UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO DI INFORMATICA

Corso di laurea in Sicurezza Informatica

Tesi di laurea in Informatica Forense

Analisi, Progettazione e Realizzazione di Tiny Core Forensic Edition

Live Distro Linux forense minimale specializzata in acquisizione

Relatore:
Ch.mo Prof. Ugo Lopez

Laureanda:
Dott.ssa Sara Ruffo

ANNO ACCADEMICO 2019/2020

Ringraziamenti

Alla fine di questo mio percorso desidero ringraziare tutti coloro che mi hanno supportato e sopportato in questi anni.

Ringrazio il mio relatore, il professor Ugo Lopez che con pazienza mi ha guidato durante questo progetto. E per l'aiuto fornitomi, ringrazio il dottor Nanni Bassetti, esperto in materia forense, che mi ha fornito numerosi e validi consigli. Ringrazio anche Lorenzo Giustiniani, Cosimo Di Pinto e tutto lo staff aziendale, Michele Troccoli e Marilena Porfido che mi hanno assistita nei vari processi burocratici.

Ringrazio tutti i professori che ho incontrato in questo mio cammino. In particolar modo il professor Malerba, grande fonte di ispirazione.

Desidero ringraziare i miei genitori per avermi incoraggiata sempre e mia sorella per essere la persona più importante della mia vita. Senza di te la mia vita sarebbe infinitamente più noiosa!

Ringrazio tutti i colleghi universitari e tutti i miei amici e in particolar modo i soliti Paola, Nicola, Ciccio, Alessia e Alessio, compagni di una vita per una vita.

Un grazie immenso va alla mia Co. Ca. e a tutto il gruppo scout Taranto 15, per credere in me molto più di quanto non faccia io stessa. Grazie a tutti gli scout che ho incontrato nel mio sentiero, ognuno di loro ha contribuito a formare ciò che sono.

The last but not the least, ringrazio Dio per avermi dato la capacità e la forza di portare a termine questo percorso e a San Giuseppe da Copertino, che mi ha accompagnato per questa strada.

Sommario

Introduzione

Panoramica della scienza forense

La scienza forense[1] consiste nell'applicazione di tecniche e metodologie scientifiche alle tradizionali investigazioni allo scopo di identificare, preservare, recuperare, analizzare e presentare le prova nel corso di un'indagine.

Le origini della moderna scienza forense si fanno risalire all'ufficiale di polizia francese Alphonse Bertillon[2], il quale introdusse un sistema di documentazione fotografica della scena del crimine. In seguito, applicò una tecnica antropometrica creando così un sistema di identificazione basato sull'analisi di misurazioni fisiche. Tale sistema ha successivamente assunto il nome di «sistema Bertillon»

Nel 1858, Sir William James Herschel[3] ha iniziato a promuovere l'uso delle impronte digitali per l'identificazione dei sospetti criminali. Durante il suo Servizio Civile in India, ha iniziato a utilizzare impronte digitali sui documenti come misura di sicurezza per evitare il ripudio, allora dilagante, di firme.

L'analisi forense del DNA è stata sviluppata nel 1984 da Sir Alec Jeffreys[4] che ha evidenziato che la variazione nel codice genetico potrebbe essere utilizzata per identificazione personale, (fingerprinting genetico).

---

1 https://it.wikipedia.org/wiki/Scienza_forense#:~:text=La%20scienza%20forense%20%C3%A8%20l,o%20ad%20un%20comportamento%20sociale.

2 https://en.wikipedia.org/wiki/Alphonse_Bertillon

3 https://en.wikipedia.org/wiki/Sir_William_Herschel,_2nd_Baronet

4 https://en.wikipedia.org/wiki/Alec_Jeffreys

La scienza forense pone il suo fondamento sul principio espresso da Edmond Locard[5], che evidenzia che "Non si può entrare e/o uscire da un posto senza lasciare qualcosa di sé". Nel 1910, fondò quello che potrebbe essere definito il primo laboratorio criminalistico del mondo. L'informatica forense è una branca della scienza digitale forense legata alle prove acquisite da computer e altri dispositivi digitali. Il suo scopo è quello di esaminare dispositivi digitali seguendo processi di analisi forense al fine di identificare, preservare, recuperare, analizzare e presentare fatti o opinioni riguardanti le informazioni raccolte.

Il concetto di informatica forense emerge nei primi anni 80, con l'avvento dei primi Personal Computer. Parallelamente all'aumento del numero di PC, cresce il numero di crimini identificati e riconosciuti come crimini informatici. L'informatica forense nasce quindi inizialmente come metodologia per il recupero e l'analisi di prove digitali da utilizzare di fronte ad un tribunale.

Al giorno d'oggi l'informatica forense non è più solo uno strumento di analisi di prove ma uno dei principali mezzi di investigazione legato a diversi crimini ed illeciti.

Storia dell'informatica forense

L'analisi storica della disciplina può aiutare a comprendere come il settore si sia evoluto e fornisce un ulteriore contesto per alcuni problemi e sfide che gli addetti ai lavori si trovano ad affrontare.

Pre-2000

[5] https://en.wikipedia.org/wiki/Edmond_Locard

I primi studi di digitalforensics (allora computer forensics) sono stati svolti a partire dal 1980. In questo periodo, la crescita dei computer domestici ha innescato un repentino interesse per la scienza forense informatica

Uno tra i primi progetti nel campo è stato sviluppato dall'FBI che ha portato alla creazione di un programma per analizzare le prove informatiche. Nel 1988 nacque Computer Emergency Response Team (CERT)[6] presso la Carnegie Mellon University di Pittsburgh a seguito dell'aumento di attacchi e abusi informatici.

Una crescita esponenziale dell'accesso ad internet e dei personal computer ha caratterizzato gli anni '90. In questo periodo è stato coniato il concetto di crimine informatico ed è stato sviluppato il primo software forense opensource chiamato The Coroner's Toolkit (TCT)[7], creato da Dan Farmer e Wietse Venema. TCT è una raccolta di programmi gratuiti per l'analisi post mortem di alcuni sistemi UNIX (FreeBSD, OpenBSD, etc.)[8].

2000–2010

All'inizio nel nuovo millennio, molti fattori hanno contribuito a far crescere la necessità di digital forensics. La crescita della tutela della proprietà intellettuale, l'aumento di fenomeno di frodi su internet e la

[6] https://en.wikipedia.org/wiki/Computer_emergency_response_team
[7] http://www.porcupine.org/forensics/tct.html
[8] https://en.wikipedia.org/wiki/The_Coroner%27s_Toolkit

comparsa del phishing hanno creato un'ulteriore richiesta di indagini e raccolta di prove informatiche.

Con la conferenza DFRWS del 2001[9] si è arrivati a una delle prime definizioni della digital forensic: *"La digital forensic è l'uso di metodi scientificamente derivati e dimostrati per la conservazione, la raccolta, la validazione, l'identificazione, l'analisi, l'interpretazione, la documentazione e la presentazione di prove digitali derivate da fonti digitali allo scopo di favorire la ricostruzione di eventi ritenuti criminali o aiutare ad anticipare azioni non autorizzate che si sono rivelate di disturbo alle operazioni pianificate".*

2010–Presente

Una serie di eventi, a partire dal 2010, ha spostato l'attenzione sull'investigazione e la raccolta di prove di attacchi informatici e violazione di dati.

Tra i casi di violazione della privacy, quelli più noti sono WikiLeaks[10] che ha pubblicato materiale trapelato dall'esercito statunitense, compresi video e comunicati diplomatici. In questo periodo Anonymous acquisita notorietà grazie ai suoi attacchi DDoS (Distributed Denial-of-Service)[11] e altre attività dannose. È stata resa pubblica la portata dello spionaggio governativo che utilizza il malware contro altri governi e industrie private ed è venuto alla luce anche il

[9] https://dfrws.org/conferences/dfrws-usa-2001/
[10] http://www.wikileaks.org/
[11] https://en.wikipedia.org/wiki/Denial-of-service_attack

worm Stuxnet[12] che ha preso di mira i sistemi SCADA[13], in particolare i sistemi di controllo del programma nucleare iraniano.
La diffusione dei dati da parte della società italiana Hacking Team[14] ha rivelato che una grossa fetta del mercato degli exploit professionali viene venduto a governi, forze dell'ordine e aziende del settore privato e nello stesso periodo l'industria bancaria globale ha dovuto affrontare un forte aumento del malware di tipo bancario (Zeus[15], Sinowal/Torpig[16], SpyEye[17], Gozi[18], Dyre[19], Dridex[20] e altri), che ha preso di mira con successo i clienti bancari a scopo di frode finanziaria.
Di recente, sono divenuti popolari gli attacchi ransom (Ransomware[21], DDoS per Bitcoin[22], ecc.)

Informatica forense nel diritto italiano

Queste vicende hanno inevitabilmente portato l'informatica ad approdare anche nelle aule di tribunale ma l'ingresso dell'informatica nel campo del diritto è subordinato al rispetto di regole che disciplinano tutte le varie fasi di un processo.

Questo ha portato alla nascita di discipline che operano in diversi ambiti del diritto.

[12] https://en.wikipedia.org/wiki/Stuxnet
[13] https://en.wikipedia.org/wiki/SCADA
[14] https://www.agi.it/estero/storia_hacking_team_omicidio_khashoggi-4720850/news/2018-12-10/
[15] https://en.wikipedia.org/wiki/Zeus_(malware)
[16] https://en.wikipedia.org/wiki/Torpig
[17] https://en.wikipedia.org/wiki/SpyEye
[18] https://malpedia.caad.fkie.fraunhofer.de/details/win.gozi
[19] https://www.secureworks.com/research/dyre-banking-trojan
[20] https://en.wikipedia.org/wiki/Dridex
[21] https://en.wikipedia.org/wiki/Ransomware
[22] http://www.revistaespacios.com/a20v41n03/a20v41n03p29.pdf

L’informatica giuridica è una delle discipline con cui l’informatica approda nell’ambito giuridico. L’informatica giuridica è la disciplina che utilizza i calcolatori elettronici nel campo del diritto, in questo caso quindi l’informatica viene usata come mezzo.

Essa si concentra sugli strumenti informatici a disposizione del giurista, sulla redazione di ipertesti giuridici e sulle applicazioni più in generale utili per la documentazione e la consultazione nell'ambito della giurisprudenza.

La scienza che studia i problemi giuridici legati all’informatica, viene denominata diritto dell’informatica, in questo caso l’informatica è l’oggetto che dà luogo ad una regolamentazione sull’uso degli elaboratori e sulle sue conseguenze.

L’informatica forense, invece, è una branca della scienza digitale forense. È legata alle prove acquisite da dispositivi digitali e attraverso la loro analisi forense persegue lo scopo di identificare, preservare, recuperare, analizzare e presentare fatti e opinioni riguardanti le informazioni raccolte.[23]

Nel diritto italiano la digital forensics si occupa quindi di rendere fruibile in ambito processuale le risultanze informatiche e permette di attribuire un valore probatorio o istruttorio alle informazioni ricavate su un insieme di dati, attraverso operazioni scientifico-investigative.

[23] https://it.wikipedia.org/wiki/Informatica_forense

Una serie di rami dell'informatica forense definisce gli specifici ambiti e strumenti analizzati, alcuni di questi sono:

- Computer forensics: ramo dell'informatica forense che si occupa dell'analisi forense di computer, server e simili;
- Mobile forensics: ramo dell'informatica forense che si occupa dell'analisi forense di smartphone e simili;
- Network forensics: ramo dell'informatica forense che si occupa dell'analisi forense di reti di calcolatori;
- Cloud forensics: ramo dell'informatica forense che si occupa dell'analisi forense di un'infrastruttura cloud;
- Incident response forensics: ramo dell'informatica forense che si occupa dell'analisi forense di un sistema compromesso;
- Open Source Intelligence (OSInt): ramo dell'informatica forense che si occupa della raccolta di informazioni attraverso l'utilizzo di fonti di pubblico accesso;

Le cinque fasi di un'analisi forense sono: l'identificazione, la conservazione, l'acquisizione, l'analisi e la presentazione[24].

In genere, un software è considerato "forense" quando assicura la non alterazione del reperto nonché garanzie in merito alla catena di custodia.

Alcuni elementi considerati indispensabili nella "costruzione" della prova scientifica, e di conseguenza anche di quella informatica, sono condensati alcuni punti cardine, universalmente riconosciuti come i criteri Daubert[25]:

[24] https://www.dmi.unict.it/~battiato/CF1213/05.pdf
[25] https://en.wikipedia.org/wiki/Daubert_standard

- la verificabilità del metodo - una teoria è scientifica solo se controllabile mediante esperimenti. La stessa ricerca effettuata da un altro ricercatore con gli stessi metodi dovrà portare allo stesso risultato;
- la falsificabilità - sottoporsi a tentativi di smentita ed individuare con certezza i limiti della stessa teoria, per questo motivo devono essere testati scientificamente;
- la conoscenza dei tassi di errore - è necessario comunicare al giudice l'ipotesi di errore calcolata;
- la sensitività – è necessario conoscere l'incidenza di errori del tipo falsi positivi;
- la specificità - è necessario conoscere l'incidenza di errori del tipo falsi negativi;
- la sottoposizione al controllo della comunità scientifica consentendo un controllo da parte degli esperti di settore tramite la pubblicazione delle tesi scientifiche nelle riviste specializzate.
- l'accettabilità – la procedura e i risultati devono essere condivisi dalla comunità scientifica;
- la generalizzabilità – i risultati devono essere applicabili anche a casi simili;
- la credibilità – la procedura e i risultati delle ricerche devono essere affidabili;
- l'affidabilità – deve essere deliberata la stabilità della misura in tempi diversi;

- la validità – si deve conoscere il grado di precisione con cui lo strumento effettua la misurazione e i risultati devono rispecchiare lo stato delle cose;
- la validità incrementale – si deve conoscere di quanto la validità della valutazione aumenti alla luce dell'aggiunta di nuove informazioni a quelle precedentemente. [26]

[26]https://www.studiolegaledelalla.it/prova-scientifica-daubert-revisione/

Capitolo 1 – Revisione della letteratura

L'open source nella digital forensics

L'uso di Software open source apporta all'analisi forense una serie di risvolti positivi che rendono più salda la presentazione dell'analisi della prova. La disponibilità del codice sorgente permette di prevenire contestazioni attinenti al funzionamento del software e la presenza numerosa di una forte community fa sì che difficilmente vengano a mancare correzioni di bug, aggiornamenti o adattamenti del software a nuove necessità contingenti, in risposta ad alcuni dei già citati criteri Daubert.
La disponibilità del codice sorgente (e di una community di supporto) consente la ripetibilità e falsificabilità delle analisi effettuate, garantendo le medesime condizioni. Questo non è possibile nel momento in cui viene utilizzato software closed source in quanto non consentono l'accesso al codice e quindi l'analisi della logica di funzionamento e la verifica dei processi a cui i dati sono sottoposti.

Una larga fetta dei software utilizzati nell'informatica forense è formata da sistemi operativi Linux e da tool Linux. Linux è un sistema operativo creato nel 1991 da Linus Torvalds e distribuito sotto licenza GNU - General Public License.
Anche se è comunemente usato come nome per l'intero sistema operativo, Linux è solo il nome del kernel.
L'espressione distribuzione Linux invece si riferisce a un sistema

operativo completo costruito sopra il kernel Linux, che di solito include un programma di installazione e molte applicazioni preinstallate.

Quindi, ogni distribuzione Linux è considerata una declinazione diversa dello stesso sistema operativo Linux. Al mondo ad oggi esistono diverse centinaia di distribuzioni che si differenziano per scelte tecniche, per possibili domini applicativi o scopi principali di utilizzo.

Analisi distribuzioni Linux non specializzate nell'indagine forense

Nell'ambito delle distribuzioni Linux, alcune distribuzioni vengono utilizzate per effettuare analisi forensi anche se non specificatamente progettate per questo scopo; tra queste troviamo alcuni sistemi Linux progettati per l'ethical hacking e il penetration testing, forniti però di tool utili per l'analisi forense, quali:

- BackBox Linux
- Kali Linux
- Parrot
- BlackArch Linux
- Backtrack (dal 2013 Kali Linux)

BackBox Linux

BackBox è una distribuzione GNU/Linux sviluppato nel 2010 in Italia Raffaele Forte. È basata su Ubuntu orientata ai penetration test e alla

valutazione della sicurezza. Fornisce alcuni degli strumenti di sicurezza e analisi più comunemente conosciuti / utilizzati per l'analisi dei sistemi, di applicazioni, di reti.
Fornisce un ambiente desktop minimale i file di configurazione sono stati organizzati e ridotti allo stretto necessario per fornire una distribuzione GNU/Linux intuitiva e di facile utilizzo.
Ha integrata una modalità anonima che attraverso il proxy Tor protegge la privacy, l'indirizzo MAC e il nome dell'host vengono cambiati in modo casuale e questo rende il sistema non rintracciabile all'interno della LAN.
Una sua caratteristica è la RAM Wiping e in fase di installazione permette di crittografare completamente l'Hard Disk.

Kali

Kali Linux è una distribuzione GNU/Linux sviluppata nel 2013 da Offensive Security attraverso la riscrittura di BackTrack. È basata su Debian ed è conforme allo standard FHS (Filesystem Hierarchy Standard).
Kali è specificamente orientata al penetration testing professionale e al security auditing e ha al suo interno oltre 600 tools preinstallati.
Kali fornisce anche una "modalità forense" chiamata "Forensic mode live boot" ereditata da BackTrack Linux. L'avvio in modalità forense comporta le seguenti modifiche al normale funzionamento del sistema:

1. L'hard disk interno non viene usato e nessuna unità interna viene montata automaticamente.

Questo può essere verificato confrontando gli “hash" calcolati prima e dopo dell’utilizzo di modalità di Kali. Così si potrà verificare che nessuna modifica viene effettuata sull’unità.

2. È disabilitato anche il montaggio automatico dei supporti rimovibili quali chiavette USB e CD

Tutto ciò è progettato nell’ottica che niente dovrebbe accadere a nessun dispositivo senza l’azione diretta dell’utente.

Parrot Security

Parrot OS è una distribuzione GNU/Linux pubblicata per la prima volta nel 2013 da Lorenzo Faletra. È sviluppata per il pentesting, il reverse engineering e lo sviluppo software. È basata su Debian Testing.

Parrot per motivi di sicurezza presenta la caratteristica di non consentire accessi root diretti e fornisce un proprio sistema sandbox.

Può essere usato nell’ambito dell’informatica forense ma per consentire l’acquisizione forense in modo sicuro è necessario disabilitare la funzione di automount predefinita

BlackArch Linux

BlackArch è una distribuzione GNU/Linux sviluppata nel 2014 per pentester e ricercatori di sicurezza informatica. Deriva da ArchLinux e permette l’installazione di tools scaricandoli da un repository utente.

Backtrack

Backtrack è una distribuzione GNU/Linux basata su Ubuntu rilasciata il 26 maggio 2006 focalizzata sulla sicurezza e sul penetration testing.

È composta da molti programmi e tools diversi, specializzati in sicurezza, protezione e hacking.
BackTrack è uno dei distribution sytem Linux più famosi, come si evince dal numero di download che ha raggiunto più di quattro milioni di a partire da BackTrack Linux 4.0 pre-final.
Backtrack è stata una pietra miliare tra le distribuzioni linux focalizzate sulla sicurezza. È stata sviluppata a partire dalla fusione di due distribuzioni precedentemente concorrenti, WHAX e Auditor Security Collection
BackTrack fornisce un accesso a una vasta e raccolta di strumenti di sicurezza, che vanno dai port scanner ai cracker di password.
Alcuni strumenti consentono di esaminare la rete utilizzando sniffers e file integrity checkers. Altri sono utilizzati per analizzare e identificare obiettivi utilizzando security testing, vulnerability scanning, wireless scanning e identificatori di network port/service.
Strumenti offerti da Backtrack coprono anche il privilege escalation, maintaining access, reverse engineering, radio frequency identification (RFID) tools, stress testing, forensics, reporting tools, network services, and altri tools vari.
Il 13 marzo 2013 cessa il supporto a Backtrack Linux e, in seguito ad una ricostruzione completa della piattaforma, Backtrack diventa Kali Linux.

Analisi distribuzioni Linux specializzate nell'indagine forense

Nonostante sia possibile effettuare analisi forensi utilizzando, con alcune accortezze, le distribuzioni precedentemente presentate, sono

state sviluppate distribuzioni Linux specializzate in informatica forense e distribuite specificatamente per un'audience target interessata all'informatica forense.
Tra queste, le principali sono:

- Iritaly
- Helix-Knoppix
- DEFT
- CAINE
- Tsurugi

Di queste sopra elencate, solo delle ultime e due il progetto é ancora in corso. Iritaly, Helix-Knoppix e DEFT sono state abbandonate.

Iritaly

Iritaly (Incident Response Italy) è un progetto sviluppato a partire dal 2003 presso il Dipartimento di Tecnologie dell'Informazione dell'Università Statale di Milano, Polo Didattico e di Ricerca di Crema e viene fornito con licenza GNU. Lo scopo principale del progetto è di informare e sensibilizzare la comunità tecnico/scientifica italiana, le aziende piccole e grandi, gli attori privati e pubblici sui temi dell'Incident Response e della Digital Forensics.
Il progetto era suddiviso in due parti, la prima documentale e la seconda consiste in un cdrom bootable set chiamato IRItaly Live CD.
Nella parte documentale, vengono presentate le best practices per analizzare le macchine al fine di ripercorrere gli episodi di hacking prestando particolare attenzione al metodo di identificazione, conservazione ed eventuale utilizzo delle prove.

IRItaly CD è un sistema operativo live basato sulla distribuzione Linux Gentoo, la quale è stata scelta per la sua versatilità e facilità di aggiornamento. La maggior parte dei tool presenti sono eseguibili da linea di comando e da interfaccia grafica.

Per evitare il rischio di utilizzare file di sistema compromessi che danneggino i risultati, al suo interno vengono utilizzati binari statici trusted compilati direttamente dal CD.

Il CD include, tra gli altri, anche strumenti che permettono di effettuare analisi dei dischi, generarne un'immagine su cui poi effettuare l'analisi, effettuare un'analisi dei log presenti.

L'ultimo rilascio di IRItaly Live CD è stata la v.3, di cui esistono tre versioni:

- IRItaly Live CD Base: contiene una serie di tool utili per le procedure di imaging e di raccolta di informazioni dei sistemi, per l'acquisizione delle fonti di prova dai media o dalla rete, per la valutazione delle vulnerabilità e per la redazione dei report. Le altre due versioni possono essere considerate suoi derivati.
- IRItaly Live CD Media: estensione della Base Version e contiene un supplemento di strumenti per l'analisi di supporti digitali
- IRItaly Live CD Network: estensione della Base Version e contiene un supplemento di strumenti per l'analisi delle reti.

Il progetto risulta attualmente abbandonato.

Helix

Helix è una distribuzione Linux per l'analisi forense. L'ultima versione gratuita, Helix3, è basata su Ubuntu e questo permette di ottenere

stabilità e facilità di utilizzo. È orientata verso utenti esperti e amministratori di sistema.

Helix riunisce una serie di strumenti specializzati nell'indagine forense ed è presente in due modalità:

- La prima è quella di base, Helix viene fornito come CD Live che consente di essere utilizzato sulla macchina target.
- La seconda permette a Helix.exe di essere utilizzato come applicativo per l'analisi su sistemi Windows. Il suo funzionamento è stato testato su Windows 98SE, Windows NT4, Windows 2000, Windows XP, Windows 2003 e Windows Vista. In questo caso è opportuno notare che l'esecuzione dell'applicativo sul sistema ne modifica lo stato.

Il toolkit include l'analizzatore di rete, diversi strumenti antivirus, per recuperare password, eseguire backup, ripristinare partizioni, esaminare i file binari, creare immagini live del sistema.

Helix ha la capacità di acquisire un'immagine live del sistema ed è in grado di inviarlo a un hard disk collegato, interno o esterno, e di inviarlo attraverso una rete

Helix presenta anche un'area chiamata " Incident Response " presenta un collegamento a diversi strumenti forensi e permette di calcolare i valori hash MD5 di qualsiasi file attraverso un file browser. Permette anche all'utente di catturare schermate per registrare le attività svolte.

Il progetto risulta attualmente abbandonato.

DEFT

Deft è una distribuzione GNU/Linux che include DART (Digital Advanced Response Toolkit), una suite dedicata alle attività di digital forensics intelligence sviluppata nel 2005 nel corso di Informatica Forense della facoltà di Giurisprudenza dell'Università degli Studi di Bologna.

In DEFT sono implementate alcune caratteristiche che riducono al minimo il rischio di alterare il dato sottoposto ad analisi. Alcune di queste sono:

1. All'avvio, il sistema non utilizza le partizioni di swap presenti nel sistema sottoposto ad analisi;
2. Non vi sono automatismi di mount delle memorie di massa all'avvio del sistema;
3. Non vi sono automatismi di alcun tipo durante l'attività di analisi delle evidenze;
4. Tutti i software di acquisizione di memorie di massa e di traffico su rete IP non alterano l'integrità del dato sottoposto ad acquisizione

Il progetto DEFT non è aggiornato da tempo, l'ultima versione DEFT Zero risale al febbraio 2017.

Il progetto risulta attualmente abbandonato.

CAINE

CAINE (Computer Aided Investigative Environment) è una distribuzione GNU-Linux italiana per l'analisi forense basata su Ubuntu. Il progetto è stato avviato nel 2008 da Giancarlo Giustini,

nell'ambito di un progetto del Centro interdipartimentale di ricerca forense digitale per la sicurezza (CRIS) dell'Università di Modena e Reggio Emilia, dalla fine del 2009 il progetto è gestito da Nanni Bassetti. L'ultima versione CAINE 11.0 è stata rilasciata il 02/12/2019.
Come le altre distribuzioni specializzate, CAINE si concentra sulla digital forensics combinando diversi tool in un comune package. Si differenzia da altre distro del settore in quanto fornisce anche una serie di strumenti desktop di uso generale che, nonostante la sua specificità d'uso, permettono di utilizzarlo come un classico sistema Ubuntu. In questo modo viene a mancare la necessità di passare da un ambiente di uso generale a uno specificatamente forense.
CAINE non monta automaticamente nessun dispositivo per non alterare le analisi, ma ciò è possibile solo attraverso un programma specifico. Include molti tool specializzati in memory, database, mobile e network forensics.
È una distribuzione Linux live avviabile da un supporto rimovibile o installabile su un sistema per poter analizzare la macchina target
CAINE offre un ambiente completo di digital forensics integrando strumenti esistenti attraverso una interfaccia grafica. CAINE fornisce

- Caine interface: un'interfaccia grafica user friendly che riunisce una serie di strumenti per l'analisi forense
- Un generatore semi-automatico di report, che permette all'utente di ottenere un documento modificabile ed esportabile contenente un riepilogo delle attività svolte

- L'adesione al procedimento di indagine definita di recente dalla Legge 48/2008 (in recepimento della convenzione europea di Budapest del 2001).

CAINE è un progetto completamente aperto che rappresenta appieno lo spirito dell'Open Source

Tsurugi

Tsurugi è una distribuzione Linux per la digital forensics rilasciata nel 2018 dal Tsurugi Linux team guidato da Giovanni Rattaro.

Come anche per le precedenti distribuzioni forensi analizzate, anche per Tsurugi lo scopo è quello di fornire un ambiente di lavoro organizzato per attività di acquisizione forense e di analisi, con strumenti testati, aggiornati e preconfigurati in modo tale da non richiedere altre installazioni e con proprietà di write-blocking software.

È basato su Ubuntu e si basa su un modello di indagine suddiviso in sei fasi:

1. Identificazione
2. Preservazione
3. Raccolta
4. Valutazione
5. Analisi
6. Presentazione

È rilasciata in tre versioni:

- Tsurugi Linux [Lab] - Versione completa di tutti gli strumenti avviabile su pendrive o DVD ma installabile su PC o macchina

virtuale. È dedicata all'informatica forense (acquisizione e analisi), OSINT (OpenSource INTelligence) e analisi malware

- Tsurugi Acquire - Versione più leggera dedicata alla sola acquisizione forense. Fornisce gli strumenti di base necessari per avviare un PC e acquisire dispositivi di archiviazione di massa, viene installato un piccolo sottoinsieme di strumenti allo scopo di ridurre le dimensioni della ISO. Funziona solo in modalità live con kernel a 32 bit per garantire maggiore compatibilità. Grazie alla sua dimensione contenuta, è possibile inserire l'intera immagine nella RAM, consentendo all'utente di rimuovere il pendrive/DVD dopo l'avvio.
- BENTO - Versione portatile per analisi live, strutturata sotto forma di raccolta di tool per Windows, Mac e Linux progettato per la live forensics e l'incident response. È possibile aggiornare e aggiungere strumenti al suo interno.

Tutti i device sono connessi di default in modalità di sola lettura attraverso un write blocker posto a livello del kernel. Tsurugi Linux assicura la funzionalità di write blocking anche nella versione installata.

Capitolo 2 – Analisi e progettazione

Introduzione

Tutte le distribuzioni Linux fin qui analizzate sono sviluppate a partire da distribuzioni preesistenti quali Debian, Ubuntu e in un caso anche Arch.

Da ciò deriva l'idea di una distribuzione Linux che unisca l'affidabilità e gli strumenti per consentire un'analisi forense di supporti di memoria di massa ma che sia compatta, minimale e rapida, al momento non presente nel panorama forense internazionale.

Definizione del problema

Perimetrato il problema, il sistema che ci si propone di realizzare dovrà soddisfare i seguenti requisiti, necessari per i singoli casi d'uso individuabili in fase di acquisizione:

Requisiti

- Il sistema non monterà automaticamente nessun dispositivo per non alterare le analisi;
- Saranno presenti tool per l'analisi forense delle memorie di massa e memorie volatili;
- Sarà possibile l'analisi via rete;
- L'analisi potrà essere effettuata su sistemi operativi Windows, Linux, Mac Os;
- La distribuzione sarà live da avviare attraverso un supporto esterno;
- L'interfaccia sarà a riga di comando;

- L’utente potrà avere permessi di root.

Write blocker

Un elemento essenziale che dovrà essere necessariamente presente nella distribuzione Linux che si andrà ad analizzare è il Write Blocker. Il Write Blocker è uno strumento utilizzato nell’ambito di analisi forensi che permette la protezione dei dati presenti su supporti di memorizzazione oggetto dell’attività forense. Essi prevengono il rischio di modifiche sul disco oggetto dell’investigazione e quindi la possibilità di invalidare potenziali prove involontariamente, assicurandone la connessione in read-only.

L’utilizzo di Write blocker permette di garantire l’integrità dei dati attraverso successiva verifica mediante uno o più algoritmi di hash.

Esistono varie tipologie di write blocker:

- Firmware based: attraverso il BIOS riesce a bloccare qualsiasi tentativo di scrittura.;
- Hardware based: dispositivo elettronico inventato da Steve Bress e Mark Menz nei primi anni 2000 con brevetto US6813682B. Esso intercetta i comandi inviati al disco che potrebbero modificare i dati. Ne esistono in commercio vari tipi, i principali si classificano sulla base dell’interfaccia utilizzata per l’acquisizione (USB, SATA, IDE, SCSI) oppure differenziandoli sulla base dell’uso o meno di adattatori tra le varie interfaccia (Write blocker native, Write blocker tailgate).

- Software based: software di basso livello che impedisce qualsiasi scrittura. In questo caso interviene il Sistema Operativo e non il Bios.

Un'altra possibile classificazione può essere fatta sulla gestione dei comandi ricevuti, in quanto i write blocker hardware operano in base a:

  - Una white list: un write blocker opera sulla base di una white list quando blocca qualsiasi comando indirizzato al dispositivo che non incluso in una lista di comandi sicuri noti;
  - Una black list: un write blocker opera sulla base di una black list quando blocca solo i comandi inclusi in una lista di comandi noti non sicuri.

I write blocker software sono complessi da implementare: il semplice montaggio di un disco in read-only (`mount -o ro`) non garantisce l'integrità del disco in quanto il kernel può ancora scrivere sul disco. Per poter essere efficace, il write blocker deve essere implementato al di sotto del livello del filesystem e dei degli altri driver del dispositivo.

Casi d'uso

Si cercherà ora di approfondire le caratteristiche peculiari del sistema descrivendone i casi d'uso previsti in fase di progettazione. Questi riguarderanno i principali casi di acquisizione forensi.

Acquisizione PC/Mac acceso – Live Acquisition

L'acquisizione live viene condotta principalmente per raccogliere prove volatili. Queste includono informazioni sui processi e sui servizi in esecuzione sul computer, nonché la chiave crittografica se viene utilizzato un algoritmo di crittografia del disco rigido. Si possono anche acquisire eventuali attività di rete.

L'analisi live viene utilizzata quando, in fase di perquisizione o sequestro, si presume che delle prove possano essere presenti su di un pc già acceso. In questo caso bisogna valutare se si ritiene necessaria l'analisi forense live, in caso affermativo, quali dati bisogna raccogliere e garantirne l'integrità.

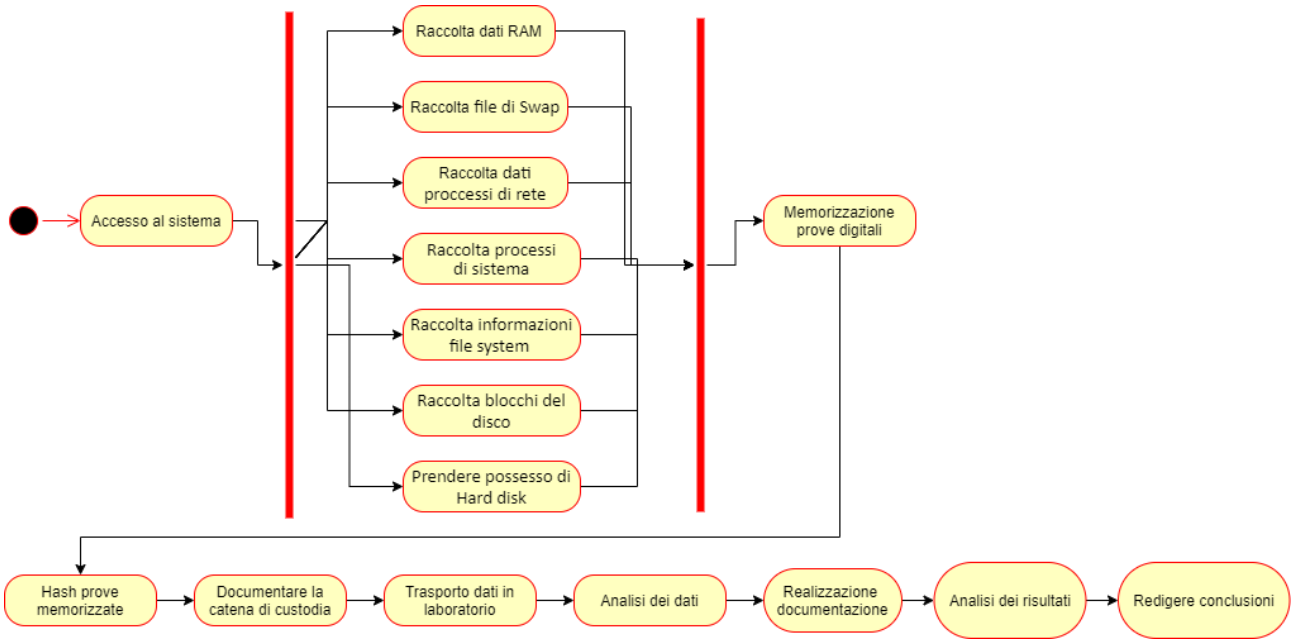
Quest'analisi presenta la caratteristica dell'irripetibilità di due tipi:

- Di natura tecnica, in quanto non esiste possibilità di realizzare analisi e acquisizione dati live senza modificare la memoria del sistema, quindi non sarà possibile eseguirla più di una volta;
- Di natura temporale: lo stato della macchina all'atto dell'attività è frutto del momento e la sua complessità è tale da non poter essere riprodotta.

Le informazioni che si possono acquisire in una memoria RAM sono:

1. Chiavi crittografiche;
2. Processi in esecuzione;
3. Comandi eseguiti da console;
4. Contenuto delle clipboard;
5. Informazioni di networking;
6. Contenuti decriptati;
7. Registri;

8. File di testo e immagini;
9. File cancellati;
10. Log dei browser web;
11. Chiavi di registro attività;
12. Password di account internet;
13. Malware;
14. Prove di attività non tipicamente memorizzate nell'hard disk.



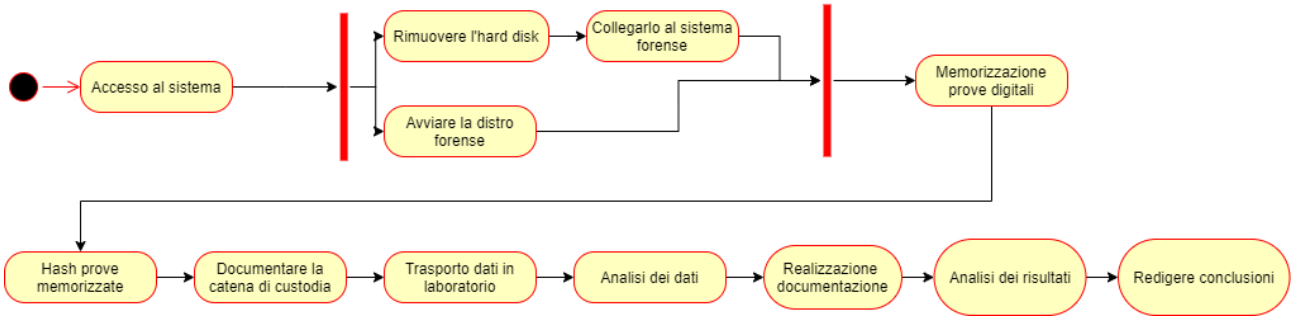
Acquisizione PC/Mac spento – Dead/Static Acquisition

L'acquisizione forense da pc spento viene utilizzata quando, in caso di perquisizione, si presume che delle prove possano essere presenti su pc già spenti o che si sceglie di spegnere. Nel secondo caso, lo spegnimento dovrà essere effettuato preferibilmente staccando il

computer dall'alimentazione elettrica (staccando la spina o rimuovendo la batteria).

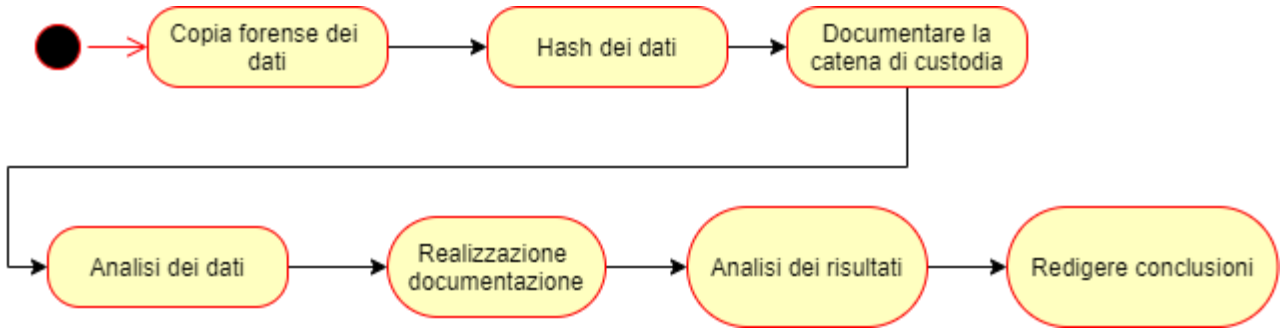
Una volta spento, l'investigatore potrà rimuovere il disco rigido dal sistema, collegarlo come device esterno e copiarne il contenuto prendendo le precauzioni necessarie affinché non sia effettuata nessuna modifica ai dati. Un'alternativa alla rimozione fisica delle memorie dal pc è l'avvio di distro forensi sul pc per poterne copiare i dati presenti.

Le successive analisi verranno effettuate sulle copie forensi ottenute



dal disco e non sul device stesso.

Acquisizione di supporti di memoria rimovibili



Acquisizione via rete

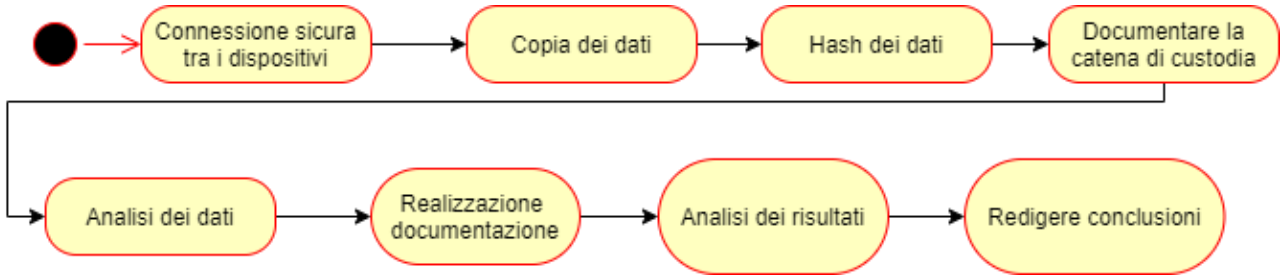
L'indagine forense svolta in modo tradizionale richiede che gli investigatori forensi debbano visitare la scena del crimine, dove poi

raccoglieranno tutte le prove pertinenti per poi analizzarle in laboratorio.

Durante l'acquisizione forense remota avviene la raccolta dei dati da dispositivi connessi in rete. Questa soluzione, come l'analisi live, consentirebbe la raccolta di dati volatili. Con l'acquisizione forense remota, le raccolte dati mirate possono essere gestite in modo efficiente.

L'acquisizione delle prove è ottenuta attraverso l'implementazione di un'architettura sicura e verificabile.

Deve essere in ogni caso garantita la catena di custodia, assicurando la non manipolazione dei dati trasmessi sulla rete



Capitolo 3 – Tiny Core

Dati i requisiti analizzati nel capitolo precedente, si è scelto di proseguire la realizzazione di una distro Linux forense minimale usando come base TinyCoreLinux, una light-weight Linux distribution.

Light-weight Linux distribution

Per Light-weight Linux distribution si intende una distribuzione Linux che richiede requisiti di memoria e/o velocità del processore inferiori rispetto alla media.

Questo obiettivo viene solitamente raggiunto tralasciando le caratteristiche ritenute superflue, evitando il sovraccarico del software.

Tiny Core Linux

TinyCore Linux è una distribuzione minimale del sistema operativo e degli strumenti di Linux. Sviluppata da Robert Shingledecker, la prima versione è stata rilasciata nel gennaio del 2009.

È un software gratuito e opensource ed è distribuito con la licenza GNU General Public License version2.

Sono presenti varie versioni della distribuzione che si differenziano in base alle loro caratteristiche. Le principali sono:

- Tiny Core ha dimensione 16 MB e dispone del sistema Core di base e di un'interfaccia grafica.
- Core ha dimensione 11 MB. È noto anche come "Micro Core Linux" ed è una variante più essenziale di Tiny Core senza desktop grafico.

- Core Plus ha dimensione 106 MB ed è un'immagine di installazione e non una distribuzione vera e propria. È composto da Tiny Core con funzionalità aggiuntive.

Tiny Core è in grado di avviarsi da pendrive o in modo definito "frugale" da hard drive.

Con "Frugal" gli sviluppatori intendono un metodo di installazione diverso rispetto a quello tradizionale, definito Scatter mode. Mentre con quest'ultimo tutti i file del sistema sono dislocati sul disco, con l'installazione frugal il sistema è sostanzialmente suddiviso in due file vmlinuz e core.gz, la cui posizione è specificata dal boot loader. Tutti i file e le estensioni degli utenti sono memorizzati al di fuori del sistema operativo di base.

A partire dalla versione 2.8.1, il core è progettato per funzionare principalmente nella RAM ma, in caso di installazione su hard disk, vengono mantenuti collegamenti simbolici con la RAM.

Tiny Core si carica nella RAM dallo storage, successivamente installa le applicazioni sullo storage, o installa le applicazioni nella RAM dallo storage.

Tiny Core non incoraggia ad eseguire una tradizionale installazione su disco rigido, questo però è ugualmente possibile, in quanto è progettato per funzionare da una copia RAM creata al momento dell'avvio.

Questo rappresenta un notevole vantaggio in quanto lo rende veloce, protegge i file di sistema dalle modifiche e garantisce un sistema incontaminato ad ogni riavvio.

Backup e Persistenza

Tiny Core Linux supporta il backup e il ripristino delle impostazioni personali e, di default, la persistenza delle directory */home* e */opt*.

Tiny Core include l'utility *filetool* per il salvataggio delle impostazioni e dei dati personali. È composto da un file di testo /opt/.filetool.lst che elenca le directory di cui eseguire il backup allo spegnimento e il backup al ripristino. Il file può essere modificato per poter effettuare il backup di altre directory. L'elenco contiene anche la voce */opt/.filetool.lst* che non deve mai essere rimossa.

Di default, il file filetool.lst include l'intera directory */home/tc*. Filetool scrive il file di backup mydata.tgz. È possibile utilizzare l'opzione di avvio *norestore* che permetterà di ignorare qualsiasi file di backup esistente.

Nella stessa directory di filetool.lst è presente il file .xfiletool.lst, il quale invece elenca tutte le directory da escludere. Le esclusioni prevalgono sulle inclusioni.

È necessario eseguire il backup ogni volta che un file viene modificato. Il comando per eseguirlo da terminale è *filetool.sh -b, i*l comando per spegnere il sistema è invece *sudo poweroff*.

Installazione app

Tiny Core Linux dispone di un proprio repository da cui è possibile scaricare e installare software. Esso varia in base alla versione e all'architettura del sistema operativo. Ad esempio, il repository per la versione corrente, la 11, con architettura x86 è presente al seguente indirizzo http://tinycorelinux.net/11.x/x86/tcz/.

Le applicazioni sono memorizzate localmente nella directory *tce*.

Da repository

Per installare app dal repository ufficiale, Tiny Core mette a disposizione il tool *tce-ab* e l'interfaccia diretta *tce-load*.

Tce-ab si presenta con la seguente interfaccia line-based con tre opzioni di ricerca.

```
$ tce-ab
tce-ab - Tiny Core Extension: Application Browser

S)earch P)rovides K)eywords Q)uit:
```

Tce-laod è un tool non interattivo usato dietro le quinte da *tce-ab*

```
$ tce-load -h
Usage: tce-load [-i -w -wi -wo -wil -ic -wic]{s} \
extensions
  -i   Loads local extension
  -w   Download extension only
  -wi  Download and install extension
  -wo  Download and create an ondemand item
  Adding -c to any -i option will force a one time \
        copy to file system
  Adding -l to any -i option indicates load only - \
        do not update onboot or ondemand
  Adding -s to any option will suppress OK message \
        used by apps GUI

Example usage:
 Load local extension:
   tce-load -i /mnt/hda1/tce/optional/nano.tcz
 Download into tce/optional directory, updates OnBoot
 and installs:
   tce-load -w -i nano.tcz
 Download only into tce/optional directory:
   tce-load -w nano.tcz
```

Questo tool può essere usato direttamente se si conosce già il nome dell'estensione da installare.

Ecco un confronto tra i package managers più comuni:

	apt (deb)	yum (rpm)	tce-load (tcz)
Install a package from the repo	apt-get install **pkg**	yum install **pkg**	tce-load -wi **pkg**
Install from a local file	dpkg -i **pkg**	yum localinstall **pkg**	tce-load -i **pkg**
Search	apt-cache search **pattern**	yum search **pattern**	tce-ab
List installed packages	dpkg -l	rpm -qa	ls /usr/local/ tce.installed

Di default, il sistema carica tutte le estensioni nel file onboot.lst presente nella directory */mnt/sda1/tce*. Questo file contiene una lista di tutte le estensioni che devono essere caricate nel boot.

Il file *.tcz* contiene l'albero diretto del file system che si può trovare quando si installa un'applicazione o una libreria. Ad esempio:

```
usr/local/bin/lxterminal
usr/local/share/applications/lxterminal.desktop
usr/local/share/lxterminal/lxterminal-preferences.ui
usr/local/share/lxterminal/lxterminal.conf
usr/local/share/pixmaps/lxterminal.png
```

Esterne

In TinyCore è possibile installare software non presente nella directory scaricando il codice sorgente sulla macchina e successivamente compilandolo con i classici comandi linux *./configure, make, make install.*

In questo caso, per rendere il software installato persistente sulla macchina, è necessario creare un'estensione .tcz ad hoc contenente i

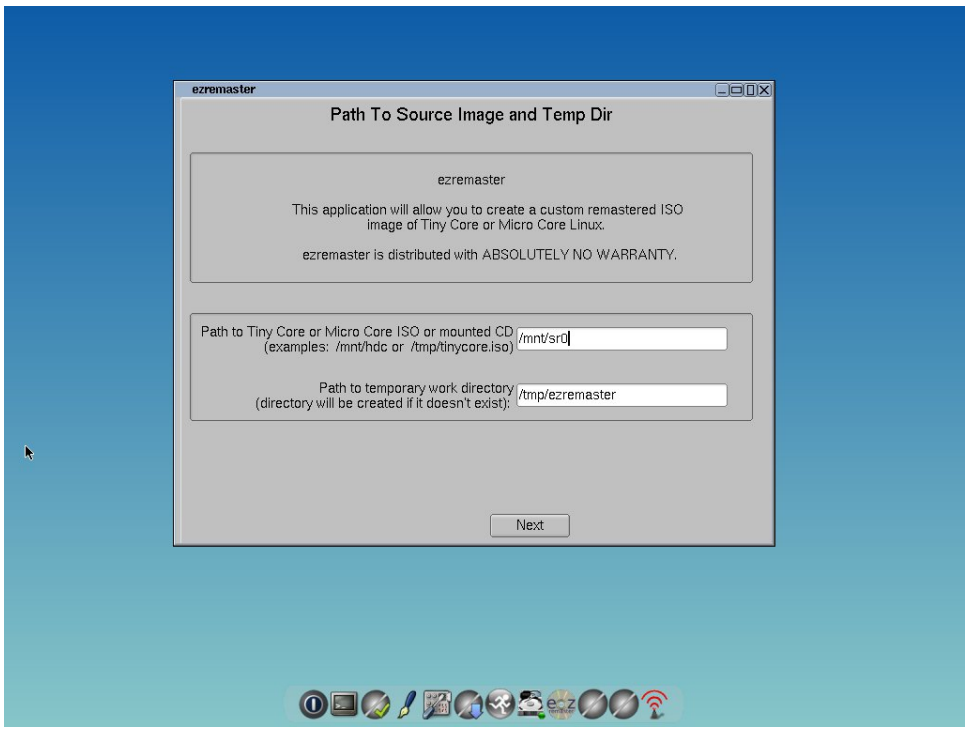
file binari. Dopo questo passaggio, sarà possibile re-installare il software con il comando *tce-load –i* [nomeapp.tcz].

Remastering e creazione ISO

Il Remastering è il processo di editing dell'immagine di sistema. Permette la produzione di una nuova immagine ISO.

È possibile effettuare il remastering da terminale o attraverso un'utility con interfaccia presente installabile in TinyCore chiamata EZRemaster. EZRemaster è un'applicazione GUI che semplifica la rimasterizzazione di TinyCore (Tiny con interfaccia grafica) o di Core (Tiny senza interfaccia grafica). Essendo EZRemaster un'app distribuita solo con interfaccia grafica, è necessario effettuare il remastering da una distro Tiny Core o Core Plus con GUI.

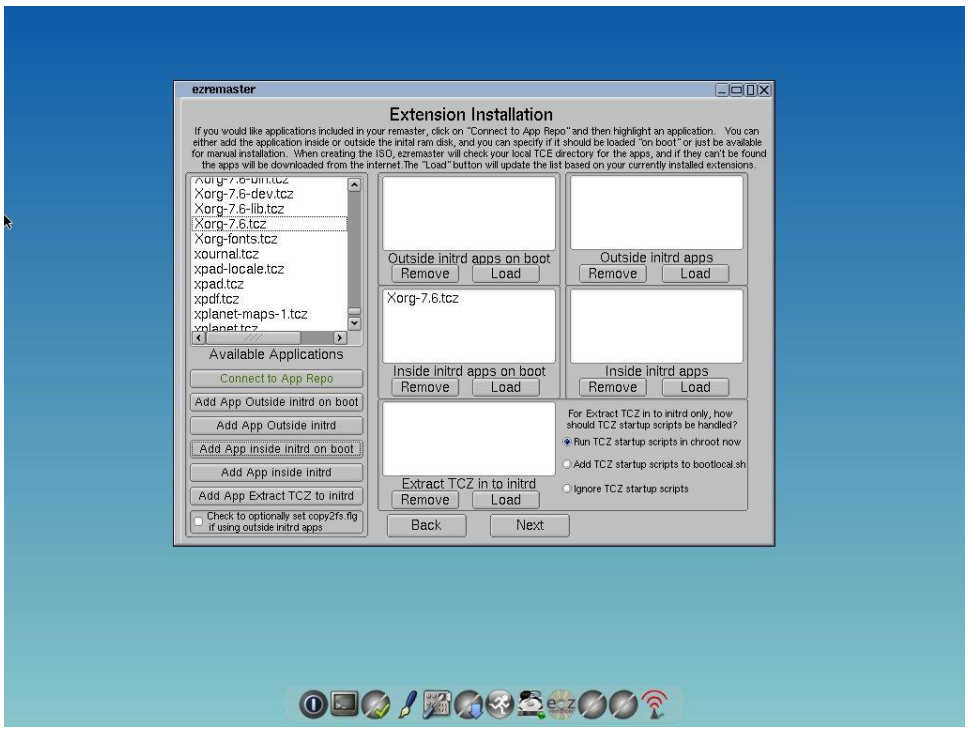
EZRemaster permette di rimasterizzare una ISO partendo da quello già montata nel sistema o da un file ISO diverso.



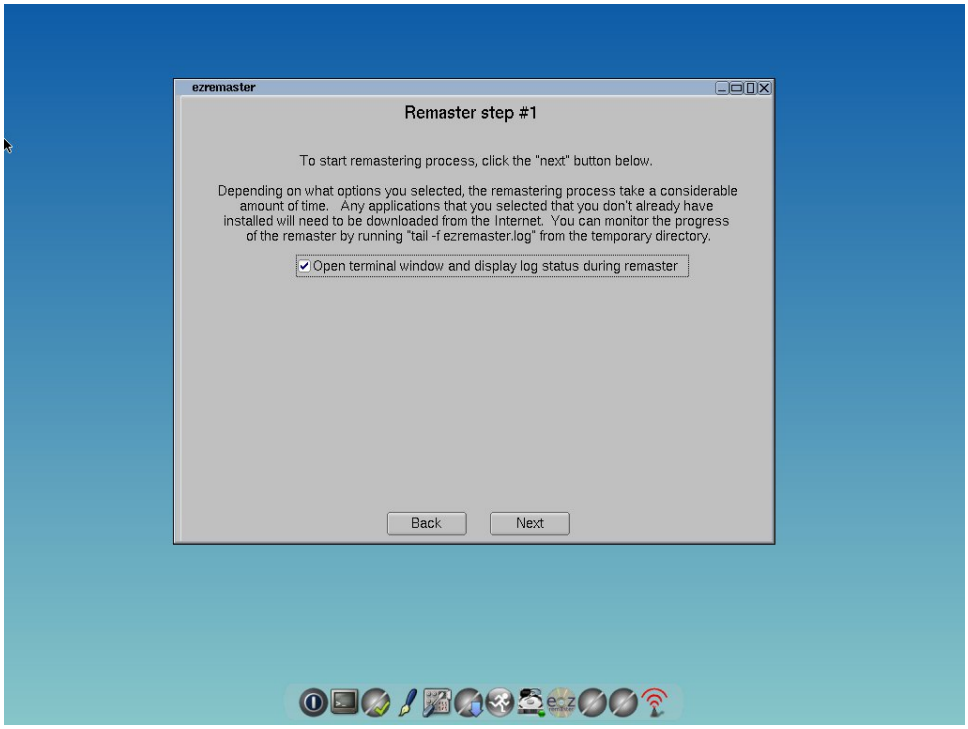
Successivamente alla scelta dell'ISO da remasterizzare, EZRemaster permette di scegliere se inserire o meno il path del file mydata.tgz.

Successivamente si potranno inserire le estensioni che si intendono inserire nel nuovo file ISO.

Per inserire software esterno è necessario che esso sia stato precedentemente trasformato in un file .tcz. In questo modo, comparirà tra i software installati e verrà mantenuto in fase di remastering.



Una volta terminati tutti i passaggi, potrà avviarsi il remastering. La nuova ISO così creata verrà salvata nella directory */tmp/ezremaster*.



L'utente di default è *tc*.

Capitolo 4 – Implementazione Tiny Core Forensic edition

Partendo dalle analisi svolte nei capitoli precedenti, si andrà ora a presentare la realizzazione del sistema operativo Linux forense minimale realizzato come progetto di tesi.

Write Blocker

Per garantire l'integrità dei dati e dei device che verranno analizzati utilizzando Tiny Core Forensic edition si è ritenuto indispensabile inserire un Write Blocker all'interno del sistema.

Al momento non esistono modalità universali per montare un file system in Linux. Montare un file system con il parametro `ro` con il comando `mount -o ro /dev/sda1 /mnt/sda1/` non garantisce che un driver del kernel non possa scriva sul dispositivo a blocchi connesso.

Citando la pagina https://man7.org/linux/man-pages/man8/mount.8.html contenente le specifiche del comando mount:

`-r, --read-only`

`Mount the filesystem read-only. A synonym is` **`-o ro`**`. Note that, depending on the filesystem type, state and kernel behavior, the system may still write to the device. For example, ext3 and ext4 will replay the journal if the filesystem is dirty. To prevent this kind of write access, you may want to mount an ext3 or ext4 filesystem with the` **`ro,noload`** `mount`

`options or set the block device itself to read-only mode, see the blockdev(8) command.`

Una modalità per poter montare un file system in sola lettura è contrassegnare un dispositivo a blocchi corrispondente come di sola lettura, prima di eseguire il comando `mount`, usando `blockdev`. Ad esempio con il comando `blockdev --setro / dev / sda1.`
In Tiny Core forensic edition si è proceduto ad implementare un Write blocker software. Si è scelto di configurare appositamente una regola Udev utilizzando il comando linux blockdev in modo da contrassegnare un dispositivo a blocchi come di sola lettura e aggiungere controlli che garantiscano la sola lettura dei devices. Questo permetterà di bloccare tutte le richieste di scrittura provenienti dai driver del kernel che potrebbero ignorare il montaggio in read-only.

Udev

Udev è il device manager del kernel linux. Il suo compito principale è quello di amministrare dinamicamente i dispositivi a blocchi per ogni periferica rilevata dal sistema.
I file contenenti le regole udev scritte dall'amministratore sono presenti nella directory `/etc/udev/rules.d/` mentre quelle fornita dall'installazione dei vari pacchetti si trovano della directory `/lib/udev/rules.d/`, hanno estensione .rules.
I file in `/etc/udev/rules.d/` vengono analizzati in ordine lessicale.
Un device può essere associato a più di una regola.

Sintassi

Ogni regola è costituita da una serie di coppie chiave-valore, separate da virgole. Le *match key* sono condizioni utilizzate per identificare il dispositivo su cui agisce la regola. Quando tutte le *match key* in una regola corrispondono al dispositivo gestito, la regola viene applicata e vengono richiamate le azioni delle *assignment key*. Ogni regola dovrebbe essere composta da almeno una match key e almeno un *assignment key*.

Per esempio:

```
KERNEL == "usb", NAME = "my_disk"
```

Include una match key (KERNEL) correlata al suo valore tramite l'operatore di uguaglianza == e una assignnmenet key (NAME) correlata al suo valore tramite l'operatore di assegnazione =. Udev fornisce diverse match key e assignment key che possono essere utilizzate per scrivere le regole.

Se si intende scrivere una regola per gestire più dispositivi è necessario, in certi casi, utilizzare gli *string substitution operator* questi operatori possono essere inseriti in qualsiasi assegnazione eseguita dalle regole e verranno valutati da udev al momento dell'esecuzione.

Gli *string substitution operator* piú comuni sono *%k* e *%n. %k* valuta il nome del kernel per il dispositivo, %n restituisce il numero di kernel per il dispositivo.

Per esempio:

```
KERNEL == "usb", NAME = "input /%k"
```

Assicurerà che il nodo del dispositivo usb appaia nella directory `/dev/input`.

Le regole udev possono eseguire programmi al verificarsi delle *match keys*. Un esempio di questo comando (RUN) è il seguente:

```
KERNEL=="usb", RUN+="/usr/bin/my_program"
```

Nel momento in cui viene eseguito `/usr/bin/my_program`, possono essere usate come variabili varie parti del sistema udev, inclusi i valori chiave come SUBSYSTEM. La variabile d'ambiente ACTION viene invece usata per rivelare se il dispositivo viene connesso, disconnesso o modificato. (add – remove - change).

Udev fornisce anche una key ENV per le variabili d'ambiente che possono essere utilizzate sia per il match che per l'assignment.

Nel caso dell'assignment, è possibile impostare variabili d'ambiente con cui è possibile effettuare confronti successivamente.

```
KERNEL=="fd0", SYMLINK+="floppy",
ENV{some_var}="value"
```

Nel caso di match, ci si può assicurare che le regole vengano eseguite solo in base al valore di una variabile di ambiente.

```
KERNEL=="fd0", ENV{an_env_var}=="yes",
SYMLINK+="floppy"
```

Blockdev

Blockdev è un comando Linux scritto da Andries E. Brouwer e riscritto da Karel Zak e permette di chiamare il block device ioctls dalla linea di comando. Fa parte del pacchetto util-linux.

Citando la pagina https://man7.org/linux/man-pages/man8/blockdev.8.html, i comandi che lo compongono sono:

`--flushbufs`

`Flush buffers.`

--getalignoff
Get alignment offset.

--getbsz
Print the blocksize in bytes. This size does not describe device topology. It's the size used internally by ithe kernel and it may be modified (for example) by filesystem driver on mount.

--getdiscardzeroes
Get discard zeroes support status.

--getfra
Get filesystem readahead in 512-byte sectors.

--getiomin
Get minimum I/O size.

--getioopt
Get optimal I/O size.

--getmaxsect
Get max sectors per request

--getpbsz
Get physical block (sector) size.

--getra
Print readahead (in 512-byte sectors).

--getro
Get read-only. Print 1 if the device is read-only, 0 otherwise.

--getsize64
Print device size in bytes.

--getsize
Print device size (32-bit!) in sectors. Deprecated in favor of
the **--getsz** option.

--getss
Print logical sector size in bytes – usually 512.

--getsz
Get size in 512-byte sectors.

--rereadpt
Reread partition table

--setbsz *bytes*
Set blocksize. Note that the block size is specific to the current file descriptor opening the block device, so the change of block size only persists for as long as **blockdev** has the device open, and is lost once **blockdev** exits.

--setfra *sectors*
Set filesystem readahead (same as **--setra** on 2.6 kernels).

--setra *sectors*
Set readahead (in 512-byte sectors).

--setro
Set read-only. The currently active access to the device may not be affected by the change. For example, a filesystem already mounted in read-write mode will not be affected. The change applies after remount.

--setrw
Set read-write.

Write Blocker Implementato

Dietro suggerimento di Giovanni 'sug4r' Rattaro, team leader dello staff sviluppatore della distro Linux Tsurugi, il file .rules che compone il write blocker in Tiny Core forensic edition è stato implementato a

partire dal progetto https://github.com/msuhanov/Linux-write-blocker di Maxim Suhanov.

Il file forensic.rule finale è stato così composto:

```
ACTION=="add", SUBSYSTEM=="block", KERNEL!="loop*",
RUN+="/usr/local/sbin/blockdev --setro /dev/%k"
ACTION!="remove",              SUBSYSTEM=="block",
KERNEL!="loop*", RUN+="/usr/local/sbin/blockdev --
setro /dev/%k"
ACTION=="change",              SUBSYSTEM=="block",
KERNEL!="loop*",                ENV{DISK_RO}=="0",
RUN+="/usr/local/sbin/blockdev --setro /dev/$name"
```

La regola udev forensic.rules eseguirà il comando `blockdev –setro`, impostando i dispositivi in modalità read-only ogni qual volta un dispositivo diverso da *loop* verrà connesso o nel momento in cui avverrà qualche modifica.

Per visualizzare tutti i dispositivi connessi e i loro attributi è necessario usare il comando `blockdev –-report`. Per impostare un disco in modalitá read-write sarà necessario utilizzare il comando `blockdev –-setrw /dev/[name]`.

Il sistema operativo Tiny Core, di default, non monta in automatico nessun device. Per poter leggere i dati contenuti in supporti di memoria e poterli analizzare, è necessario effettuare il mount della memoria con il comando: `mount ro /dev/[name] /mnt/[name]`.

Per poter invece montare in scrittura un qualsiasi dispositivo sarà necessario rimuovere il write blocker con il comando `blockdev –-`

`setrw /dev/[name]` e successivamente eseguire il comando `mount rw /dev/[name] /mnt/[name].`

App installate

Partendo dalle necessità espresse attraverso i casi d'uso, si è ritenuto necessaria l'implementazione nella distribuzione Tiny Core forensic edition di una serie di software che potessero supportare al meglio l'acquisizione e l'analisi forense di memorie.

- Foremost
- DdRescue
- Nmap
- Rhash
- Sleuthkit
- Volatility3
- Netcat
- Ewfacquire

Vengono ora analizzati i tool selezionati.

Interni

Sono stati inseriti nella categoria "Interni" quei tool già presenti nel repository di Tiny Core e che quindi non hanno richiesto procedure personalizzate per poter essere installati e mantenuti nel file ISO della distribuzione finale

Foremost

Foremost è un tool forense open source realizzato per la piattaforma Linux. Foremost permette agli analisti forensi di recuperare automaticamente file o file parziali da un'immagine o da un supporto di memorizzazione.

Foremost basa il suo funzionamento sull'header del file e sui tipi di footer specificati nel file di configurazione definito dall'utente.

Prima carica in memoria un segmento alla volta del supporto o dell'immagine in questione (dimensione di default di 10 mb); n seguito, usando un processo noto come file carving (un processo di riassemblaggio dei file partendo da alcuni suoi frammenti e in assenza di metadati del file system), ricerca nella memoria un tipo di file header che corrisponda ad uno tra quelli presenti nel file di configurazione di Foremost. Quando viene trovata una corrispondenza, scrive l'intestazione e i dati che la seguono in un file, interrompendosi quando viene trovato un piè di pagina o finché non viene raggiunto il limite di dimensione del file.

Riportando quanto scritto nella pagina https://linux.die.net/man/1/foremost, la sua sintassi è:

foremost[-h][**-V**][**-d**][**-vqwQT**][**-b**<blocksize>][**-o**<dir>] [**-t**<type>] [**-s**<num>][**-i**<file>]

Utilizzando il parametro –t, permette di recuperare i file da un'immagine disco sulla base di tipi di file specificati dall'utente, che possono essere:

- **jpg**
- **gif**
- **png**
- **bmp**
- **avi**
- **exe**
- **mpg**
- **wav**
- **riff**
- **wmv**
- **mov**
- **pdf**
- **ole**
- **doc**
- **zip**
- **rar**
- **htm**
- **cpp**
- **all**

Il file di configurazione è utilizzato per controllare quali tipi di file l'utente intende ricercare. Il file di configurazione foremost.conf è incluso di default. Per ogni tipo di file, esso descrive l'estensione, le caratteristiche dell'header e del footer, la dimensione massima del file.

Dd

Dd sta per Data Dump ed è un comando dei sistemi operativi Unix e Unix-like che copia dei dati in blocchi. Può effettuare opzionalmente conversioni.

Permette di clonare bit a bit un disco nella destinazione prescelta, a prescindere dal tipo di filesystem o di sistema operativo. Permette di clonare partizioni o interi hard disk.

L'immagine raw ottenuta può essere letta ed analizzata dalla maggior parte degli strumenti software comunemente utilizzati. Dd può anche leggere e convertire il clone effettuato.

L'acquisizione del clone con dd permette di ottenere una copia forense del disco su cui è possibile effettuare le operazioni di analisi senza così manomettere e modificare la prova originale.

DdRescue

GNU ddrescue è un tool Linux di data recovery. Permette di copiare di dati da un file o un supporto di memorizzazione ad un altro. Esso è in grado di recuperare le porzioni "buone" dei file nel caso riscontri errori di lettura.

Nonostante possa così sembrare dal nome, ddrescue non è un derivato di dd. La sostanziale differenza è che ddrescue utilizza un algoritmo per copiare i dati dalle unità guaste causando loro il minor danno aggiuntivo possibile.

La struttura dei comandi ddrescue è `ddrescue [options] infile outfile [logfile]`.

Nmap

Nmap («Network Mapper») è uno strumento open-source per la network exploration e l'auditing. È stato progettato per scansionare rapidamente reti di grandi dimensioni, ma è indicato anche per l'utilizzo verso singoli host. Nmap usa pacchetti IP "raw" (grezzi, non formattati) in varie modalità per determinare quali host sono disponibili su una rete, che servizi (nome dell'applicazione e versione) vengono offerti da questi host, che sistema operativo (e che versione del sistema operativo) è in esecuzione e che tipo di firewall e packet filters sono usati.

L'output di Nmap contiene un elenco degli obiettivi scansionati, con informazioni supplementari per ognuno a seconda delle opzioni usate.

Esterni

Sono stati inseriti nella categoria “Esterni” quei tool non presenti nel repository di Tiny Core e che hanno richiesto procedure personalizzate per poter essere installati e resi persistenti nel file ISO della distribuzione finale.

Netcat

L’utility netcat (nc) legge e scrive dati attraverso le connessioni di rete, utilizzando il protocollo UDP o TCP.

È uno strumento di debug ed esplorazione di rete che può creare quasi tutti i tipi di connessione di si necessita.

Alcune delle principali caratteristiche di netcat sono:

- Connessioni in uscita o in entrata, TCP o UDP, da o verso qualsiasi porta
- Checking DNS completo forward/reverse, opportuni warning.
- Possibilità di utilizzare qualsiasi local source port
- Possibilità di utilizzare qualsiasi locally-configured network source address
- Funzionalità di port-scanning integrato, con randomizer
- Capacità di source-routing incorporato
- Può leggere command line arguments dallo standard input
- Dump esadecimale della trasmissione e dati ricevuti
- Possibilità di consentire a un altro programma di stabilire connessioni

La sintassi dei comandi netcat è la seguente:

```
nc [-46DdhklnrStUuvzC] [-i interval] [-p source_port] [-s source_ip_address] [-T ToS] [-w timeout] [-X proxy_protocol] [ -x proxy_address[ :port]] [hostname] [port[s]]
```

Rhash

RHash (Recursive Hasher) è un'utility per il calcolo e la verifica dei valori hash dei file.

Supporta CRC32, MD4, MD5, SHA1, SHA256, SHA512, Tiger, DC ++ TTH, BitTorrent BTIH, ED2K, AICH, GOST R 34.11-94, RIPEMD-160, HAS-160, EDON-R 256/512, Whirlpool e Snefru -128/256 algoritmi.

Il calcolo dell'hash viene utilizzato per garantire e verificare e garantire l'integrità di dati.

RHash possiede le seguenti caratteristiche:

- Può creare e verificare Magnet links e eDonkey ed2k:// links.
- L'output può essere in un formato predefinito (SFV, BSD) o scelto dall'utente.
- Capacità di elaborare le directory in modo ricorsivo.
- Aggiornamento di file hash esistenti.
- Calcola diverse somme hash in un unico passaggio.
- Portabilità: il programma funziona allo stesso modo sotto Linux, * BSD o Windows.
- RHash è scritto in C puro, è perciò di piccole dimensioni ed è distribuito con licenza Open Source

La sintassi dei comandi di rhash è la seguente:

```
rhash [ option ]... [ file ]...
```

Sleuthkit

Sleuthkit è un toolkit forense opensouce sviluppato per analizzare filesystem e dischi utilizzando un approccio non intrusivo.

Lo Sleuth Kit consente di analizzare un'immagine del disco o del file system creata attraverso "dd" o applicazioni analoghe che permettano di creare un'immagine "raw" in un formato non proprietario.

Il formato dei dati della partizione esaminata (NTFS, FAT, UFS, EXT2 o EXT3) non dipende dal sistema operativo della macchina su cui viene eseguito lo Sleuth Kit. Sleuth Kit può essere eseguito da un sistema UNIX anche durante un incident response, mostrando i file che

potrebbero essere nascosti eseguendo i rootkit senza modificare i file oggetto dell'indagine.

Sleuth Kit permette di:

- Analizzare la cancellazione e allocazione dei nomi dei file;
- Visualizzare i dettagli e del contenuto di tutti gli NTFS;
- Visualizzare i dettagli del file system e della struttura dei metadati;
- Creare file di activity timelines che possono essere importati in fogli di calcolo per la creazione di grafici e report;
- Visualizzare file hash in un database hash;
- Effettuare un raggruppamento sulla base dei tipi dei file.

Sleuth Kit è stato scritto in C e Perl.

I tools presenti in Sleuth Kit si suddividono in:

- File system tool: permettono l'elaborazione generale dei dati provenienti dai file system, incluso il layout e le strutture di allocazione;
- File Name tool: permettono l'elaborazione della file name structure, tipicamente presente nelle parent directory;
- Metadata tool: effettuano l'elaborazione della metadata structure, memorizzando i dettagli dei file;
- Data Unit tool: permettono l'elaborazione delle unità di dati in cui viene salvato il contenuto del file (cluster FAT e NTFS, e blocchi/frammenti UFS, EXT2 e EXT3);
- Media management tool: questi strumenti ricevono un'immagine del disco come input e analizzano la management structure su cui sono organizzati.

Volatility3

Volatility è un framework utilizzato per l'estrazione di artefatti digitali da campioni di memoria volatile (RAM). Le tecniche di estrazione sono eseguite in modo indipendente rispetto al sistema in esame, ma offrono visibilità sullo stato di funzionamento del sistema. Nel 2019, la Volatility Foundation ha pubblicato una riscrittura completa del framework, Volatility 3, che è stato implementato nella distro Tiny Core Forensic Edition oggetto di questo progetto di tesi.

È completamente open source, rilasciato sotto la GNU General Public License6 e scritto in Python. Al momento Volatility comprende il supporto ufficiale per Microsoft Windows, Linux e Mac OS.

Data un'immagine di memoria, Volatility può estrarre processi in esecuzione, socket di rete aperti, mappe di memoria per ogni processo, o moduli del kernel.

Ewfacquire

Ewfacquire è un tool che permette di acquisire dati multimediali da un sorgente e archiviarli in formato EWF (Expert Witness Compression Format). È presente nella libreria Libewf.

Acquisisce i dati multimediali in un formato equivalente a EnCase e FTK imager, inclusi i metadati. In sistemi operativi Linux, FreeBSD, NetBSD, OpenBSD, MacOS-X / Darwin, ewfacquire supporta la lettura direttamente dai file del dispositivo.

Su altre piattaforme ewfacquire può convertire un'immagine raw (dd) nel formato EWF.

La sua sintassi è la seguente:

ewfacquire [**-A** codepage] [**-b** number_of_sectors] [**-B** number_of_bytes] [**-c** compression_values] [**-C** case_number] [**-d** digest_type] [**-D** description] [**-e** examiner_name] [**-E** evidence_number] [**-f** format] [**-g** number_of_sectors] [**-l** log_filename] [**-m** media_type] [**-M** media_flags] [**-N** notes] [**-o** offset] [**-p** process_buffer_size] [**-P** bytes_per_sector] [**-r** read_error_retries] [**-S** segment_file_size] [**-t** target] [**-T** toc_file] [**-2** secondary_target] [**-hqRsuvVwx**] source

Implementazione

Per l'installazione di software "esterno", non presente nel repository ufficiale di Tiny Core, è stato necessario scaricare il codice sorgente e successivamente compilarlo.

Questo è stato fatto per i software:

- Rhash: codice sorgente scaricato presso https://github.com/rhash/RHash
- Sleuthkit: codice sorgente scaricato presso https://github.com/sleuthkit/sleuthkit
- Volatility3: codice sorgente scaricato presso https://github.com/volatilityfoundation/volatility3

- Netcat: codice sorgente scaricato presso https://netix.dl.sourceforge.net/project/netcat/netcat/0.7.1/netcat-0.7.1.tar.gz
- Ewfacquire: https://github.com/libyal/libewf/

I software Rhash, Sleuthkit e Netcat sono stati installati attraverso i comandi:

```
./configure
make
make install
```

Il software Ewfacquire è stato installato attraverso i seguenti comandi:

```
./synclibs.sh
./autogen.sh
./configure
make -j8
sudo make install
```

Il software Volatility3 è avviabile direttamente senza installazione attraverso il comando:

```
python3 /home/tc/Volatility3
```

In seguito all'installazione, per i software Rhash, Sleuthkit e Nmap si è reso necessario un ulteriore passaggio per poterli rendere persistenti sulla ISO che si andrà a creare.

Come spiegato nel capitolo precedente, si è reso necessario creare un'estensione .tcz contenente i file binari dei software. In questo modo è stato possibile mantenere anche queste app in fase di remastering.

Per garantire la persistenza dell'utility framework Volatility è necessario invece includere nel file di testo /opt/.filetool.lst una stringa con la directory della cartella contenente i file che compongono Volatility.

Installazione app

Per creare un file con estensione .tcz a partire da file binari è necessario eseguire i seguenti passaggi.

1. Installazione del codice sorgente

```
./configure
make
make install
```

2.
```
cd /tmp
mksquashfs [destless] [myless].tcz
```

3.
```
cd /tmp
mv -v [myless].tcz
/etc/sysconfig/tcedir/optional
```

4.
```
cd /etc/sysconfig/tcedir
echo [myless].tcz >> onboot.lst
```

Ora si potrà procedere all'installazione con il comando `tce-load -i [myless].`

Creazione della Iso

La creazione della Iso è avvenuta attraverso il software EZremaster, descritto in precedenza.

Una volta che si è provveduto all'installazione dei vari software e si è provveduto ad implementare il write blocker è possibile procedere al remastering del sistema.

EZRemaster permette di rimasterizzare una ISO partendo da quello già montata nel sistema o da un file ISO diverso. Si è scelto di rimasterizzare l'iso Core-current.iso (http://tinycorelinux.net/11.x/x86/release/Core-current.iso).

Core è il sistema Tiny Core Linux di base che fornisce solo un'interfaccia a riga di comando (Tiny Core Linux),la più leggera, poiché è quella che si adatta al meglio agli scopi di questo progetto.

Le app con estensione .tcz che sono state incluse nella rimasterizzazione di Core.iso sono le seguenti:

- wget.tcz
- bash.tcz
- curl.tcz
- python.tcz
- python-dev.tcz
- python-pip.tcz
- python3.6.tcz
- python3.6-dev.tcz
- python3.6-pip-licenses.tcz
- curl-dev.tcz
- linux-5.4_api_headers.tcz
- vim.tcz
- nmap.tcz
- nmap-dbs.tcz

- wireshark.tcz
- wireshark-dev.tcz
- coreutils.tcz
- bash-dev.tcz
- python3.6-setuptools.tcz
- python-setuptools.tcz
- ddrescue.tcz
- foremost.tcz
- mynetcat.tcz
- myrhash.tcz
- mysleuthkit.tcz
- myewfacquire.tcz
- mylibewf.tcz
- util-linux.tcz
- openssh.tcz
- util-linux-dev.tcz
- ntfs-3g-dev.tcz
- udev-extra.tcz
- udev-keymap.tcz
- xz.tcz
- xz-locale.tcz

Al termine del processo di remastering, nella cartella `/tmp/ezremaster/` si è ottenuto il file finale ezremaster.iso, rinominato in seguito Tiny_Core_Forensic_Edition.iso

Questo file risultante è l'Iso finale funzionante della distribuzione Tiny Core Forensic Edition.

Capitolo 5 - Risultati

Upload iso

Il file iso risultante da questo progetto è stato caricato in Google Drive al seguente link: https://drive.google.com/file/d/1PIUsYC8Svlt82zyn9qumXZEDpMDGpIa6/view

Installazione su usb per live

Il sistema Tiny Core Forensic Edition potrà essere installato su una memoria USB in modo da ottenere un'unità flash USB avviabile.
A tal fine può essere, ad esempio, utilizzata l'utility Rufus https://rufus.ie/.

Dimensione

L'ISO ottenuta risulta avere dimensioni di 181,8 MB.

Kernel

Il kernel Linux utilizzato nella distro Core.iso che si è provveduto con questo progetto a rimasterizzare nella distribuzione Tiny Core Forensic Edition e la versione 5.4.3.

Sviluppi futuri

L'obiettivo di questo progetto di tesi è stato quelli di analizzare, progettare e realizzare una live distro Linux forense specializzata in acquisizione.
Dal momento del rilascio di questa prima versione stabile in poi, sarà necessario svolgere un lavoro di mantenimento della distribuzione realizzata.

Un possibile sviluppo futuro potrebbe essere l'implementazione di un'interfaccia grafica che non appesantisca eccessivamente la distribuzione e che nel contempo la renda maggiormente usabile da parte degli utenti.

Bibliografia

- L'informatica giuridica oggi – Antonio A. Martino
- Prova scientifica e processo penale. La sentenza Daubert ed i canoni per una corretta valutazione. La revisione del processo - Dott.ssa Alice Del Pero
- https://it.qwe.wiki/wiki/Forensic_science
- http://www.porcupine.org/forensics/tct.html
- https://dfrws.org/conferences/dfrws-usa-2001/
- Practical Forensic Imaging Securing Digital Evidence with Linux Tools - Bruce Nikkel
- Marco Mendola - Aspetti informatici delle prove biometriche: Il problema dei "Falsi positivi" (2013)
- Gli Strumenti Open Source dell'Investigatore Digitale - Vincenzo Giovanni Calabro
- http://www.digital-evidence.org/papers/opensrc_legal.pdf
- https://medium.com/@theinfovalley097/what-is-linux-an-overview-of-the-linux-operating-system-77bc7421c7e5
- Kali Linux – Assuring Security by Penetration Testing - Lee Allen Tedi Heriyanto Shakeel Ali
- Una distribuzione Linux con scopi pedagogici: analisi e progettazione Chiara Gandolfi
- https://www.backbox.org/
- https://wiki.backbox.org/
- Kali Linux – The BackTrack Successor
- https://tools.kali.org/

- https://www.kali.org/docs/general-use/kali-linux-forensics-mode/
- https://en.wikipedia.org/wiki/Parrot_OS
- https://parrotlinux.org/
- Guida a BlackArch Linux https://www.blackarch.org/
- http://distrowatch.com/
- BackTrack System: Security against Hacking Munir A. Ghanem
- https://www.backtrack-linux.org/
- http://www.swappa.it/wiki/Uni/IRItaly
- The State of the Art in Digital Forensics - DARIO FORTE
- https://web.archive.org/web/20130308110527if_/http://www.iritaly-livecd.org/Versione3.htm
- http://www.swappa.it/wiki/Uni/IRItaly
- https://www.dedoimedo.com/computers/helix.html
- https://web.archive.org/web/20080120014125/http://www.e-fense.com/helix/
- DEFT Digital Evidence & Forensics Toolkit – Manuale D'uso – Stefano Fratepietro, Sandro Rossetti, Paolo Dal Checco
- https://web.archive.org/web/20190115042536/http://www.deftlinux.net/about
- https://www.caine-live.net/
- Methodology for an Open Digital Forensics Model Based on CAINE Casimer Decusatis Aparicio Carranza, Alassane Ngaide, Sundas Zafar,
- https://web.archive.org/web/20190311233531/forensicswiki.org/wiki/CAINE_Live_CD

- https://medium.com/@shoaib629/is-tsurugi-linux-lab-a-sans-sift-killer-160d9b1d4ef1
- https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1569844675.pdf
- https://www.dalchecco.it/tsurugi-linux-distribuzione-forense/
- https://medium.com/@shoaib629/tsurugi-acquire-linux-distro-for-live-disk-acquisitions-3a749cc9399f
- https://tsurugi-linux.org/documentation.php
- Live Forensics - Fabio Fulgido, Gaetano Rocco, Mario Fiore Vitale
- Digital Forensics: acquisizione forense di prove con moderni strumenti digitali - Mimma Ruggiero
- Are hardware write blockers more reliable than software ones? - Maxim Suhanov
- Case-Based Reasoning In Live Forensics - Bruno Hoelz, Celia Ralha and Frederico Mesquita
- Digital Forensics Basics - A Practical Guide Using Windows OS - Nihad A. Hassan
- Formalizing Computer Forensics Process with UML - Chun Ruan and Ewa Huebner
- Live Forensic Acquisition as Alternative to Traditional Forensic Processes - Marthie Lessing1, Basie von Solms
- https://www2.deloitte.com/in/en/pages/finance/topics/forensic/remote-forensic.html
- https://en.wikipedia.org/wiki/Light-weight_Linux_distribution
- https://en.wikipedia.org/wiki/Tiny_Core_Linux

- http://tinycorelinux.net/corebook.pdf
- http://tinycorelinux.net/concepts.html
- http://tinycorelinux.net/intro.html
- https://www.linuxsecrets.com/tinycorelinux-wiki/
- https://github.com/msuhanov/Linux-write-blocker
- https://wiki.archlinux.org/index.php/Udev_(Italiano)
- http://www.reactivated.net/writing_udev_rules.htmlt
- https://web.archive.org/web/20120526150853/http://cisr.nps.edu/downloads/theses/05thesis_mikus.pdf
- https://www.drdobbs.com/the-foremost-open-source-forensic-tool/199101633
- https://en.wikipedia.org/wiki/Foremost_(software)
- https://linux.die.net/man/1/foremost
- https://wiki.archlinux.org/index.php/Disk_cloning_(Italiano)
- https://linux.die.net/man/1/dd
- https://www.gnu.org/software/ddrescue/
- https://www.gnu.org/software/ddrescue/manual/ddrescue_manual.html#Algorithm
- https://linux.die.net/man/1/ddrescue
- https://nmap.org/man/it/index.html#man-description
- http://netcat.sourceforge.net/
- https://nc110.sourceforge.io/
- https://linux.die.net/man/1/nc
- http://rhash.sourceforge.net/
- http://rhash.sourceforge.net/manpage.php

- https://dfir.science/2017/11/EWF-Tools-working-with-Expert-Witness-Files-in-Linux.html
- Computer Forensics with The Sleuth Kit and The Autopsy Forensic Browser Ricardo Kléber Martins Galvão
- https://github.com/sleuthkit/sleuthkit
- https://github.com/volatilityfoundation/volatility3
- Leveraging Forensic Tools for Virtual Machine Introspection - Brendan Dolan-Gavitt Bryan Payne Wenke Lee
- https://linux.die.net/man/1/ewfacquire
- http://manpages.ubuntu.com/manpages/xenial/man1/ewfacquire.1.html