch contengono anche il nome dell'eseguibile, l'hash del percorso dal quale esso è stato richiamato e i file utilizzati per l'avvio del programma.

Per filtrare le informazioni disponibili nei file di Prefetch viene sfruttato il nome dell'applicativo, ottenuto tramite l'analisi della Amcache.

```
- name: Prefetch
  queries:
    - SELECT * FROM Artifact.Windows.Forensics.Prefetch() WHERE
      lowcase(string=Executable) in lowcase(string=amcacheData.Name)
```

Netstats

L'associazione viene effettuata mediante il Pid del processo legato ad una determinata connessione e quello riportato da `Pslist`, i cui risultati sono stati filtrati per ottenere solamente i processi facenti riferimento a un eseguibile individuato come sospetto.

```
- name: NetstatEnriched
  queries:
```

```
- SELECT * FROM Artifact.Windows.Network.NetstatEnriched() WHERE
  Pid in programsInExecution.Pid
```

Come ultimo aspetto analizzato vengono ricercate connessioni in corso da parte dei processi potenzialmente malevoli; queste sono ottenute mediante l'utilizzo di `Netstat Enriched`.

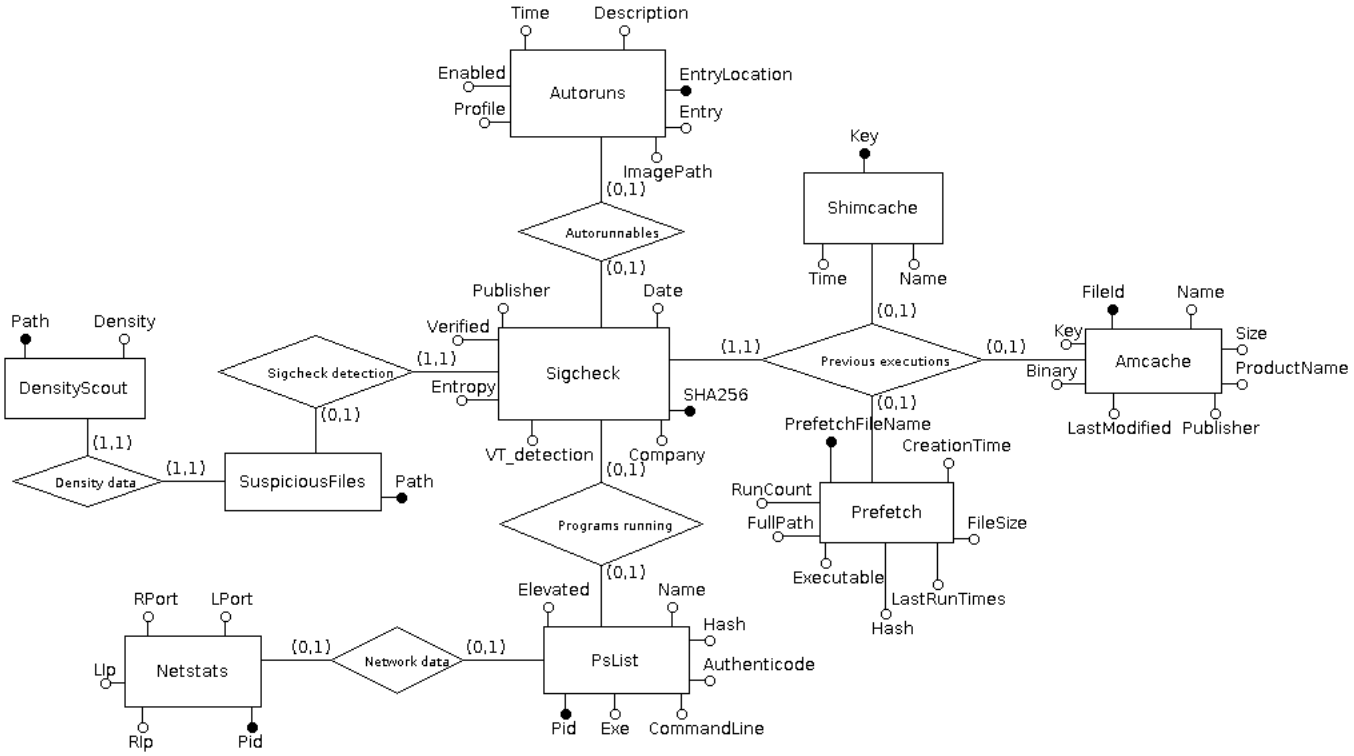

Figura 2.3: BasicProgramInspection

Il risultato finale delle correlazioni effettuate è visibile nel diagramma entità-relazione in figura 2.3.

2.2.2 ImprovedLateralMovement

La seconda logica di correlazione sviluppata riguarda il c.d. movimento laterale. Questo fenomeno consiste nell'ingresso di un soggetto non autorizzato all'interno

dell'infrastruttura di rete aziendale e nel suo movimento all'interno di questa.

Le fasi di questa attività sono tipicamente tre:

- ottenimento di informazioni sul network aziendale;
- infiltrazione nella rete, ad esempio mediante credenziali rubate tramite phising, e arricchimento delle informazioni in proprio possesso sul network aziendale;
- ottenimento di ulteriori credenziali presenti sui dispositivi e riutilizzo delle stesse per muoversi su altri calcolatori nella rete.

L'individuazione del movimento laterale può risultare particolarmente difficoltosa in quanto gli attaccanti, al fine di fare passare inosservata la loro presenza sulla rete, possono utilizzare normali programmi per l'amministrazione di sistema. In aggiunta a questo, l'attività stessa di movimento si svolge tra più calcolatori e dunque la mole di dati da selezionare e analizzare può risultare a dir poco vasta.

Attività di autenticazione

I primi dati che verranno analizzati sono quelli relativi all'attività di autenticazione su un calcolatore. Windows offre un logging preciso per quanto riguarda questi eventi; sono quindi stati realizzati alcuni artefatti per ottenere dai log di sistema i login falliti, quelli avvenuti con successo e, di questi, quelli che riguardano credenziali con speciali privilegi, come ad esempio l'account di amministrazione del sistema.

In tal modo l'esaminatore potrà chiaramente individuare possibili tentativi di brute force avvenuti sulla rete o login inusuali tra calcolatori.

Al fine di ottenere informazioni addizionali è stato modificato un artefatto, già incluso in Velociraptor, riguardante le autenticazioni per le quali sono state fornite credenziali esplicite. Questo dato è utile per individuare account che hanno eseguito azioni assumendo l'identità di altri utenti.

```
- name: FailedLogin
  queries:
      - LET failedLoginData <= SELECT * FROM Artifact.Custom.Windows.
        Events.FailedLogin() WHERE (EventData.LogonType = 3 or
```

```
        EventData.LogonType = 8 or EventData.LogonType = 10) and
        (EventData.SubjectUserName =~ UserFilter or EventData.
        TargetUserName =~ UserFilter or EventData.IpAddress =~
        IpFilter)
```

Nel listato precedente è stata riportata l'estrapolazione dai log di sistema degli eventi riguardanti i tentativi di login falliti[28]: questi, come anche quelli avvenuti con successo, sono selezionati sulla base di specifici tipi (3, 8 e 10) che rappresentano le autenticazioni avvenute attraverso la rete.

L'operatore può filtrare ulteriormente i risultati ottenuti mediante l'inserimento di un username e/o un indirizzo ip che veranno utilizzati nella clausola `where` per selezionare solamente le voci che riguardano quello specifico indirizzo di connessione o utente.

Evidenze di esecuzione

La raccolta di evidenze di esecuzione rappresenta un'altro importante passo per l'individuazione del movimento laterale.

È stata quindi stilata una lista di programmi tipicamente utilizzati da un ipotetico attaccante per svolgere operazioni di ottenimento delle password, di informazioni sugli account e sulla rete.

L'utilizzatore può liberamente aggiungere ulteriori nomi di eseguibili alla lista - ad esempio sulla base di quanto emerso dall'esecuzione di altre logiche, come la `BasicProgram Inspection` - per l'individuazione di programmi malevoli, o rimuoverne, se ritiene che l'utilizzo di un determinato programma rientri nella normale attività dell'azienda.

```
- name: ExeFilter
  description: "Filter evidence of execution using executable names"
  type: csv
  default: |
      ProgramList, Notes
```

[28] A causa della similitudine con il listato precedente, per economia di estensione del presente lavoro non vengono riportate le query per ottenere i login avvenuti con successo, quelli ad account con speciali privilegi e quelli mediante l'utilizzo di credenziali esplicite.

```
      wmic, Windows Management Instrumentation
      wmiprvse, Windows Management Instrumentation
      wmiadap, Windows Management Instrumentation
      mofcomp, Windows Management Instrumentation
      mstsc, Remote Desktop Protocol
      rdpclip, Remote Desktop Protocol
      psexec, PsExec
      psexesvc, PsExec service
      wsmprovhost, Powershell remote session
      net, Network tools
      ipconfig, Network tools
      whoami, User and groups information gatering
      mimikatz, Password obtaining tool

- name: Amcache
  queries:
    - LET exeLookupList <= SELECT * FROM parse_csv(filename=ExeFilter,
      accessor='data')
    # evidence of execution
    - SELECT * FROM foreach(row={SELECT * FROM exeLookupList},
      query={SELECT * FROM FROM Artifact.Windows.System
      .Amcache() WHERE lowcase(string=Name) =~ ProgramList})
- name: PowershellDump
  queries:
    # powershell usage
    - SELECT * FROM foreach(row={SELECT * FROM exeLookupList},
      query={SELECT * FROM Artifact.Custom.Windows.Events.
      PWSDump() WHERE EventData.Data =~ ProgramList})
```

Come per la precedente logica di correlazione sviluppata, anche nella realizzazione di questa sono state cercate evidenze di esecuzione all'interno dell'Amcache, Shimcache

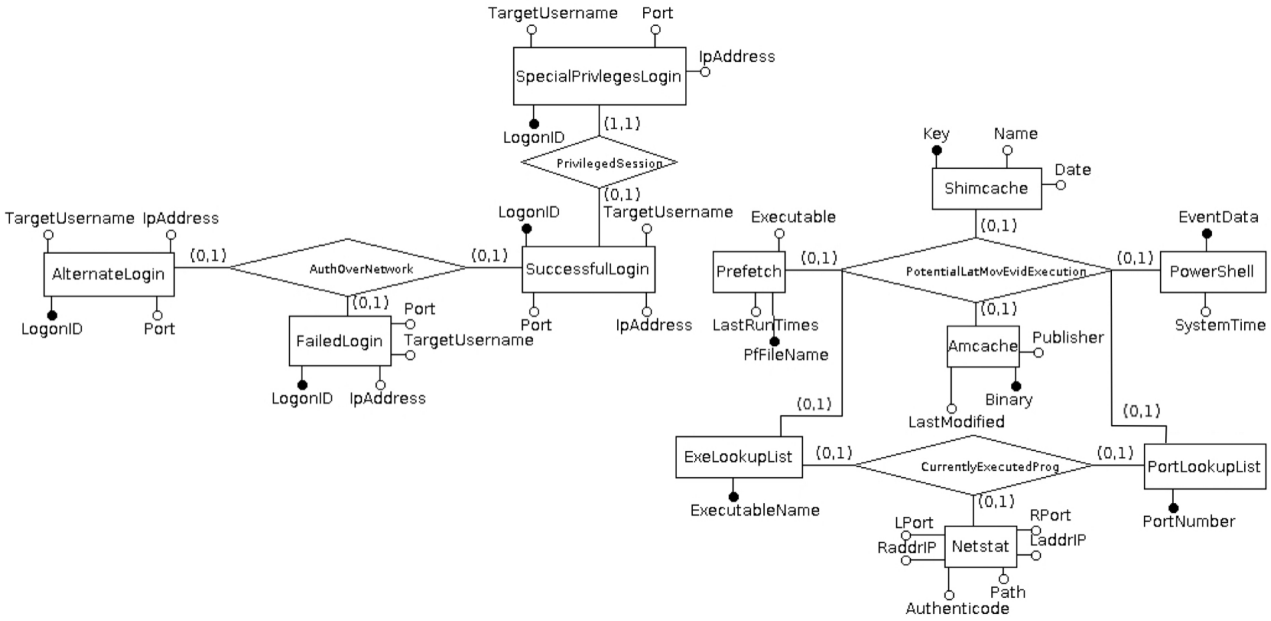

Figura 2.4: ImprovedLateralMovement

e nei file di Prefetch[29]; tuttavia questa volta è stato aggiunto un ulteriore artefatto per l'ottenimento dei comandi lanciati via PowerShell, uno strumento sempre più utilizzato da amministratori di sistema e attaccanti a causa della sua flessibilità. I comandi vengono poi analizzati alla ricerca di termini che rientrino tra i nomi degli eseguibili nella lista.

In ultima istanza viene sfruttato l'artefatto `EnrichedNetstats`, già incluso in Velociraptor, per selezionare dai programmi attualmente in esecuzione quelli riguardanti determinati eseguibili, che utilizzano particolari porte o contattano specifici ip indicati dall'utente. Quanto finora descritto è visibile nel diagramma entità-relazione in figura 2.4.

[29]Il listato non include la totalità delle query in quanto estremamente simili a quella relativa all'Amcache.

Capitolo 3

Test delle logiche di correlazione

3.1 Setup dell'ambiente

Per testare l'efficacia delle logiche di correlazione prodotte sono stati creati due ambienti di test distinti: il primo consiste in una installazione di Windows 10 Pro su un singolo calcolatore, il secondo è rappresentato dall'installazione di ESXi 7.0 come hypervisor per quattro macchine virtuali contenenti una istanza di Windows Server 2019 e tre di Windows 10 Pro.

Il secondo ambiente, quello virtualizzato, è stato configurato per utilizzare l'infrastruttura Active Directory[1] di Microsoft al fine di simulare quanto più fedelmente un contesto aziendale.

Sono quindi stati creati un utente amministratore e tre utenti con privilegi inferiori, che possono autenticarsi presso qualsiasi computer appartenente al dominio `tesi.domain`.

A livello di connettività le macchine virtuali appartengono tutte alla stessa subnet e dispongono del protocollo SMB abilitato e una cartella di rete condivisa a cui sono autorizzati ad accedere mediante le credenziali utente.

Al fine di interagire con il dominio è stato settato come server DNS l'indirizzo IP 192.168.1.25 a cui è reperibile la macchina virtuale di Windows 10 Server 2019.

[1] Active Directory è una collezione di servizi gestiti da un domain controller, ossia un computer che esegue una istanza di Windows 10 Server. Alcuni esempi di servizi erogati possono essere la condivisione di cartelle sulla rete, la gestione degli account utente e dell'esecuzione di determinati software, etc.

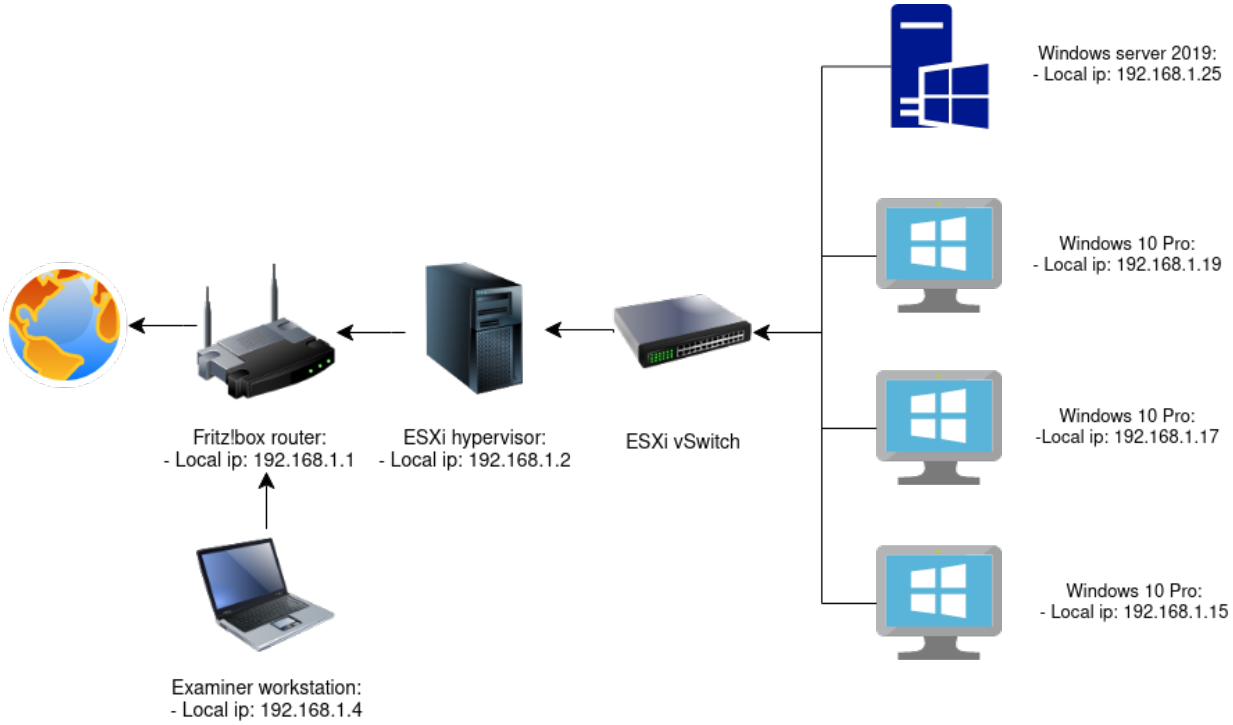

Figura 3.1: Rappresentazione grafica della rete.

L'interazione con il resto della Local Area Network avviene tramite uno switch virtuale, gestito da ESXi, la cui porta di uplink è associata alla scheda di rete fisica del calcolatore il cui indirizzo è 192.168.1.2.

Il server di Velociraptor, da cui saranno avviate tutte le analisi, verrà eseguito su un computer portatile con Ubuntu 20.04 e sarà raggiungibile all'indirizzo IP 192.168.1.4.

Una rappresentazione grafica della rete è visibile in figura 3.1.

3.2 Test BasicProgramInspection

Per testare la logica di correlazione `BasicProgramInspection` sono stati scaricati dal portale `any.run` due network trojan. Il sito offre un report riassuntivo delle azioni svolte dai programmi malevoli comprendente le connessioni effettuate, i file modificati, gli eseguibili installati, le voci di registro manipolate, etc. Tutto ciò rende la fase di test verificabile, in quanto è possibile confrontare i risultati ottenuti rispetto a una serie di comportamenti attesi.

3.2.1 EcV01.04.R.exe

Il primo malware testato è rappresentato da EcV01.04.R.exe. Una volta installato l'applicativo sono stati immediatamente osservati diversi nuovi processi eseguiti sul calcolatore[2]. È stato quindi eseguito un riavvio per simulare l'utilizzo del computer e generare dati di esecuzione nei file di Prefetch, Amcache e Shimcache.

In seguito è stata eseguita una copia bitstream mediante l'utilizzo di `dd` ed è stato verificato che gli hash dei due supporti di memorizzazione, calcolati mediante la funzione sha256, coincidessero[3].

[2]Il report completo è disponibile per la consultazione presso: `https://web.archive.org/web/20201026152744/https://any.run/report/e596d6af81ecbb9fb5903c85ecacade2aa806482fcb6700699e69e676d342b0c/4a5499e5-f66a-4462-aa08-eb9ccf8bcbca`.

[3]L'uguaglianza dei due hash garantisce che la procedura di clonazione abbia prodotto due supporti di boot identici.

Terminata l'acquisizione della memoria di massa è stato disconnesso il disco "origine" ed è stato effettuato il boot del clone appena realizzato[4].

Una volta lanciata la hunt dal server di Velociraptor, dopo pochi minuti sono stati prodotti i risultati visibili nelle tabelle a seguire.

Sigcheck e Densityscout				
Path	VTDetection	Signature	Entropy	Density
`c:\programfiles(x86)\signtool\SignTool.exe`	1—65	Unsigned	6.22	0.95064
`c:\programfiles(x86)\xyrz\xyrzsvc.exe`	39—70	Unsigend	6.023	1.02940
`c:\windows\syswow64\pluginmanager\MPlugin.exe`	44—71	Unsigned	6.05	0.99563
`c:\windows\syswow64\pluginmanager\Plugin.exe`	49—70	Unsigned	6.09	0.99806
`c:\windows\syswow64\pluginmanager\XSDMPlugin.exe`	39—69	Unsigned	6.052	0.99584
`c:\windows\syswow64\pluginmanager\XSDPlugin.exe`	45—70	Unsigned	6.09	0.99849

Nella tabella superiore sono stati inglobati i risultati di Sigcheck e Densityscout. Il campo `Entropy`[5] e `Density` assumono valori rispettivamente poco superiori al 6[6] e poco inferiori all'1[7].

I valori citati, uniti al responso di VirusTotal e alla mancanza di una firma da parte del produttore degli eseguibili, dovrebbero generare sospetti nell'esaminatore ed infatti, come confermato dal report per questo malware, consultabile alla nota numero 2, gli eseguibili individuati sono componenti installate sul sistema dal malware in analisi.

[4]Considerando che il lavoro di analisi live viene effettuato direttamente sul clone realizzato e che ogni azione svolta sul sistema causa inevitabilmente mutamenti al contenuto del supporto, sarebbe stato opportuno realizzare una seconda copia; tuttavia, per velocizzare in questa sede le operazioni, tale passaggio non è stato svolto. In ogni caso sarebbe stato possibile effettuare un ulteriore clone a partire dal disco di origine scollegato dal calcolatore poco prima di eseguire il boot.

[5]Il concetto di entropia è ripreso dalla teoria dell'informazione e, similmente alla densità, misura il grado di casualità del contenuto di un messaggio.

[6]La scala di misura dell'entropia è compresa tra 0, che rappresenta il massimo della prevedibilità, e 8, che rappresenta la massima randomicità.

[7]Si tenga presente che la densità ha una scala inversa rispetto all'entropia: valori minori rappresentano file che hanno una distribuzione dei valori dei byte molto ampia e ciò può indicare l'utilizzo di tecniche di packing.

Amcache		
Binary	LastModified	Name
c:\programfiles(x86)\signtool\signtool.exe	2020-10-20T14:06:55Z	SignTool.exe
c:\programfiles(x86)\xyrz\xyrzsvc.exe	2020-10-20T14:06:55Z	xyrzsvc.exe
c:\windows\syswow64\pluginmanager\mplugin.exe	2020-10-20T14:06:55Z	MPlugin.exe
c:\windows\syswow64\pluginmanager\plugin.exe	2020-10-20T14:06:55Z	Plugin.exe
c:\windows\syswow64\pluginmanager\xsdmplugin.exe	2020-10-20T14:06:55Z	XSDMPlugin.exe
c:\windows\syswow64\pluginmanager\xsdplugin.exe	2020-10-20T14:06:55Z	XSDPlugin.exe

Poco sopra sono state riportate le voci di registro presenti in Amcache con relativa data di ultima modifica delle stesse; si ricordi che essa coincide con la prima esecuzione degli applicativi.

Shimcache	
Name	LastModifiedTime
C:\ProgramFiles(x86)\SignTool\SignTool.exe	2020-07-08T09:45:08Z
C:\ProgramFiles(x86)\XYRZ\xyrzsvc.exe	2020-06-04T09:25:14Z
C:\Windows\SysWOW64\PluginManager\MPlugin.exe	2020-07-08T09:26:04Z
C:\Windows\SysWOW64\PluginManager\Plugin.exe	2020-07-08T09:26:08Z
C:\Windows\SysWOW64\PluginManager\XSDMPlugin.exe	2020-07-08T09:25:48Z
C:\Windows\SysWOW64\PluginManager\XSDPlugin.exe	2020-07-08T09:25:52Z

Come è possibile vedere dai dati sovrastanti, tutti gli eseguibili hanno sfruttato funzionalità relative agli strumenti di compatibilità di Windows e le corrispondenti tracce sono state rinvenute in Shimcache.

È importante notare come le date di ultima modifica, a differenza di quelle riportate in Amcache, non facciano riferimento all'ultima modifica delle voci di registro, ma a quella degli eseguibili a livello di file system.

NetstatEnriched						
Path	Type	Status	LAddrIp	LAddrPort	RAddrIp	RAddrPort
C:\ProgramFiles(x86)\SignTool\SignTool.exe	TCP	LISTEN	192.168.1.2	54355	0.0.0.0	0

A livello di connettività è stata individuata una connessione TCP in ascolto.

Il campo `LAddrIp` corrisponde all'IP del calcolatore nella rete locale. Il computer sembrerebbe essere in attesa di connessioni da parte del server remoto; l'indirizzo e la porta di quest'ultimo, visibili nel campo `RAddrIp` e `RAddrPort`, sono settate a zero, equivalente a dire che saranno accettate comunicazioni da parte di qualsiasi indirizzo o porta.

PsList			
Path	PID	Username	TokenIsElevated
`C:\ProgramFiles(x86)\SignTool\SignTool.exe`	2972	DESKTOP-VNSUD9Q/admin	true
`C:\ProgramFiles(x86)\XYRZ\xyrzsvc.exe`	9552	NT AUTHORITY/SYSTEM	true
`C:\Windows\SysWOW64\PluginManager\MPlugin.exe`	5700	NT AUTHORITY/SYSTEM	true
`C:\Windows\SysWOW64\PluginManager\Plugin.exe`	10448	NT AUTHORITY/SYSTEM	true
`C:\Windows\SysWOW64\PluginManager\XSDMPlugin.exe`	2800	NT AUTHORITY/SYSTEM	true
`C:\Windows\SysWOW64\PluginManager\XSDPlugin.exe`	4772	NT AUTHORITY/SYSTEM	true

Per ogni eseguibile individuato come malevolo è stato rinvenuto un processo in esecuzione con privilegi elevati, come riportato nel campo `TokenIsElevated`.

Prefetch			
Executable	LastRunTimes	RunCount	CreationTime
SIGNTOOL.EXE	[”2020-10-21T09:45:30Z”, ”2020-10-21T09:38:25Z”]	2	2020-10-21T09:38:25Z
MPLUGIN.EXE	[”2020-10-21T09:45:30Z”, ”2020-10-21T09:38:25Z”]	2	2020-10-21T09:38:25Z
PLUGIN.EXE	[”2020-10-21T09:45:30Z”, ”2020-10-21T09:38:25Z”]	2	2020-10-21T09:38:25Z
XSDMPLUGIN.EXE	[”2020-10-21T09:45:30Z”, 2020-10-21T09:38:25Z”]	2	2020-10-21T09:38:25Z
XSDPLUGIN.EXE	[”2020-10-21T09:45:30Z”, ”2020-10-21T09:38:25Z”]	2	2020-10-21T09:38:25Z
XYRZSVC.EXE	[”2020-10-21T09:45:52Z”, ”2020-10-21T09:39:52Z”, ”2020-10-21T09:38:32Z”]	3	2020-10-21T09:38:32Z

Grazie ai file di Prefetch, sono state individuate tutte le esecuzioni precedenti degli applicativi in esame.

L'indicatore del numero di avvii, come anche il `CreationTime`, potrebbero sembrare superflui; tuttavia risulterebbero di grande importanza qualora si fossero superati sette avvii listati nella colonna `LastRunTimes`, dato che, in tale caso, le date verrebbero sovrascritte a partire dalla più remota.

Autoruns			
ImagePath	Enabled	Cathegory	Profile
`c:\programfiles(x86)\signtool\signtool.exe`	enabled	Logon	System-wide
`c:\programfiles(x86)\xyrz\xyrzsvc.exe`	enabled	Services	System-wide
`c:\windows\syswow64\pluginmanager\xsdplugin.exe`	enabled	Services	System-wide
`c:\windows\syswow64\pluginmanager\plugin.exe`	enabled	Services	System-wide
`c:\windows\syswow64\pluginmanager\mplugin.exe`	enabled	Services	System-wide
`c:\windows\syswow64\pluginmanager\xsdmplugin.exe`	enabled	Services	System-wide

In relazione all'avvio automatico, ogni eseguibile ha aggiunto delle chiavi di registro[8] che rientrano nella categoria dei servizi e degli applicativi avviati al login.

3.2.2 figg.exe

Il secondo malware è nominato figg.exe[9]. Anche in questo caso, a seguito dell'installazione è stato eseguito un riavvio del calcolatore e si è provveduto a eseguire un clone che è stato usato come supporto di boot.

Una volta lanciata l'analisi sono stati prodotti i risultati visibili di seguito.

Sigcheck e Densityscout				
Path	VTDetection	Signature	Entropy	Density
`c:\users\admin\appdata\local\temp\gentekinc\GentekInc.exe`	58—74	Unsigned	7.05	0.95319

[8]Per questioni di spazio nella tabella non vengono riportate le chiavi di registro; in compenso si trascrivono i percorsi degli eseguibili al fine di agevolare il confronto con le tabelle precedenti.

[9] Il report è consultabile presso `https://web.archive.org/save/https://any.run/report/4753a049547fa90686a21d981602b4675228b2f7f49e6c7e9dccf8b06469f950/336a56c0-0a91-4d99-b284-3fb343ed1733`.

Come per il malware precedentemente osservato, pure questa volta l'eseguibile[10] è stato individuato dagli engine utilizzati da VirusTotal, l'applicativo risulta inoltre non firmato, con un'alta entropia e una bassa densità.

Amcache		
Binary	LastModified	Name
`c:\users\admin\appdata\local\temp\gentekinc\gentekinc.exe`	2020-10-20T14:06:55Z	Gentek Inc.exe

All'interno delle chiavi di registro relative alla Amcache è stata poi correttamente individuata la creazione dell'entry relativa alla prima esecuzione.

Shimcache	
Name	Time
`C:\Users\admin\AppData\Local\Temp\GentekInc\GentekInc.exe`	2020-10-20T01:55:48Z

Anche per questo applicativo è stato rilevato lo sfruttamento di strumenti di compatibilità di Windows.

In merito all'utilizzo della rete non sono stati prodotti risultati, per quanto dal report venisse evidenziata la presenza di connessioni con la rete esterna.

Di conseguenza è stata svolta un'analisi manuale delle connessioni, che non ha individuato flussi di traffico sospetto. Una possibile ipotesi relativamente a quanto emerso è che il programma apra connessioni ad intervalli temporali specifici, o che la mancanza di queste sia dovuta ad una impossibilità di contattare il server remoto a seguito della quale il collegamento viene chiuso per essere riaperto in seguito; senza un'analisi del codice sorgente non è però possibile avere certezze del motivo dell'assenza di connessioni.

PsList			
Path	PID	Username	TokenIsElevated
`C:\Users\admin\AppData\Local\Temp\GentekInc\GentekInc.exe`	6708	DESKTOP-VNSUD9Q/admin	true

Anche per questo malware è stato rinvenuto un processo in esecuzione tramite l'account amministratore.

[10]Si noti che, al contrario di quanto indicato nel report in nota [11], il programma relativo al malware ha un nome differente. Il cambio di nome è avvenuto a seguito del reboot della macchina e successivamente è rimasto figg.exe.

Prefetch			
Executable	LastRunTimes	RunCount	CreationTime
GENTEK INC.EXE	[”2020-10-21T09:55:43Z”, ”2020-10-20T14:06:55Z”]	2	2020-10-20T14:06:55Z

I file di Prefetch hanno poi consentito di individuare due esecuzioni precedenti del programma, avvenute in giorni distinti.

Autoruns			
ImagePath	Enabled	Cathegory	Profile
`c:\users\admin\appdata\local\temp\gentekinc\gentekinc.exe`	enabled	Logon	DESKTOP-VNSUD9Q/admin

Infine è stata individuata una entry, relativa a questo applicativo, tra i programmi che vengono eseguiti al login.

3.3 Test ImprovedLateralMovement

Per il test della seconda logica di correlazione, relativa al movimento laterale, sono stati sfruttati due strumenti: Infection Monkey[12] e APTSimulator[13].

Entrambi dispongono di specifiche funzionalità che li rendono buoni candidati per testare alcuni aspetti di un attacco prevedente il movimento laterale.

Infection Monkey è una piattaforma sviluppata al fine di svolgere l'attività di penetration testing di una rete in modo automatico e di valutare la propagazione di un attaccante tra i calcolatori di un'azienda.

L'esecuzione tipica prevede che su un computer venga eseguita una istanza di Infection Monkey, che agisce da server di comando e controllo per l'infezione. Il server esegue una scansione della rete e sugli host individuati vengono sfruttate alcune vulnerabilità conosciute (shellshock, sambacry, etc.[14]). In alternativa sarà tentato un attacco brute force

[12]Per ulteriori informazioni è possibile consultare la pagina `http://web.archive.org/web/20201108130207/https://github.com/guardicore/monkey`.

[13]Il progetto è disponibile presso `http://web.archive.org/web/20201101031231/https://github.com/NextronSystems/APTSimulator`.

[14]Per un elenco esaustivo delle vulnerabilità sfruttabili, e dei sistemi operativi per cui sono disponibili, si rimanda alla documentazione ufficiale di Infection Monkey.

mediante credenziali di uso comune o inserite dall'utilizzatore in fase di configurazione del server.

Una volta infettato un calcolatore vengono eseguite alcune operazioni, tra cui l'estrazione tramite mimikatz di credenziali in chiaro o di hash delle stesse. La successiva propagazione nella rete avviene aggiungendo quanto ottenuto all'elenco di username e password con le quali è possibile tentare l'autenticazione.

L'intero stato della propagazione attraverso la rete è visibile dinamicamente grazie ad una mappa reperibile sulla stessa pagina web da cui si esegue la configurazione iniziale del server di Infection Monkey.

APTSimulator è una piattaforma contenente script aventi l'obiettivo di simulare attività malevole. Nello specifico verrà utilizzato per testare le azioni intraprese da mimikatz, sfruttando l'esecuzione di una copia del programma senza che esso venga salvato sulla memoria di massa[15].

3.3.1 Infection Monkey

Il server di Infection Monkey è stato eseguito dalla stessa workstation dell'esaminatore utilizzata per i test della precedente logica di correlazione.

Durante la configurazione della piattaforma si sono selezionati SMB e Windows Management Infrastructure come metodi per tentare l'accesso ai dispositivi nella rete e sono state fornite alcune credenziali da utilizzare per i tentativi di brute force.

Una volta lanciata la procedura di movimento laterale è stato possibile vedere la progressione sulla mappa in figura 3.2.

Al termine dell'esecuzione di Infection Monkey è stata svolta una copia bitstream dell'intero supporto contenente l'installazione di ESXi, su cui ricordiamo essere contenute le macchine virtuali, ed è stato effettuato il boot del clone appena svolto; in seguito è stata

[15]Per un esempio dell'utilizzo della powershell al fine di eseguire applicativi senza che essi vengano salvati sulla memoria secondaria si rimanda a `https://web.archive.org/web/20200924195608/https://clymb3r.wordpress.com/2013/04/06/reflective-dll-injection-with-powershell/` e anche `https://web.archive.org/web/20191027105153/https://medium.com/@Bank_Security/technical-guide-for-insider-cyber-attacks-9583f0d6325f`.

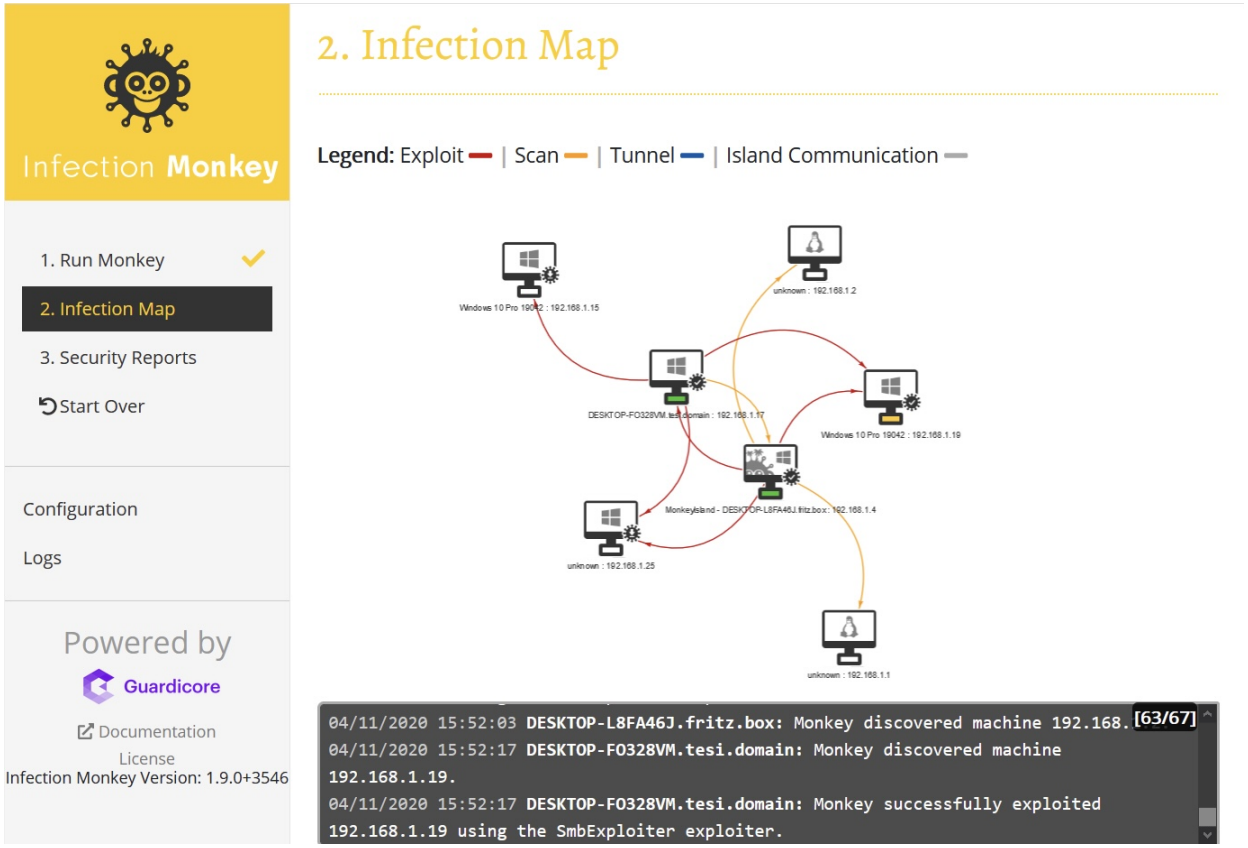

Figura 3.2: Movimento laterale all'interno della rete.

eseguita la logica di correlazione `ImprovedLateralMovement` e sono stati visualizzati i risultati dei computer[16].

FailedLogon					
SystemTime	FailureReason	Username	LogonType	IpAddress	IpPort
1604504894.9656992	%%2313	Utente	3	192.168.1.4	51299
1604505141.1736655	%%2313	User	3	192.168.1.17	49409
1604505445.3223472	%%2313	Administrator	3	192.168.1.15	64920

[16]Per non appesantire eccessivamente il presente lavoro i dati riportati a seguito dell'analisi saranno parziali e riguarderanno il calcolatore avente indirizzo IP 192.168.1.19.

Come riportato nella tabella precedente, è possibile notare alcuni tentativi di autenticazione falliti da parte di differenti calcolatori all'interno della rete.

Durante il brute force sono stati utilizzati tre username differenti; in tutti i casi il motivo del fallimento ha codice %%2313, che indica l'impiego di nomi utente inesistenti o password errate durante il login.

SuccessfulLogon					
SystemTime	TargetLogonId	Username	LogonType	IpAddress	IpPort
1604504901.1468832	1293087	Administrator	3	192.168.1.4	51315
1604505141.64228	1753794	Administrator	3	192.168.1.17	49415
1604505445.7744737	2862430	Administrator	3	192.168.1.15	64925

A seguito dei tentativi di accesso non riusciti, sono stati individuati alcuni login avvenuti con successo riguardanti l'utente Administrator; ciò indica che tra le credenziali utilizzate durante i tentativi di brute force vi era anche una coppia di username e password corretti.

SpecialPrivilegesLogon			
SubjectLogonId	Username	SubjectDomainName	Privileges
1293087	Administrator	TESI	SeSecurityPrivilege SeBackupPrivilege SeRestorePrivilege SeTakeOwnershipPrivilege SeDebugPrivilege SeSystemEnvironmentPrivilege SeLoadDriverPrivilege SeImpersonatePrivilege SeDelegateSessionUserImpersonatePrivilege
1753794	Administrator	TESI	”
2862430	Administrator	TESI	”

Per quanto già partire dall'username dell'utente coinvolto si possa sospettare che l'account compromesso sia dotato di privilegi elevati, ciò viene confermato dal fatto che il `SubjectLogonId` è stato utilizzato con successo per trovare informazioni tra i log di sistema riguardanti le autenticazioni di utenti con privilegi elevati.

Come è possibile osservare dalla colonna `Privileges`, l'utente Administrator può svolgere importanti operazioni come, ad esempio, caricare o disabilitare driver di sistema e impersonare altri utenti[17].

EnrichedNetstats								
PID	Path	Username	Type	Status	Laddr.IP	Laddr.Port	Raddr.IP	Raddr.Port
6988	`C:\Windows\Temp\monkey64.exe`	NT AUTHORITY/SYSTEM	TCP	LISTEN	0.0.0.0	26384	0.0.0.0	0
6988	`C:\Windows\Temp\monkey64.exe`	NT AUTHORITY/SYSTEM	TCP	SENT	192.168.1.19	64077	192.168.1.171	3389
6988	`C:\Windows\Temp\monkey64.exe`	NT AUTHORITY/SYSTEM	TCP	SENT	192.168.1.19	64078	192.168.1.171	445
6988	`C:\Windows\Temp\monkey64.exe`	NT AUTHORITY/SYSTEM	TCP	SENT	192.168.1.19	64079	192.168.1.171	22

A livello di connettività è possibile osservare come il processo con PID 6988, facente riferimento all'eseguibile monkey64.exe, abbia una connessione in ascolto su qualsiasi interfaccia di rete del calcolatore[18] e abbia scansionato l'indirizzo IP 192.168.1.171 sulle porte 22 (SSH), 445 (SMB) e 3389 (RDP).

Amcache e Shimcache			
Name	Binary	Amc.LastModified	Shim.LastModified
wmiprvse.exe	`C:\Windows\system32\wbem\wmiprvse.exe`	2020-11-03T11:28:54Z	2020-10-22T08:13:41Z
mofcomp.exe	`C:\Windows\system32\wbem\mofcomp.exe`	2020-11-03T11:20:41Z	2020-10-22T08:13:09Z
net.exe	`C:\Windows\system32\net.exe`	2020-11-03T11:27:23Z	2019-12-07T09:09:33Z
whoami.exe	`C:\Windows\system32\whoami.exe`	2020-11-03T11:27:45Z	2019-12-07T09:09:51Z

Nella tabella superiore è possibile visionare alcuni dei programmi dei quali è stata loggata l'esecuzione in Amcache e Shimcache. Gli eseguibili sono relativi alla WMI nonchè a strumenti per l'analisi di rete e dei privilegi relativi agli utenti.

Prefetch			
Executable	LastRunTimes	RunCount	CreationTime
WMIPRVSE.EXE	[2020-11-04T15:26:12Z, 2020-11-04T12:26:45Z, 2020-11-04T10:21:44Z, 2020-11-03T14:54:33Z, 2020-11-03T11:28:44Z]	5	2020-11-03T11:28:54Z

[17]Per una chiara disamina dei privilegi si rimanda a `http://web-old.archive.org/web/20201118120150/https://docs.microsoft.com/it-it/windows/security/threat-protection/auditing/event-4672`.

[18]Si fa riferimento alla prima riga della tabella precedente.

MOFCOMP.EXE	[2020-11-03T11:20:41Z]	1	2020-11-03T11:20:41Z
NET.EXE	[2020-11-04T15:26:30Z, 2020-11-04T12:25:44Z, 2020-11-03T11:27:23Z]	3	2020-11-03T11:27:23Z
WHOAMI.EXE	[2020-11-04T15:24:24Z, 2020-11-04T12:23:37Z, 2020-11-03T11:27:45Z]	3	2020-11-03T11:27:45Z

L'analisi dei file di Prefetch ha infine prodotto ulteriori dati per gli stessi applicativi rinvenuti in Amcache e Shimcache.

3.3.2 APTSimulator

Una volta scaricata su uno dei calcolatori in analisi la suite APTSimulator, è stato lanciato lo script per testare l'utilizzo di mimikatz senza il salvataggio dell'eseguibile in memoria secondaria.

In seguito la logica di correlazione in esame è stata mandata nuovamente in esecuzione, consentendo di individuare, grazie alla cronologia dei comandi della powershell mantenuta da windows, l'azione svolta dallo script di APTSimulator. Il risultato prodotto è riportato nella tabella sottostante.

PowershellDump	
System	EventData
Provider : "Name":"PowerShell" EventID : "Qualifiers":0,"Value":600 Version : 0 Level : 4 Task : 6 Opcode : 0 Keywords : 36028797018963970 TimeCreated : "SystemTime":1604485488.6258688 EventRecordID : 479 Execution : "ProcessID":0,"ThreadID":0 Channel : Windows PowerShell Computer : DESKTOP-FO328VM.tesi.domain	powershell.exe iex (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/ PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1');Invoke-Mimikatz -DumpCreds

Conclusioni

L'obiettivo del presente lavoro era quello di sviluppare alcune logiche di correlazione che fossero semplicemente espandibili, riutilizzabili e che potessero risultare un valido supporto per la fase di analisi di un sistema.

Siamo quindi partiti, nel primo capitolo, da un inquadramento giuridico e metodologico dell'informatica forense, premessa ineludibile per fornire la cornice normativa e tecnica entro la quale si muove l'informatico forense nella suddettà attività.

Precisamente, dopo un esame dei vari tipi di *computer crimes*, legati al fenonemo della informatizzazione massiva della società contemporanea, abbiamo fornito le nozioni definitorie della scienza dell'informatica forense. Abbiamo poi sviscerato i vari ambiti tecnici che essa ricopre (*Disk Forensics*, *Embedded Forensics*, *Mobile Forensics*, *Network Forensics* e *Cloud Forensics*).

Nel passare poi a considerare i principi e gli obiettivi dell'informatica forense, abbiamo concentrato la nostra attenzione sul tema della modificabilità del dato, peculiarità tipica di quello informatico, osservando come non tenerne conto possa condurre alla creazione non solo di una cattiva scienza, ma anche di cattive pratiche. In tal senso, come concreto esempio di queste ultime, abbiamo sommariamente illustrato il caso Garlasco.

Poichè proprio al fine di evitare *malpractices* è scaturito un articolato quadro normativo - giuridico e tecnico -, ci siamo quindi addentrati nell'analisi dello stesso. In particolare, per le norme del diritto, abbiamo esaminato dapprima la Convenzione sul *Cybercrime*, evidenziando i profili di specifico interesse per il presente lavoro (ossia l'art. 19), e poi la relativa legge italiana di ratifica della stessa, cioè la legge 18 marzo 2008, n. 48, con una puntuale disamina di tutte quelle norme del codice di procedura penale italiano, revisionate in seguito alla sua entrata in vigore, nelle quali si è andati a esplicitare

in modo particolarmente evidente l'affermazione di quel principio di "non alterazione" del dato che fa *pendant* alla suddetta sua modificabilità.

Poichè peraltro il legislatore non si è espresso su quali siano le procedure più adatte per preservare i dati, di fatto rinviando a una tecno-regolazione, l'esame degli standard tecnici ha costituito la naturale prosecuzione della riflessione. In particolare, ci siamo soffermati sugli standard ISO/IEC 27037:2012 e ISO/IEC 27042:2015, il primo contenente le linee guida per l'identificazione, raccolta, acquisizione, conservazione delle prove digitali e il secondo quelle per lo stadio immediatamente successivo di analisi e interpretazione delle stesse.

Da ultimo, sotto il profilo del metodo, dopo aver riferito - in modo peraltro critico - la posizione della giurisprudenza italiana in materia di copia forense, il *focus* è stato portato sulle varie possibilità operative che si offrono all'informatico forense in fase di acquisizione del dato, dividendole fondamentalmente nelle due categorie di *post mortem* e *live*, per soffermarsi infine sugli argomenti della copia *bitstream*, della copia logica e dell'acquisizione della memoria volatile.

Nei seguenti due capitoli ci siamo addentrati sull'oggetto specifico della tesi, ossia lo sviluppo delle logiche di correlazione rispondenti alle caratteristiche sopra descritte.

In particolare, nel secondo capitolo, preso atto che l'analisi dei dati costituisce oggi in un certo qual senso una sfida, considerato che la loro mole è notevolmente incrementata rispetto al passato a causa dell'aumento considerevole della capacità delle memorie di massa, siamo partiti con l'illustrare il *Computer Forensics Field Triage Process Model*, uno dei più significativi esempi di metodologia operativa per velocizzare il processo investigativo, sebbene il dibattito sia aperto sulla possibilità di considerarlo parte dell'informatica forense.

Quindi, dopo aver analizzato lo stato dell'arte dell'analisi automatizzata, aver rapidamente relazionato su un prototipo di applicativo, denominato Racoon, da me sviluppato in Python durante l'attività di tirocinio, e svolto alcune considerazioni sull'analisi *live*, è stato scelto, a motivo delle sue caratteristiche tecniche, Velociraptor come base per il progetto. L'applicativo infatti realizza l'attività di analisi grazie a specifici file chiamati artefatti, scritti secondo la sintassi YAML, il funzionamento dei quali è stato quindi spiegato. Esso - che può essere eseguito secondo due modalità esposte nel presente lavoro

(ossia secondo un'architettura client-server e in triage mode) - si è rivelato una piattaforma potente, flessibile e in grado di offrire un linguaggio di interrogazione, il *Velociraptor Query Language*, grazie al quale è stato possibile codificare in modo semplice potenti *query*, che potranno essere riutilizzate in più indagini e modificate per adattarsi meglio al contesto di esecuzione.

L'utilizzo degli strumenti sopra citati permette a un investigatore forense di interrogare i dispositivi in esame, attraverso l'elaborazione di logiche di correlazione, e di ottenere da essi un sottoinsieme ristretto di risultati desiderati; ciò consente di abbassare drasticamente i tempi di lavoro umano, permettendo così all'utilizzatore di dedicarsi maggiormente alla fase di interpretazione e presentazione dei dati ottenuti.

Abbiamo quindi sviluppato due logiche di correlazione: `BasicProgramInspection` (relativamente alla quale si è riutilizzato parte del lavoro del tirocinio) e `ImprovedLateral Movement`. La prima aveva l'obiettivo principale di individuare file e programmi sospetti al fine di ottenere informazioni aggiuntive sulle esecuzioni precedenti; la seconda il c.d. movimento laterale, ossia l'ingresso di un soggetto non autorizzato all'interno dell'infrastruttura di rete aziendale e il suo movimento all'interno di questa.

Per entrambe queste logiche abbiamo quindi dettagliato in questa sede le modalità di funzionamento.

Nel terzo capitolo si è infine proceduto a illustrare i risultati dei test delle logiche di correlazione progettate, effettuati creando due ambienti distinti, ovvero un'installazione di Windows 10 Pro su un singolo calcolatore e un'installazione di ESXi 7.0 come *hypervisor* per quattro macchine virtuali contenenti un'istanza di Windows Server 2019 e tre di Windows 10 Pro.

La risposta è stata assolutamente soddisfacente, in quanto nei test effettuati sono stati individuati con successo i dati di esecuzione di alcuni *malware* su un sistema e le tracce di movimento laterale nella riproduzione di un *network* aziendale. Più specificatamente, la logica di correlazione `BasicProgramInspection` ha individuato le attività del *malware* EcV01.04.R.exe e del *malware* figg.exe (entrambi scaricati dal portale `any.run`), mentre la logica di correlazione `ImprovedLateralMovement` ha individuato le azioni svolte dai due strumenti Infection Monkey (una piattaforma sviluppata al fine di svolgere l'attività di *penetration testing* di una rete in modo automatico e di valutare la propagazione di un

attaccante tra i calcolatori di un'azienda) e APTSimulator (utilizzato in questo lavoro per testare l'attività di mimikatz mediante un suo *script* che sfrutta l'esecuzione di una copia del programma senza che esso venga salvato sulla memoria di massa).

In conclusione, quanto sviluppato ha consentito di automatizzare operazioni di ricerca che, al contrario, avrebbero richiesto un intervento manuale da parte di un operatore e - a fronte di una breve e opzionale configurazione iniziale per adattare le logiche di correlazione alle proprie necessità - ridurre a pochi minuti il tempo impiegato per filtrare i dati.

Sulla base dei risultati ottenuti a seguito della fase di test, l'obiettivo fissato all'inizio del presente lavoro può pertanto considerarsi raggiunto.

Tuttavia, pur essendo semplicemente espandibili e modificabili, le logiche sviluppate, in quanto concepite per risolvere i problemi della ricerca di *malware* e di individuazione di movimento laterale, hanno un grado di specificità che ne confina l'utilizzo a questi due particolari domini.

Eventuali sviluppi futuri del presente lavoro potrebbero consistere nella creazione di una più vasta libreria di artefatti da poter sfruttare come base di partenza per creare più agevolmente ulteriori logiche di correlazione adattabili ai casi che un informatico forense può trovarsi a dover esaminare, proseguendo così il percorso qui iniziato di alleggerimento del suo lavoro umano.

Bibliografia

[1] ANSA, cur. *Omicidio in Florida, Alexa unico testimone*. 2019. URL: https://web.archive.org/web/20200208105432/https://www.ansa.it/sito/notizie/tecnologia/hitech/2019/11/07/omicidio-in-florida-alexa-testimone_8ab45007-2f20-440a-8e5e-cb531a1c40da.html.

[2] G. Cantell D. Dampier Y. S. Dandass N. Niu C. Bogen. «Research toward a Partially-Automated, and Crime Specific Digital Triage Process Model». In: *Computer and Information Science* 5 (2012), pp. 29–38. URL: https://web.archive.org/web/20201022200025/https://www.researchgate.net/publication/278668325_Research_Toward_a_Partially-automated_and_Crime_Specific_Digital_Triage_Process_Model.

[3] R. Brighi. «Una governance integrata per nuovi modelli dell'informatica forense». In: *i-lex. Scienze Giuridiche, Scienze Cognitive e Intelligenza Artificiale* (2017), pp. 47–70. URL: http://web.archive.org/web/20200923205251/http://www.i-lex.it/articles/volume11/fascicolo1/brighi_governance_integrata.pdf.

[4] R. Brighi C. Maioli. «Un cambio di paradigma nelle scienze forensi. Dall'armonizzazione tecnico-giuridica a una nuova cornice epistemologica.» In: *Informatica e diritto* 24 (2015), pp. 217–234. URL: https://web.archive.org/web/20200923205024/http://www.ittig.cnr.it/EditoriaServizi/AttivitaEditoriale/InformaticaEDiritto/IeD_2015_1-2_BrighiMaioli.pdf.

[5] M. K. Rogers J. Goldman R. Mislan T. Wedge S. Debrota. «Computer Forensic Field Triage Process Model». In: *Journal of Digital Forensics, Security and Law* 1 (2006), pp. 19–38. URL: https://web.archive.org/web/20201022195157/

https://www.researchgate.net/publication/288761713_Computer_Forensics_Field_Triage_Process_Model.

[6] O. M. Fal'. «Standardization in Information Technology Security». In: *Cybernetics and Systems Analysis* 53 (2017), pp. 78–82.

[7] FBI, cur. *Digital Evidence: Standards and Principles*. 2000. URL: https://web.archive.org/web/20190331095748/https://archives.fbi.gov/archives/about-us/lab/forensic-science-communications/fsc/april2000/swgde.htm.

[8] M. Ferrazzano. *Aspetti metodologici, giuridici e tecnici nel trattamento di reperti informatici nei casi di pedopornogradia*. Aracne, 2018.

[9] M. Ferrazzano. «Indagini forensi in tema di scambio di file pedopornografici mediante software di file sharing a mezzo peer-to-peer». Tesi di Dottorato. Alma Mater Studiorum - Università di Bologna, 2014. URL: https://web.archive.org/web/20200716074703/http://amsdottorato.unibo.it/6697/1/ferrazzano_michele_tesi.pdf.

[10] A. Gammarota. «Informatica forense e processo penale: la prova digitale tra innovazione normativa e incertezze giurisprudenziali». Tesi di Dottorato. Alma Mater Studiorum - Università di Bologna, 2016. URL: https://web.archive.org/web/20200822005439/http://amsdottorato.unibo.it/7723/1/Gammarota_Antonio_tesi.pdf.

[11] G. Fagioli A. Ghirardini. *Digital forensics*. A cura di Apogeo. 2013.

[12] ISO/IEC, cur. *Guidelines for identification, collection, acquisition and preservation of digital evidence*. 2012.

[13] ISO/IEC, cur. *Guidelines for the analysis and interpretation of digital evidence*. 2015.

[14] H. Mohammed N. Clarke F. Li. «An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data». In: *Journal of Digital Forensics, Security and Law* 11 (2016), pp. 137–152. URL: https://web.archive.org/web/20201112105234/https://www.researchgate.net/publication/308903458_

An_Automated_Approach_for_Digital_Forensic_Analysis_of_Heterogeneous_Big_Data.

[15] X. Fu X. Du B. Luo. «Data correlation-based analysis methods for automatic memory forensic». In: *Security and Communication Networks* 9 (2015), pp. 4213–4226.

[16] C. Maioli. *Dar voce alle prove: elementi di informatica forense*. 2004. URL: https://web.archive.org/web/20170209074130/http://informaticaforense.it/materiali2011/Maioli-Darvocealleprove.pdf.

[17] K. Chen A. Clark O. De Vel G. Mohay. *Event Correlation for Forensics*. 2003. URL: https://web.archive.org/web/20201112105932/https://core.ac.uk/download/pdf/10887935.pdf.

[18] NIST, cur. *Guidelines on PDA Forensics*. 2004. URL: https://web.archive.org/web/20200509115250/https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-72.pdf.

[19] Osservatorio permanente sulla criminalità organizzata, cur. *Convenzione del consiglio d'Europa sulla criminalità informatica*. 2001. URL: https://web.archive.org/web/20200822120733/https://www.garanteprivacy.it/documents/10160/10704/CONSIGLIO+D'EUROPA+-+Convenzione+sulla+criminalit%C3%A0+informatica.pdf/1b436196-4064-4dc7-91f7-a9a3dbf65c0c?version=1.1.

[20] L. Picotti. «La nozione di criminalità informatica e la sua rilevanza per le comptenze penali europee». In: *Riv. Trim. dir. Pen. Ec.* (2011), p. 827 ss.

[21] A. Case A. Cristina L. Marziale G. G. Richard V. Rousseu. «FACE: Automated digital evidence discovery and correlation». In: *Digital Investigation* 5S (2008), pp. 65–75. URL: http://web.archive.org/web/20201112111141/https://www.sciencedirect.com/science/article/pii/S1742287608000340.

[22] A. R. Arasteh M. Debbabi A. Sakha M. Saleh. «Analyzing multiple logs for forensic evidence». In: *Digital Investigation* 4S (2007), pp. 82–91. URL: https://web.archive.org/web/20201112104727/https://www.sciencedirect.com/science/article/pii/S1742287607000448.

[23] E. Gahrmanov V. Jusas D. Birvinskas. «Methods and Tools of Digital Triage in Forensic Context: Survery and Future Directions». In: *Symmetry* 9 (2017), pp. 1–19. URL: https://web.archive.org/save/https://www.researchgate.net/publication/315928408_Methods_and_Tools_of_Digital_Triage_in_Forensic_Context_Survey_and_Future_Directions.

[24] R. Martell V. Roussev C. Quates. «Real-time digital forensics and triage». In: *Digital Investigation* 10 (2013), pp. 158–167.

[25] Volatility, cur. *TrueCrypt Master Key Extraction And Volume Identification*. 2014. URL: https://web.archive.org/web/20191222072540/https://volatility-labs.blogspot.com/2014/01/truecrypt-master-key-extraction-and.html.

[26] Wikipedia, cur. *FBI - Apple encryption dispute*. 2020. URL: https://web.archive.org/web/20200719200744/https://en.wikipedia.org/wiki/FBI%E2%80%93Apple_encryption_dispute.

Ringraziamenti

Dietro ad ogni progetto vi sono delle persone e questi tre anni di università sono stati pesantemente influenzati da esse.

Grazie a Shine per avermi avviato alla programmazione; se non fosse stato per le sue spiegazioni, e per i suoi consigli, la strada per imparare a programmare sarebbe certamente stata più impervia.

Grazie al gruppo JavaTheHutt per aver collaborato a quello che è stato il primo nostro grande progetto della triennale e di cui conserverò per sempre un ricordo indelebile.

Grazie al Dott. Alessandro Di Carlo di Bit4Law per il tempo dedicatomi e le spiegazioni tecniche fornitemi durante il tirocinio e la stesura del presente lavoro.

Grazie alla Prof.ssa Raffaella Brighi per aver supervisionato il presente lavoro di tesi.

Grazie alla mia famiglia per avermi sostenuto in questi anni di Università, supportandomi nei momenti di bisogno e di maggiore stress.

Grazie infine a Matteo e ad Angela per l'amicizia che mi hanno sinceramente donato in questi anni e per essere stati esempi di come la forza di volontà sia alla base di ogni grande viaggio.