

Appunti di Computer Forensics
Docente: Lorenzo Laurato

Matteo Rocco

Anno accademico 2023-2024



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Indice

1 Lezione 1	12
1.1 La Computer Forensics	12
1.2 Nascita della Computer Forensics	12
1.3 Panorama Giuridico	14
1.4 Metodologie della Computer Forensics	15
2 Lezione 2	16
2.1 Procedimento Penale	16
2.1.1 Fase Iniziale - Iscrizione della notizia di reato	16
2.1.2 Indagini Preliminari	16
2.1.3 Accertamento Tecnico	17
2.1.4 Misure Cautelari	17
2.1.5 Incidente Probatorio	17
2.1.6 Richiesta di Archiviazione	18
2.1.7 Rinvio a Giudizio	18
2.1.8 Udienza Preliminare	18
2.1.9 Citazione Diretta a Giudizio	18
2.1.10 Fase Dibattimentale	19
2.1.11 Sentenza	19
2.1.12 Impugnazione	19
2.1.13 Giudicato Penale	19
2.2 Procedimento Civile	21
2.2.1 Procedimento Ordinario	22
2.2.2 Procedimento con Ricorso	22
3 Lezione 3	24
3.1 Attori del Procedimento Penale	24
3.2 Struttura Organizzativa	24
3.3 Organizzazione della Procura	25
3.4 Il Pubblico Ministero	25
3.4.1 I poteri del Pubblico Ministero	25
3.5 La Polizia Giudiziaria	26
3.6 La Persona Offesa	26
3.7 Esposto, Denuncia e Querela, cosa sono ?	26
3.8 Indagato ed Imputato	27
3.9 Avvocato Difensore	27
3.10 Giudice delle Indagini Preliminari	27
3.11 Giudice dell'Udienza Preliminare	28
3.12 Giudice del Dibattimento	28
3.13 Il Computer Forensic nel Procedimento Penale	28
3.13.1 Le indagini preliminari	28
3.13.2 Il ruolo del C.F.	29
3.13.3 Accertamento Irripetibile (Art. 360 C.P.P.)	29
3.13.4 Perito	30

3.13.5	Verbale di consegna materiale	32
3.13.6	Richiesta proroga termini	33
3.13.7	Incarico di perizia	34
3.13.8	Verbale operazioni compiute	35
4	Lezione 4	36
4.1	Il Reato	36
4.2	Il Reato Informatico	36
4.3	Consiglio d'Europa del 1989	37
4.4	Legge 547/1993	37
4.4.1	Art. 392 C.P. - Esercizio arbitrario delle proprie ragione con violenza sulle cose	38
4.4.2	Art. 420 C.P. - Attentato ad impianti di pubblica utilità .	38
4.4.3	Art. 491-bis C.P. - Documenti Informatici	38
4.4.4	Art. 615-ter C.P. - Accesso abusivo ad un sistema informatico o telematico	39
4.4.5	Art. 615-quater C.P. - Detenzione e diffusione abusiva di codici di accesso a sistemi informatici o telematici . . .	39
4.4.6	Art. 615-quinquies C.P. - Diffusione di programmi diretti a danneggiare o interrompere un sistema informatico . .	39
4.4.7	Art. 616 C.P. - Violazione, sottrazione e soppressione di corrispondenza	40
4.4.8	Art. 617-quater C.P. - Intercettazione, impedimento o interruzione illecita di comunicazioni informatiche . .	40
4.4.9	Art. 617-quinquies C.P. - Installazione di apparecchiature atte a intercettare, impedire o interrompere comunicazioni informatiche o telematiche	40
4.4.10	Art. 617-sexies C.P. - falsificazione, alterazione o soppressione del contenuto di comunicazioni informatiche o telematiche	41
4.4.11	Art. 621 C.P. - Rivelazione del contenuto di documenti segreti	41
4.4.12	Art. 623-bis C.P. - Altre comunicazioni e conversazioni .	41
4.4.13	Art. 635-bis C.P. - Danneggiamento di sistemi informatici o telematici	41
4.4.14	Art. 640-ter C.P. - Frode informatica	42
4.5	Frode Informatica	42
4.6	Evoluzione Normativa - Legge n°48 del 18/03/2008	42
4.6.1	Convenzione sulla criminalità informatica	42
4.6.2	Legge 48/2008	43
4.7	Differenza tra Ispezione e Perquisizione	44
4.7.1	Art. 244 C.P.P. - Casi e forme delle ispezioni	44
4.7.2	Art. 247 C.P.P. - Casi e forme delle perquisizioni	44
4.7.3	Art. 259 C.P.P. - Custodia delle cose sequestrate	45
4.7.4	Art. 260 C.P.P. - Apposizione di sigilli alle cose sequestrate. Cose deperibili. Distruzione di cose sequestrate. . . .	45

4.8	Altri articoli	46
4.8.1	Art. 248 C.P.P. - Richiesta di consegna	46
4.8.2	Art. 254 C.P.P. - Sequestro di corrispondenza	46
4.8.3	Art. 254-bis C.P.P. - Sequestro di dati informatici presso fornitori di servizi informatici, telematici e di telecomunicazioni	47
4.8.4	Art. 256 C.P.P. - Dovere di esibizione e segreti	47
4.8.5	Art. 352 C.P.P. - Perquisizioni	47
4.8.6	Art. 353 C.P.P. - Acquisizione di plichi o di corrispondenza	47
4.8.7	Art. 354 C.P.P. - Accertamenti urgenti sui luoghi, sulle cose e sulle persone. Sequestro.	48
4.8.8	Art. 420 C.P. - Attentato a impianti di pubblica utilità	48
4.8.9	Art. 491-bis C.P. - Documenti Informatici	48
4.8.10	Art. 495-bis C.P. - Falsa dichiarazione o attestazione al certificatore di firma elettronica sull'identità o su qualità personali proprie o di altri	49
4.8.11	Art. 615-quinquies C.P. - Diffusione di apparecchiature, dispositivi o programmi informatici diretti a danneggiare o interrompere un sistema informatico o telematico	49
4.8.12	Art. 635-bis C.P. - Danneggiamento di informazioni, dati e programmi informatici	49
4.8.13	Art. 635-ter C.P. - Danneggiamento di informazioni, dati e programmi informatici utilizzato dallo stato o da altro ente pubblico o comunque di pubblica utilità	49
4.8.14	Art. 635-quater C.P. - Danneggiamento di sistemi informatici o telematici	50
4.8.15	Art. 635-quinquies C.P. - Danneggiamento di sistemi informatici o telematici di pubblica utilità	50
4.8.16	Art. 640-quinquies C.P. - Frode informatici del soggetto che presenta servizi di certificazione di firma elettronica	50
5	Lezione 5	51
5.1	Ricorda, cos'è la computer forensics?	51
5.2	Fasi del Trattamento - Identificazione e Raccolta.	51
5.2.1	Identificazione	51
5.2.2	La Preview	51
5.2.3	Cambiamento di stato del dispositivo	53
5.2.4	La Raccolta	53
5.2.5	Sequestro	53
5.2.6	Acquisizione Fisica	54
5.2.7	Strumenti per la Copia Forense	55
6	Lezione 6	56
6.1	Copia Forense	56
6.1.1	Hash	56
6.1.2	Accertamenti Ripetibili VS Irripetibili	57

6.1.3	File di Log	58
6.2	Comandi per eseguire la Copia Forense	59
6.2.1	Comando <i>DD</i>	59
6.2.2	Patch del comando <i>DD</i>	64
6.3	Calcolare l'Hash	66
6.3.1	Metodo n°1	66
6.3.2	Metodo n°2	67
7	Lezione 7	68
7.1	Disk Image	68
7.1.1	Supporti Ottici	68
7.1.2	Dischi Virtuali	68
7.1.3	Formato <i>DD/Raw</i>	69
7.1.4	Expert Witness Disk Image Format - (E.W.F.)	69
7.2	Software di Acquisizione	71
7.2.1	Guymager	71
7.2.2	FTK Imager	74
8	Lezione 8	85
8.1	Crittografia	85
8.2	Protocolli	86
8.2.1	Cifrario Simmetrico	86
8.2.2	Cifrario Asimmetrico	87
8.2.3	Firma Digitale	87
8.2.4	Funzioni di Hash	88
8.2.5	Funzione M.A.C.	89
8.3	Proprietà di Sicurezza	89
8.4	Funzioni di Hash - Proprietà	89
8.5	Funzioni di Hash - Costruzione	90
8.5.1	Modello Parallello	90
8.5.2	Modello Iterato	91
8.5.3	Modello Cascata	91
9	Lezione 9	92
9.1	Funzioni di Hash	92
9.1.1	Little-endian e Big-endian	92
9.2	MD4/MD5 - Message Digest	92
9.2.1	Obiettivi di MD4/MD5	92
9.2.2	Padding del messaggio	93
9.2.3	Operazioni	94
9.2.4	Funzioni	94
9.2.5	Funzione di Compressione - MD4	95
9.2.6	Funzione di Compressione - MD5	95
9.2.7	Implementazione dell'algoritmo MD4	96
9.2.8	Implementazione dell'algoritmo MD5	97
9.3	Algoritmo SHA	99

9.3.1	Espansione del blocco di iterazione	99
9.3.2	Rappresentazione Algoritmo SHA-1	101
9.4	Breve differenza tra MD4/MD5 e SHA-1	101
10	Lezione 10 ed 11	102
10.1	Analisi	102
10.2	Montare un file immagine - Linux	102
10.2.1	Come montare l'immagine?	103
10.3	Montare un file immagine - Windows	105
10.3.1	OSFMount Tool	105
10.3.2	FTK Imager Tool	107
10.4	Pro vs. Contro di montare un file immagine	107
10.5	FTK Imager come strumento di Analisi	108
10.5.1	Come eseguire l'analisi con FTK Imager	109
10.5.2	Quali sono i limiti di FTK Imager?	110
10.5.3	Analisi di un file immagine	111
10.5.4	La GUI di FTK Imager	112
10.5.5	Export dei file di interesse	114
10.6	Strumenti software per l'analisi	115
10.6.1	Toolkit - FTK vs. Autopsy	115
10.6.2	Toolkit - File Immagine Supportati	116
10.6.3	Toolkit - File System Supportati	116
10.6.4	Le viste nei Toolkit	117
10.6.5	File Type View	117
10.6.6	Known File View	119
10.6.7	Artefatti View	119
10.6.8	Image Gallery View	122
10.6.9	Video Gallery View	122
10.6.10	Social Analyzer View	123
10.6.11	Timeline View	123
10.7	Toolkit - Altri Strumenti	124
10.7.1	File Carving	124
10.7.2	Ricerche semi-manuali	124
10.7.3	Indicizzazione	125
10.8	Ancora altri Strumenti	125
10.9	Export/Report Toolkit	126
11	Lezione 12	127
11.1	Autopsy - Configurazione	127
11.1.1	Central Repository	127
11.1.2	Creazione del Caso	128
11.1.3	Ricordiamo i file supportati da Autopsy	129
11.2	Autopsy - Interfaccia	129
11.3	Moduli di Elaborazione - Ingest Module	131
11.3.1	Ingest Module - Hash Lookup	131
11.3.2	Ingest Module - File Type Identification	134

11.3.3 Ingest Module - Extension Mismatch Detector	134
11.3.4 Ingest Module - Picture Analyzer	135
11.3.5 Ingest Modulo - Embedded File Extractor	135
11.3.6 Ingest Module - Email Parser	136
11.3.7 Ingest Module - Interesting Files Identifier	136
11.3.8 Ingest Module - Encryption Detection	137
11.3.9 Ingest Module - Plaso	138
11.3.10 Ingest Module - Virtual Machine Extractor	138
11.3.11 Ingest Module - Data Source Integrity	139
12 Lezione 13	140
12.1 Ingest Module - Recent Activity	140
12.1.1 Artefatti Web	140
12.1.2 Analisi dei Registri	142
12.1.3 Recycle Bin	143
12.2 Ingest Module - Keyword Search	144
12.3 File Search by Attributes	146
12.4 Search Central Repository	147
12.5 Ingest Module - Central Repository	147
12.6 New Evidence	148
12.7 Ingest Module - PhotoRec Carver	150
12.8 Ingest Module - Android Analyzer	151
12.9 Ingest Module - iLEAPP, aLEAPP	151
12.10 Viste Specializzate	152
12.10.1 Timeline Graphic Interface	152
12.10.2 Image Gallery Interface	153
12.10.3 Communication Interface	154
12.10.4 Geolocation Interface	155
12.11 Tag e Report	155
12.11.1 Tagging	155
12.11.2 Comments	156
12.11.3 Reporting	157
12.11.4 Reporting - Portable Case	157
13 Lezione 14	158
13.1 L'analisi - Il disco	158
13.2 L'analisi - I volumi	159
13.2.1 Indirizzamento dei settori	161
13.3 Esempi più pratici	162
13.3.1 La lista delle partizioni in un file immagine	162
13.3.2 Estrazione delle partizioni in un file immagine	162
13.3.3 Recupero delle partizioni danneggiate in un file immagine	163
13.4 I Volumi - DOS Partition	163
13.4.1 Boot Code	165
13.4.2 Partition Table	165
13.5 DOS Partition - Analisi	166

13.6 I Volumi - Apple Partition Map	170
13.7 Apple Partition Map - Analisi	171
13.8 I Volumi - Guid Partition Table	172
13.9 Guid Partition Table - Analisi	174
14 Lezione 15	175
14.1 File System	175
14.1.1 Overview	175
14.2 File System Category	176
14.3 Content Category	177
14.3.1 Content Category - Analisi	177
14.4 Strategie di Allocazione	178
14.4.1 Strategia del Primo Disponibile	178
14.4.2 Strategia del Prossimo Disponibile	179
14.4.3 Strategia del più Adatto	180
14.4.4 Data Unit Danneggiate	180
14.5 File System - Slack Space	181
14.6 Metadata Category	182
14.6.1 Metadata Category - Analisi	182
14.6.2 Metadata Category - Logical File Address	182
14.6.3 Metadata Category - File Recovery	182
14.7 File Name Category	184
14.7.1 File Name Category - File Recovery	184
14.8 Application Category	186
14.8.1 Application Category - Analisi	186
15 Lezione 16	187
15.1 FAT File System	187
15.1.1 Physical Layout	187
15.1.2 File System Category	188
15.1.3 Boot Sector	189
15.1.4 Boot Sector - Analisi	191
15.1.5 FSINFO	192
15.1.6 FSINFO - Analisi	192
15.1.7 Physical Layout Updated	192
15.2 Content Category	193
15.2.1 FAT Structure	193
15.2.2 FAT - Analisi	194
15.3 Metadata Category	195
15.3.1 Directory Entry	195
15.3.2 Directory Entry - Analisi	196
15.3.3 FAT - Cluster Chain	197
15.3.4 Metadata Category - Directory	198
15.3.5 Metadata Category - Informazioni Temporali	199
15.4 File Name Category	200

16 Lezione 17	201
16.1 NT File System	201
16.1.1 Master File Table	201
16.1.2 MFT Entry	202
16.1.3 MFT - Analisi	203
16.2 File System Metadata	204
16.3 Attributi	204
16.3.1 Tipi di Attributi standard	206
16.4 Base/Non-Base MFT Entry	206
16.5 Sparse Attributes	207
16.6 Attribute Header	208
16.6.1 Run	209
16.7 File System Category	210
16.7.1 \$MFTMIRR File	211
16.7.2 \$BOOT File	211
16.7.3 \$VOLUME File	212
16.7.4 \$ATTRDEF File	213
16.8 File System Category - Analisi	214
16.9 Content Category	214
16.9.1 \$BITMAP File	214
16.9.2 \$BADCLUS File	215
17 Lezione 18	216
17.1 Metadata Category	216
17.1.1 \$STANDARD_INFORMATION Attribute	217
17.1.2 \$FILE_NAME Attribute	218
17.1.3 \$DATA Attribute	219
17.1.4 \$ATTRIBUTE_LIST Attribute	219
17.1.5 \$SECURITY_DESCRIPTOR Attribute	220
17.2 Metadata Category - Algoritmi di Allocazione	220
17.2.1 Aggiornamento informazioni temporali	221
17.3 Metadata Category - Analisi	221
17.4 File Name Category	222
17.4.1 Root Directory	222
17.5 Application Category	223
17.6 Logging/Journaling	223
18 Lezione 19	224
18.1 Sistemi Operativi	224
18.2 Microsoft Windows	224
18.2.1 Windows - Storia	224
18.2.2 Users	224
18.2.3 Secure Boot	225
18.2.4 Registro di Sistema	225
18.2.5 Registro di Sistema - Analisi	228
18.2.6 Thumbnails	229

18.2.7	Shell Bag	229
18.2.8	Event Viewer	231
18.2.9	Application Data	232
18.2.10	File Swap	233
18.2.11	Pro vs. Contro di Windows	234
18.3	Apple OSx/MacOS	235
18.3.1	Overview	235
18.3.2	Configurazione	236
18.3.3	Configurazione Server	236
18.3.4	Cifratura	237
18.3.5	File Swap	237
18.3.6	Portachiavi	237
18.3.7	Analisi	238
18.4	Linux	239
18.4.1	Overview	239
18.4.2	Sistema	240
18.4.3	Log	241
18.4.4	Configurazioni	243
18.4.5	Home Directory	243
18.4.6	/VAR Directory	244
18.4.7	Analisi	244
19	Lezione 20	245
19.1	Mobile Forensics	245
19.1.1	Overview	245
19.1.2	Evidence	246
19.1.3	GSM/CDMA	246
19.1.4	La Raccolta	247
19.1.5	Acquisizione - Strumenti	249
19.1.6	Acquisizione - Memory Card	250
19.1.7	Acquisizione - SIM Card	250
19.1.8	Tipologie di Acquisizione	251
19.1.9	Cellebrite UFED - GUI	251
19.1.10	Manual Extraction	253
19.1.11	Logical Extraction	254
19.1.12	File System Extraction	256
19.1.13	Physical Extraction	258
19.1.14	ChipOff	260
19.2	Analisi - I Sistemi Operativi	261
19.2.1	Android	261
19.2.2	Apple	261
19.3	Analisi - App	261
19.4	Analisi - Strumenti	262
19.4.1	UFED Physical Analyzer	262
19.4.2	PlugIn	263
19.4.3	Creazione di una catena di PlugIn personalizzati	264

20 Lezione 21	268
20.1 La relazione Tecnica - Le Fasi	268
20.2 La prova digitale	268
20.3 Accertamenti	268
20.3.1 Accertamento Ripetibile	269
20.4 La Relazione Tecnica	269
20.5 Forma della Relazione Tecnica	271

1 Lezione 1

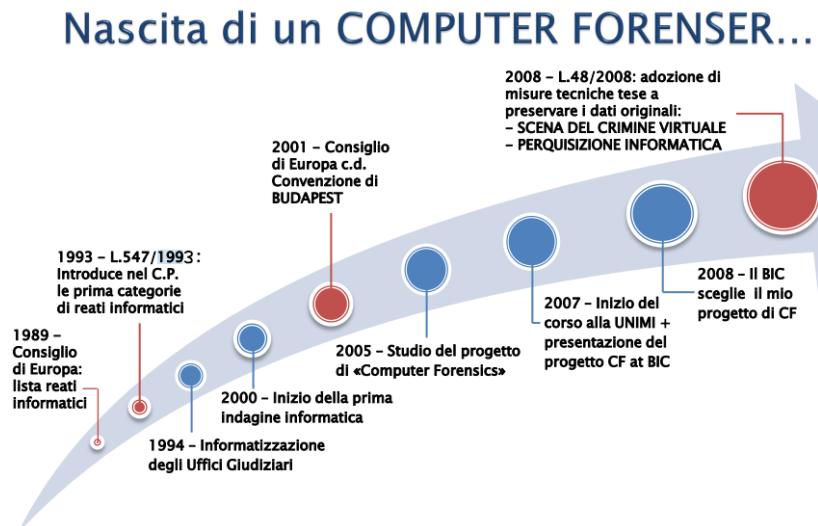
1.1 La Computer Forensics

Essa è l'insieme delle metodologie, scientificamente provate, ovvero che sono ben definite e ripetibili, finalizzate alla ricostruzione di eventi ai fini probatori che coinvolgono direttamente o indirettamente un supporto digitale.

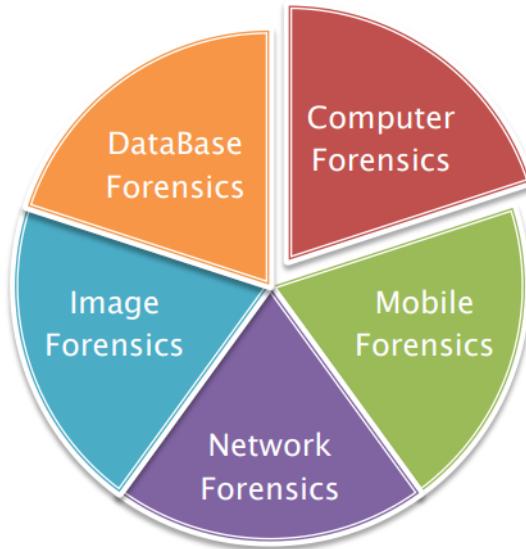
1.2 Nascita della Computer Forensics

Una lista dei primi reati informatici fu stilata nel consiglio d'Europa del **1989**, in Italia si iniziò a parlare di reati informatici quando furono inserite nel Codice Penale le prime categorie di reati informatici, questo con la legge **L.549/1993**. Nel **1994** furono per la prima volta informatizzati gli uffici giudiziari, quindi tutto quello che veniva effettuato tramite l'ausilio di carta e penna venne digitalizzato. Nel **2001**, dopo essere stati testimoni dell'attentato alle torri gemelle, venne fatto un consiglio d'europa al fine di prendere dei provvedimenti riguardo le falte che circondavano i mezzi informatici e che davano la possibilità di organizzare attentati...

Fù allora deciso di introdurre delle misure cautelari. In Italia ci siamo arrivati qualche anno dopo, nel **2008** di preciso, con la legge **L.48/2008** che introduceva l'adozione di misure tecniche tese a preservare i dati ed inoltre introduce per la prima volta la **scena del crimine virtuale** e la **perquisizione informatica**, fornendone delle linee guida sul come preservare i dati.

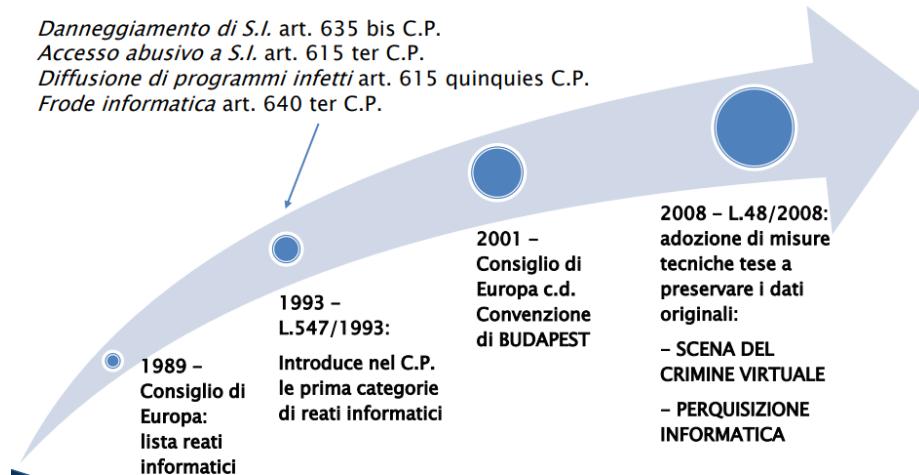


Da notare come la **Computer Forensics** sia solo una delle branche della **Digital Forensics** che racchiude appunto varie tipologie di scienze forensi.



1.3 Panorama Giuridico

Come accennato in precedenza il panorama giuridico è cambiato circa dal **1989** dopo il consiglio d'europa con una lista minima di reati informatici, in italia poi nel **1993**, introducendo il **Codice Penale** vennero introdotte anche le prime categorie di reati informatici mostrate in figura.



1.4 Metodologie della Computer Forensics

Una lista di procedure da effettuare per ogni caso. Questo iter resta sempre lo stesso da seguire ed è composto da:

- **Identificazione:** In questa fase bisogna individuare i dispositivi che possono contenere dati rilevanti.
- **Raccolta - Validazione - Preservazione:** Queste tre fasi sono necessarie alla produzione della **Copia Forense**, ovvero una copia conforme all'originale che serve a non alterare i dati originali, è quindi totalmente sconsigliato lavorare sui dati originali, bensì sulla copia forense.
- **Analisi - Interpretazione:** In queste due fasi lo scopo è quello di analizzare la copia forense e ricostruire le dinamiche dei fatti per poi interpretare i dati correttamente.
- **Documentazione:** Fase della creazione di una documentazione ben dettagliata e ben scritta di quanto trovato.
- **Presentazione:** Alla fine si dovranno presentare i fatti e la documentazione redatta al magistrato/giudice.

2 Lezione 2

2.1 Procedimento Penale

Le attività di indagine e poi il processo stesso, si svolgono in un ufficio detto **Procura della Repubblica**, qui si trova un magistrato chiamato **Pubblico Ministero** il quale gestisce e coordina le indagini. Mentre all'interno del **Tribunale** troviamo un altro magistrato che però si chiama **Giudice** che ha il compito di valutare le tesi di entrambe le parti durante il processo.

2.1.1 Fase Iniziale - Iscrizione della notizia di reato

L'**iscrizione della notizia di reato** è la formalizzazione di una denuncia o di un accertamento di polizia che viene inoltrata al **Pubblico Ministero** che si farà carico di svolgere le indagini preliminari avvalendosi delle opportune forze dell'ordine (polizia, carabinieri, finanza, polizia locale) le quali vengono chiamate più generalmente **Polizia Giudiziaria** per tutta la durata delle indagini. Quando il **Pubblico Ministero** riceve una notizia di reato si fa carico di iscriverla in un apposito registro, detto **Registro Generale Notizie di Reato (R.G.N.R.)** dove le denunce vengono divise in tre macrocategorie:

- Denunce contro soggetti noti.
- Denunce contro ignoti.
- Fatti non costituenti reato.

2.1.2 Indagini Preliminari

Durante la fase di **indagini preliminari** il Pubblico Ministero deve quindi cercare prove, tramite interrogatori, documenti oppure informazioni provenienti da supporti digitali. Il **P.M.** inoltre scambia informazioni, durante tutta la fase delle indagini preliminari, con il **Giudice delle Indagini Preliminari**. Il **P.M.** ha a sua disposizione due strumenti fondamentali, che sono:

- La **perquisizione**, ovvero la fase di ricerca delle prove su un fondato sospetto, nel caso ci fossero prove utili, si passa alla fase di **tutela** delle prove per fornire garanzia, poi si passa alla fase di..
- **Sequestro**, questo è utilizzato per tutelare la prova da possibili alterazioni. Questo è utilizzato anche in circostanze di accertamento tecnico, come specificato nell'articolo **253 C.P.P.**, qualora non si avessero strumenti specifici nell'immediatezza.

Dopo aver sequestrato un reperto inizia una **catena di custodia** che non è altro che la rimozione del reperto dall'utilizzo del proprietario passandone la gestione alle autorità giudiziarie e da lì in poi questo dovrà essere tutelato ed ogni suo utilizzo o spostamento dovrà essere documentato e verbalizzato.

2.1.3 Accertamento Tecnico

Durante la fase di indagini, il **P.M.**, può avere la necessità di effettuare degli **accertamenti tecnici**, questi richiedono competenze tecniche ben specifiche e che esulano dalle competenze dell'organo inquirente. Il **P.M.** quindi può avvalersi di un **consulente tecnico** per effettuare i dovuti accertamenti. Ci sono due tipologie di accertamenti da distinguere però:

- **Accertamento Tecnico Ripetibile**: questo tipo di accertamento è definito dall'articolo 359 C.P.P.
- **Accertamento Tecnico Irripetibile**: questo tipo di accertamento è definito dall'articolo 360 C.P.P., esso introduce degli elementi di garanzia. Si dice irripetibile un accertamento quando il reperto sequestrato, o a causa della metodologia di accertamento o a causa della natura stessa del reperto, la sua analisi porta inevitabilmente all'alterazione del reperto.

Da questo punto parte un principio che è il **contraddittorio** ovvero il **P.M.** sarà tenuto ad avvisare la parte offesa dell'accertamento da svolgere e qualora la parte offesa volesse può decidere di nominare un consulente tecnico di parte.

2.1.4 Misure Cautelari

Se durante la fase di indagini preliminari il **P.M.** ha il sospetto che si verifichi una delle seguenti opzioni:

- **Inquinamento delle prove.**
- **Fuga dell'indagato/imputato.**
- **Reiterazione del reato.**

Allora il **P.M.** può richiedere al **G.I.P.** l'applicazione di una misura cautelare, che può essere di tipo:

- **Personale**: se limitante della libertà personale (**coercitiva**), oppure limitante delle facoltà o diritti (**interdittiva**).
- **Reale**: se agisce su beni o cose.

2.1.5 Incidente Probatorio

Qualora ci fosse la necessità di anticipare la formazione/acquisizione di una prova, che di solito avviene durante la fase dibattimentale, durante le indagini preliminari, si fa richiesta al **G.I.P.** per consolidare quella prova. Questa procedura viene scelta quando vi sono potenziali limitazioni di tempo legate alla formazione della prova e pertanto non la si vuole rimandare a un futuro dibattimento, in quanto si vuole evitare il rischio che, con il trascorrere del tempo, la fonte di prova si comprometta o venga meno la genuinità della prova stessa. Tale procedura avviene più raramente dei normali atti di indagine, o comunque in modo straordinario, e per tale motivo viene definita "incidente".

2.1.6 Richiesta di Archiviazione

Al termine delle indagini preliminari il **P.M.** può presentare al **G.I.P.** la richiesta di archiviazione qualora l'attività investigativa non ha portato alla luce prove a sostenere l'accusa. Ci sono inoltre svariati motivi per decidere di inoltrare una richiesta di archiviazione, tra questi abbiamo:

- Gli elementi acquisiti nelle indagini non sono idonei a sostenere l'accusa.
- L'autore del reato è rimasto ignoto.
- Il reato risulta estinto, ovvero non più punibile dalla legge.
- Il fatto non è previsto dalla legge come reato.
- Il fatto risulta particolarmente tenue.

La parte offesa potrebbe presentare una richiesta di opposizione all'archiviazione al **G.I.P.** o potrebbe essere stesso il **G.I.P.** a respingere la richiesta di archiviazione.

2.1.7 Rinvio a Giudizio

Qualora le informazioni raccolte dal **P.M.** siano sufficienti a sostenere l'accusa, quest'ultimo ha la facoltà di esercitare l'azione penale, tramite il **rinvio a giudizio**, ovvero la possibilità di formalizzare l'accusa e rendere così l'indagato un **imputato**, che verrà avvisato tramite l'avviso **415 bis C.P.P.** indicando il capo di imputazione.

2.1.8 Udienza Preliminare

Il rinvio a giudizio può avvenire per due strade:

- Attraverso il **Giudice dell'Udienza Preliminare** che segna il passaggio dalla fase procedurale alla fase **processuale**. Qui l'imputato può chiedere al giudice di essere **prosciolto** oppure di **rinunciare alla fase dibattimentale**, arrivando direttamente alla sentenza.
- In alternativa si può procedere per **Citazione Diretta a Giudizio**.

2.1.9 Citazione Diretta a Giudizio

Un'alternativa all'udienza preliminare è la **Citazione Diretta a Giudizio**; ci sono alcuni reati che non hanno una pena prevista particolarmente importante, questi sono quei reati che prevedono una pena massima di 4 anni, per questa tipologia di reati il **P.M.** ha la possibilità di bypassare il **G.U.P.** ed andare direttamente in tribunale per la **fase dibattimentale** e questo appunto avviene tramite la **Citazione Diretta a Giudizio**.

2.1.10 Fase Dibattimentale

Questo non è altro che il processo vero e proprio ed è la fase centrale di un processo penale. La struttura, in questa fase, può prevedere il singolo giudice, quindi diremo che c'è una **composizione monocratica**, oppure il giudice può essere affiancato da altri due giudici ed essere quindi in **collegio**, oppure, in alcuni casi, ci può essere una **giuria popolare**. Lo scopo durante questa fase è quello di raccogliere e far diventare delle prove tutte le informazioni che hanno raccolto le due parti, queste prove sono dette **prove documentali**. Vanno poi ascoltati eventuali testimoni (**esame testimoniale**) le cui deposizioni diventano prove. Se infine ci fosse necessità di un ulteriore accertamento tecnico sarà facoltà del giudice nominare un perito (**perizia**).

2.1.11 Sentenza

Al termine del processo di primo grado c'è una **sentenza**, essa è divisa in:

- **Proscioglimento** → che è a sua volta suddiviso in:
 - **Sentenza di non doversi procedere**: qualora mancasse una condizione tecnica di procedibilità o una causa estintiva del reato.
 - **Sentenza di assoluzione**: il giudice si convince che l'imputato è innocente.
- **Condanna**: l'imputato risulta essere colpevole, viene quindi condannato.

2.1.12 Impugnazione

Una volta che un imputato riceve una condanna, diciamo che per lui non è proprio l'ultima spiaggia, infatti quest'ultimo può opporsi **impugnando** la sentenza di primo grado, arrivando a dibattere nuovamente presso un tribunale di livello superiore, ovvero la **Corte d'Appello** che è il secondo grado di giudizio. Esiste infine un terzo ed ultimo grado di giudizio che non giudica l'imputato ma bensì la sentenza d'appello, questo quando si ritiene che il processo sia stato condotto non interpretando correttamente le leggi, in questo caso si dibatte alla **Corte di Cassazione**.

2.1.13 Giudicato Penale

Una volta decorso il periodo nel quale è possibile per l'imputato impugnare la sentenza, allora quella sentenza diventerà **irrevocabile**, non sarà quindi più possibile appellarla ed avremo quindi il **Giudicato Penale**. L'imputato, condannato o prosciolto, non può essere nuovamente messo a processo per il medesimo fatto storico.

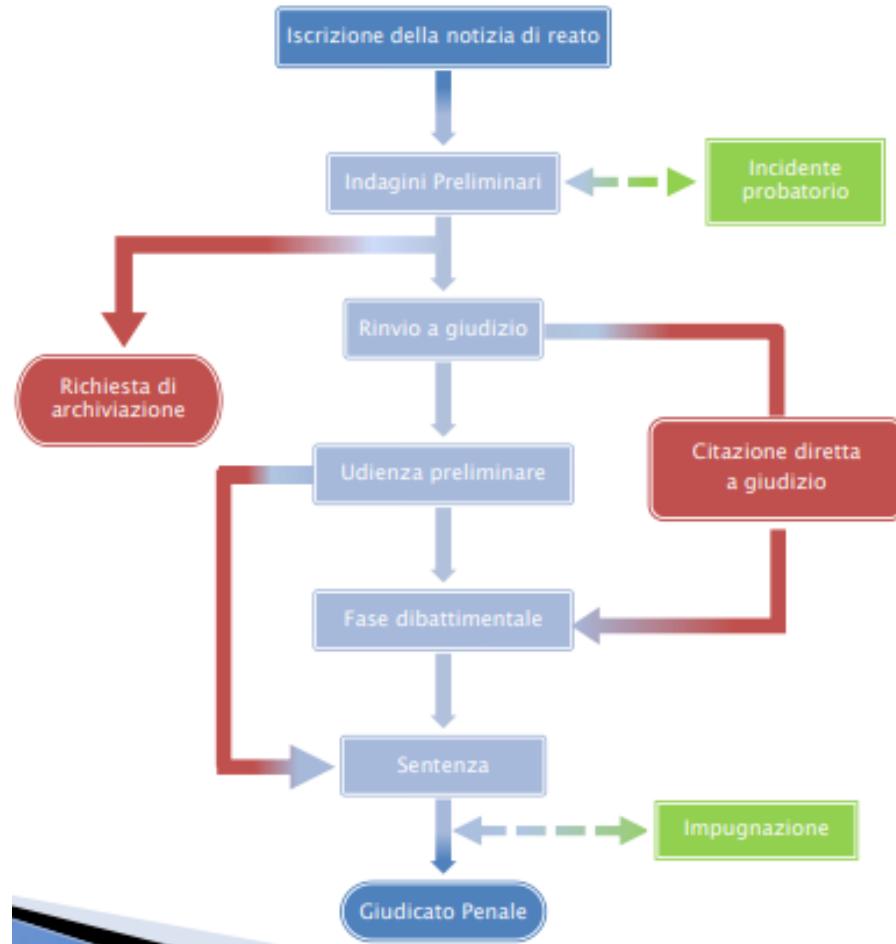


Immagine raffigurante tutto il processo sopra descritto nei minimi particolari.

2.2 Procedimento Civile

Penale vs Civile

- | | |
|---|--|
| <ul style="list-style-type: none">1. Diritto Penale;2. Si realizza in due strutture diverse: Procura e Tribunale;3. Ha lo scopo di accertare la verità nell'interesse dello Stato e della collettività;
4. Si instaura anche d'ufficio.
5. il giudice non si pone una situazione di indifferenza, ma persegue uno scopo ben preciso: accettare la verità del reato; | <ul style="list-style-type: none">1. Diritto Privato;2. Si realizza in un'unica struttura: il Tribunale;3. Ha lo scopo di verificare l'esistenza di un diritto reclamato da un privato cittadino nei confronti di un altro e quale, tra le due parti in causa, ha ragione;4. Si instaura esclusivamente su iniziativa di una parte: l'attore5. il giudice si attiene solo alle prove presentate dalle parti, ponendosi in una posizione di equidistanza e imparzialità (principio dispositivo); |
|---|--|

2.2.1 Procedimento Ordinario

Il procedimento ordinario civile prevede quattro fasi, che sono:

- **Fase Introduttiva:** qui l'**attore** ovvero la parte che instaura un giudizio, tramite il suo avvocato espone i fatti che vengono posti a giudizio, detto **atto di citazione**. L'atto di citazione viene notificato alla controparte, che in questo caso viene detta **il convenuto**.
- **Fase Istruttoria:** qui vengono acquisite in giudizio le prove richieste dalle parti, esse possono essere:
 - **Testimoniali**
 - **Documentali**
 - **Consulenze Tecniche di Parte**
- Il giudice può nominare un **Consulente Tecnico d'Ufficio**.
- **Fase Conclusiva:** qui entrambe le parti devono chiarire definitivamente le proprie richieste anche alla luce di quanto emerso nel corso del procedimento.
- **Fase Decisoria:** qui non resta che prendere una decisione da parte del giudice, avendo a disposizione tutti gli elementi per pronunciarsi.

La netta differenza che si ha rispetto ad un procedimento penale è la mancanza di quegli organi e figure di intramezzo prima di arrivare al tribunale. Qui l'attore, quindi il cittadino, accusa il convenuto, ovvero un altro cittadino, attraverso il proprio avvocato viene portato in tribunale e sulla base dei fatti presentati si inizia il processo.

2.2.2 Procedimento con Ricorso

In questa tipologia di procedimento, che risulta per lo più simile al precedente, tranne che per un fatto, ovvero, la fase istruttoria viene aggregata a quella introduttiva in modo tale da abbreviare il procedimento, inoltre il nome dei soggetti cambia. Le fasi che avremo durante questo procedimento quindi saranno:

- **Fase Introduttiva:** qui il **ricorrente**, ovvero la parte che instaura giudizio, tramite il proprio avvocato espone i fatti che vengono posti a giudizio direttamente al giudice in questo caso. Il giudice poi emetterà un decreto di fissazione dell'udienza ed il ricorrente dovrà notificare l'udienza alla controparte, ovvero al **resistente**. Le parti devono già esporre tutte le proprie difese e formulare le istanze istruttorie.

- **Fase Conclusiva:** qui entrambe le parti devono chiarire definitivamente le proprie richieste anche alla luce di quanto emerso nel corso del procedimento.
- **Fase Decisoria:** qui non resta che prendere una decisione da parte del giudice, avendo a disposizione tutti gli elementi per pronunciarsi.

3 Lezione 3

3.1 Attori del Procedimento Penale

- Pubblico Ministero → P.M.
- Polizia Giudiziaria → P.G.
- Parte Offesa
- Indagato/Imputato
- Giudice delle Indagini Preliminari → G.I.P.
- Giudice dell'Udienza Preliminare → G.U.P.
- Giudice del Dibattimento [Monocratico - Collegiale]

3.2 Struttura Organizzativa

- La **Procura** è l'ufficio dove vengono fatte le indagini preliminari, ovvero l'ufficio **inquirente**. Viene chiamato inquirente perché il nostro sistema giudiziario è di tipo accusatorio, quindi è il **P.M.** che ti accusa di qualcosa.

Uffici magistratura inquirente:

- Procure della Repubblica c/o i Tribunali Ordinari / per i Minorenni / Militari;
- Procure Generali c/o le Corti d'Appello;
- Procura Generale c/o la Suprema corte di Cassazione;

Questi uffici sono in ordine di grado di giudizio, dal primo, fino al terzo grado.

- Alla pari della magistratura inquirente abbiamo quella **giudicante** e questi non sono altro che gli uffici dove si tengono le fasi dibattimentali di un processo.

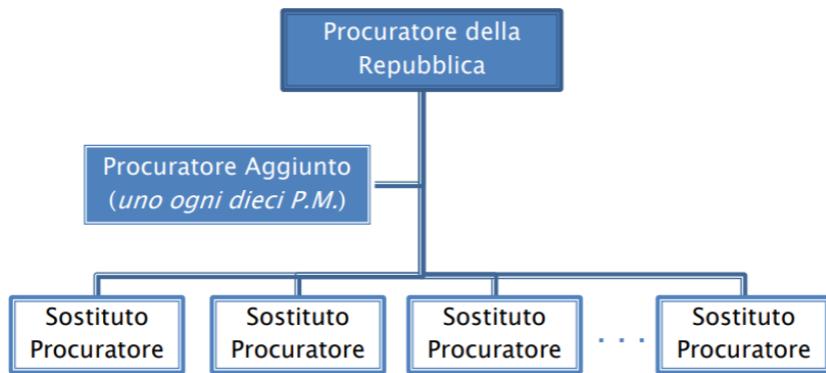
Uffici magistratura giudicante:

- Tribunali Ordinari;
- Tribunali per i Minorenni;
- Tribunali Militari;
- Corte di Appello;
- Suprema Corte di Cassazione;

Questi uffici sono in ordine di grado di giudizio, dal primo, fino al terzo grado.

3.3 Organizzazione della Procura

Gli uffici della procura sono organizzati in sezioni, ognuna delle quali è specializzata nel trattamento di specifici reati. Al vertice della procura vige il **Procuratore della Repubblica** che gestisce e coordina tutti i magistrati in procura avvalendosi dei **Procuratori Aggiunti** che sono divisi a loro volta uno ogni 10 **Sostituti Procuratori** che non sarebbero altro che i P.M.



3.4 Il Pubblico Ministero

Il **Pubblico Ministero** è un organo dell'amministrazione giudiziaria dello stato, esso è il titolare delle indagini ed ha il compito di esercitare l'azione penale. Rappresenta inoltre la pubblica accusa.

3.4.1 I poteri del Pubblico Ministero

- Dirige le indagini preliminari avvalendosi della polizia giudiziaria e si occupa di trovare le prove d'accusa nei confronti dell'indagato.
- Può decidere di nominare un consulente tecnico per effettuare degli accertamenti.
- Valuta l'esito delle indagini decidendo se archiviare oppure rinviare a giudizio.
- Quando decide di rinviare a giudizio esercita l'azione penale formulando il capo di imputazione, ovvero l'articolo di reato che è stato violato. Inoltre esso sostiene l'accusa una volta trovatisi in dibattimento.

3.5 La Polizia Giudiziaria

Sono forze dell'ordine con funzione **repressiva** che collaborano con il **P.M.** nelle attività di indagine e che dipendono direttamente dalla procura. Alcune forze dell'ordine hanno un ufficio, detto **aliquota**, che è sotto il diretto controllo del **P.M.**. Essi svolgono attività di indagine sia in modo autonomo, sia su delega del **P.M.**, possono svolgere:

- **Attività Informativa** : Acquisizione della notizia di reato ed inoltro al **P.M.**
- **Attività Investigativa** : Ricerca dell'autore del reato.
- **Attività di Prevenzione** : Impediscono l'aggravarsi di reati già commessi.
- **Attività Assicurativa** : Individua e protegge le fonti di prova.

3.6 La Persona Offesa

È il soggetto titolare del bene giuridico leso dall'autore di un reato. Ha il diritto di fare una **querela**. Può inoltre presentare **memorie**, indicare **elementi di prova**, nominare un difensore ed eventualmente dei consulenti tecnici.

3.7 Esposto, Denuncia e Querela, cosa sono ?

Quando la parte offesa decide di segnalare un reato lo può fare in diversi modi:

- **Esposto**: è la segnalazione all'autorità giudiziaria di un fatto allo scopo di far valutare se ricorre un'ipotesi di reato. In breve, faccio valutare il fatto alle autorità competenti per capire se è un reato o meno.
- **Denuncia**: è un atto con il quale si informa l'autorità giudiziaria di una notizia di reato perseguibile d'ufficio*. In breve, sono a conoscenza di un reato, di conseguenza lo denuncio.
- **Querela**: è una dichiarazione della persona offesa con la quale si esprime la volontà di punire il colpevole per un reato subito, non persegibile d'ufficio**.

*Un reato è **perseguibile d'ufficio** quando la denuncia/-querela è irrevocabile, come reati sessuali a danni di minori.

Mentre è **non persegibile d'ufficio quando può essere ritirata.

3.8 Indagato ed Imputato

- **Indagato:** è la persona nei cui confronti vengono svolte delle indagini a seguito dell'iscrizione di un fatto a lui addebitato. Resta indagato per tutto il periodo delle indagini preliminari, ovvero fin quando il **P.M.** non fa un rinvio a giudizio o un archiviazione.
- **Imputato:** è la persona indagata nei confronti della quale è stata esercitata l'azione penale, in questo caso il **rinvio a giudizio**. Resta imputato in ogni stato e grado del processo finché non risulti una sentenza definitiva. L'assenza dell'imputato in udienza non pregiudica lo svolgimento di quest'ultima, che si dice essere celebrata in **contumacia**.

Che sia indagato o imputato, la persona ha l'obbligo di farsi assistere da un **Difensore**. Essi possono produrre memorie o essere interrogati solo in presenza del proprio difensore. Possono entrambi avvalersi di consulenti tecnici.

3.9 Avvocato Difensore

Ha il ruolo di **assistenza**, resta una collaborazione di natura tecnica ed è l'unico a fare da tramite tra il cliente e il **P.M.** Esso ha inoltre il ruolo di **rappresentanza**, ovvero agisce in sostituzione dell'interessato nell'esercizio di diritti e facoltà. L'avvocato è nominato sia dalla parte offesa, sia da quella indagata/imputata. La sua presenza è condizione prima di legittimità e regolarità dello stesso procedimento penale. Se l'indagato/imputato non nomina un difensore di fiducia gliene viene assegnato uno d'ufficio. Il difensore può accedere agli atti delle indagini preliminari.

3.10 Giudice delle Indagini Preliminari

Ha la funzione di garanzia dell'indagato nella fase delle indagini preliminari. Può decidere se accogliere le richieste del **P.M.**, come ad esempio:

- Applicare misure cautelari, reali o personali che esse siano.
- Autorizzare e convalidare l'uso delle intercettazioni come mezzi di ricerca della prova.

Esso ha inoltre la funzione di garanzia dell'azione penale, ovvero può accogliere o rifiutare la richiesta di archiviazione. Non ha però autonomia probatoria, ovvero provvede solo alle richieste di parte, il difensore potrà interfacciarsi col **G.I.P.** ad esempio nel momento

in cui si richiede un incidente probatorio per ulteriori analisi. Il **G.I.P.** inoltre non ha fascicolo, gli atti sono quelli che il **P.M.** decide di allegare alle istanze che presenta.

3.11 Giudice dell’Udienza Preliminare

È un altro giudice che si trova all’interno del tribunale, il quale però interviene solo dopo l’esercizio dell’azione penale da parte del **P.M.**, esso giudica le richieste di rinvio a giudizio, esamina il fascicolo delle indagini preliminari e valuta le fonti delle prove raccolte, inoltre ascolta le ragioni della difesa dell’imputato. Il **G.U.P.** infine potrà decidere se emettere il decreto di rinvio a giudizio oppure emettere una sentenza di non luogo a procedere qualora non ci fossero i presupposti per andare a processo.

3.12 Giudice del Dibattimento

Una volta arrivati al processo ci sono i giudici del dibattimento. Il giudice del dibattimento presiede tutta la fase dibattimentale e alle relative udienze. Abbiamo poi il **P.M.** che sta alla destra del giudice ed il difensore alla sua sinistra. Quanto descritto è una composizione **monocratica**, ovvero con un singolo giudice, ma essa può variare in base alla situazione ed al reato, potremmo quindi avere una composizione **collegiale** dove il giudice è affiancato da altri due giudici detti **a latere**. Per reati più efferati esiste anche un ulteriore composizione detta **Corte d’Assise** dove è presente anche la **giuria popolare**.

Monocratica → vale per i reati che non superano i 4 anni di condanna.

Collegiale → vale quando la pena del reato supera i 4 anni ma non supera i 16/20 anni.

Corte d’Assise → vale quando il reato prevede un possibile ergastolo.

3.13 Il Computer Forensen nel Procedimento Penale

3.13.1 Le indagini preliminari

Se sono richieste particolari competenze, può essere nominato dall’autorità giudiziaria un consulente tecnico (Articolo 348 c.4 C.P.P), questo viene chiamato **Consulente Tecnico del Pubblico Ministero** quando viene nominato dal **P.M.**, oppure detto **Consulente Tecnico d’Ufficio (C.T.U.)**, viene invece chiamato **Ausiliario di Polizia**

Giudiziaria quando questo viene nominato direttamente dalla **Po-lizia Giudiziaria** per la necessità di un esperto per non inquinare le prove, quindi per permettere di raccogliere i dati in modo sicuro.

3.13.2 Il ruolo del C.F.

Per svolgere il suo ruolo, il Computer Forensen, deve applicare metodi e strumenti che garantiscano l'inalterabilità della prova, anche se non dettagliatamente descritti dalla legge. Il C.F. deve inoltre preservare la ripetibilità della procedura.

Attenzione! → Nella legge 48/2008 non è descritto il modo in cui non devi alterare i reperti. Di conseguenza, se un C.F. altera un reperto potrebbe essere indagato per averlo fatto intenzionalmente, reato di **Perizia Infedele**.

3.13.3 Accertamento Irripetibile (Art. 360 C.P.P.)

È la tipologia di accertamento che se compiuto comporta l'alterazione della fonte di prova e non garantisce più la ripetibilità della procedura. Dispositivi che ad esempio non sono in buono stato dovrebbero essere accertati irripetibilmente, ed è compito del C.F. far capire al **P.M.** che è più logico un accertamento di questo tipo in una situazione del genere. Il **P.M.** esegue questa attività di accertamento avvisando preventivamente l'indagato e il suo difensore in modo da dare loro la possibilità di assistere a tutta l'operazione a garanzia del rispetto delle procedure. L'indagato può nominare e farsi assistere da un consulente tecnico di parte.



3.13.4 Perito

In caso di un incidente probatorio o di un'udienza in cui sono richieste particolari competenze tecniche il giudice può nominare un consulente tecnico detto **Perito** che farà la sua analisi e sarà poi il giudice a confrontarla con le analisi del **C.T.U.** e/o del **C.T.P.**. Al consulente tecnico verrà dato un verbale del conferimento di incarico che non è altro che la richiesta di presenziare in processo ed effettuare un accertamento tecnico.



- **nr. Procedimento Penale:** è un numero progressivo, che si resetta ogni anno, per identificare ogni reato all'interno del **Registro Generale Notizie di Reato**. Di fianco a questo numero troviamo la dicitura *Mod.X* con un numero, questo può variare ed avere vari significati, tra cui:
 - **Mod.21** sta ad indicare che è un fascicolo con indagato *noto*, detto **fascicolo noti**.
 - **Mod.44** sta ad indicare che è un fascicolo con indagato *ignoto*.
 - **Mod.45** sta ad indicare che è un fascicolo con *fatti non constituenti reato*.
 - **Mod.46** sta ad indicare che è un fascicolo *anonimo*.
- **Art. 360:** sta ad indicare un accertamento di tipo *irripetibile*.
- **Reato:** sta descrivendo i vari reati di cui:
 - **Art. 615 ter** → accesso abusivo ad uno strumento informatico.
 - **Art. 640 ter** → frode informatica.
 - **Art. 416** → associazione a delinquere. (416 ter: associazione a stampo mafioso/camorristico)

il PM dà atto che nessun altro è comparso sino alle ore 12.55. Si dà atto che l'avviso ex art 360 c.p.p. è stato ritualmente notificato a tutti gli indagati e loro difensori, nonché alle persone offese.

I CTU a questo punto rendono le proprie generalità:

- Sono Lorenzo LAURATO nato a OMISSIS domiciliato in Napoli presso la società SSRI in Via Coroglio n.57;
- sono Consulente Tecnico nato a OMISSIS professionale in OMISSIS , domiciliato presso lo studio

A domanda se i CTU si trovino in una delle condizioni previste dall'art. 222 c.p.p., il CTU rispondono: "NO". Il Pubblico Ministero, quindi, informa i consulenti e le parti dell'oggetto dell'incarico e formula i seguenti quesiti:

"Previo esame dei reperti in sequestro (PC, telefoni, supporti informatici), forniranno i ctu -con software e dispositivi idonei allo scopo, collegialmente e d'intesa con la Polizia Postale delegata - copia forense delle memorie informatiche dei dispositivi elettronici in sequestro (come da elenchi inviati dalla Polizia Postale), operando secondo le regole della computer forensics in maniera da non alterare il contenuto dei reperti e da rendere l'esame e l'analisi ulteriormente ripetibile. Effettuano i CTU l'analisi del contenuto dei reperti, illustrando in particolare se emergano dati, documenti o tracce informative dell'attività di frode informatica e dei reati per cui si procede, nonché documentazione relativa all'incasso e movimentazione dei proventi di truffe e di sostituzione di persona. Dicano quant'altro utile a fini di giustizia".

Vista la complessità dei quesiti, i Consulenti Tecnici chiedono un termine per il deposito della relazione, che il Pubblico Ministero concede nella misura di giorni 60 dall'inizio delle operazioni.

Possiamo notare come attraverso l'articolo **222 C.P.P.** viene chiesto al **C.T.U.** se conosce gli indagati, riportando anche la risposta di quest'ultimo.

Richieste del C.T.U.

Data e Luogo inizio operazioni

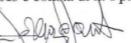
I CTU chiedono di essere autorizzati: esaminare i reperti in sequestro presso il proprio studio, usare il mezzo proprio per gli spostamenti, e ad avvalersi di collaboratori ausiliari, nonché a noleggiare il dispositivo denominato "CELLEBRITE UFED" essenziali per l'acquisizione forense delle memorie dei telefoni sequestrati; nonché all'acquisto di appositi hard disk per memorizzare le copie forensi.

Il P.M. autorizza quanto sopra richiesto.

Le parti presenti dichiarano che allo stato non intendono nominare propri CTP.

I CTU dichiarano che l'inizio delle operazioni avverrà **in data 25.9.2018 ore 10.00 presso lo studio SSRI del dott. Laurato in via Coroglio n.57**, con ritiro ed esame reperti.

Verbale chiuso alle ore 10.57 del 19.2.2018 e consta di 1 pagina

Letto e sottoscritto.
 - I CTU _____ 
 I Difensori _____
 Il M.O.T. _____

IL PUBBLICO MINISTERO

**COPIA CONFORME
ALL'ORIGINALE**

3.13.5 Verbale di consegna materiale

<p style="text-align: center;"> Guardia di Finanza NUCLEO DI POLIZIA ECONOMICO-FINANZIARIA NAPOLI Gruppo Tutela Mercato Beni e Servizi - Sezione Polizia Economica ed altre attività di p.g. Via Card. G. Sanfelice, 49 - 80134 - Napoli - ☎ 081/9703649 - - na1820000p@pec.gdf.it</p>	
<p>VERBALE DELLE OPERAZIONI COMPIUTE</p> <p>L'anno 2019 addì 25 del mese di ottobre, alle ore 13:00, presso gli uffici Nucleo pt in intestazione, viene redatto il presente atto.</p>	
<p style="text-align: center;">VERBALIZZANTI</p> <p>Lgt. c.s. <i>AGENTE DI P.G. A</i> Lgt. c.s. <i>AGENTE DI P.G. B</i></p>	
<p style="text-align: center;">PARTE</p> <p>ALFE' Marco, nato a OMISSIS. Identificato a mezzo patente di guida nr. OMISSIS rilasciata dalla MCTC di Napoli in data 05.03.2001 nella sua qualità di collaboratore di LAURATO Lorenzo, in altri atti già compiutamente generalizzato - CTU informatico del Pubblico Ministero.</p>	
<p style="text-align: center;">FATTO</p> <p>Il dott. P.M. – Sost. Proc. presso il Tribunale di Napoli, nell'ambito del procedimento penale xxxx/18, ha conferito l'incarico di C.T.U. informatico al sig. LAURATO Lorenzo, nato a OMISSIS, per l'esame del materiale informatico sequestrato in data 14 e 23 ottobre 2019, nei confronti dei seguenti soggetti:</p>	
<p>1. INDAGATO A, nato ad OMISSIS e residente in Avellino alla via ➤ nr. 1 pc marca HP nr. seriale ABCD123456789; ➤ nr. 1 pc marca PAVILLION nr. seriale ABCD123456789; ➤ nr. 1 pc marca ACER Aspire nr. seriale ABCD123456789; ➤ nr. 1 pc marca ACER Veriton nr. seriale ABCD123456789; ➤ nr. 1 pc marca GLITE nr. seriale ABCD123456789;</p>	
<p>2. INDAGATO B nato a S. Andrea di Conza (AV) il OMISSIS e residente ad Avellino, OMISSIS: ➤ nr. 1 pc portatile marca HP PAVILLON serial number: 1234ABCD con relativo alimentatore; ➤ pc fisso, modello ASPIRE AX 3950 serial number: 123456789ABCD ➤ nr. 1 pendrive marca DIKOM;</p>	
<p>3. INDAGATO C nato ad Atripalda (AV) il OMISSIS e residente ad Avellino, via OMISSIS: ➤ telefono cellulare marca XIAOMY corredato della scheda telefonica nr. ;</p>	
<p>Per quanto sopra, come concordato telefonicamente con il CTU LAURATO Lorenzo, il materiale sopra descritto viene consegnato a ALFE' Marco, per l'espletamento della consulenza tecnica</p>	
<p>Si rappresenta che quanto sopra descritto viene consegnato nei plachi approntati in sede di sequestro, come da verbali all'uopo redatti.</p>	
<p>L.C.S.</p>	
<p>I VERBALIZZANTI</p>	<p>LA PARTE</p>

3.13.6 Richiesta proroga termini

*V°, n° ente 17
Napoli, 21/5/19
Il SOST. PROCURATORE DELLA REPUBBLICA*

SSRI
Sicurezza Sistemi Reti Informatiche

PP N. xxxx/17 R.G.N.R. Mod.21

**Procura della Repubblica
presso il Tribunale di NAPOLI**

alla c.a. del Pubblico Ministero
Dott. PUBBLICO Ministero

Oggetto : Richiesta Proroga Termini

I sottoscritti Dott. Lorenzo Laurato e OMISSIONIS, nominati CTU dalla SVI nell'ambito del procedimento penale nr. xxxx/2017 R.G.N.R., considerata l'enorme quantità del materiale informatico sottoposto a sequestro, e considerato che il giorno 23/03/2019, scadono i termini di presentazione dell'elaborato peritale, con la presente

C H I E D O N O

una proroga dei termini di presentazione della relazione peritale di giorni 60 (sessanta) a partire dalla data del **"22-05-2019"**, periodo di scadenza previsto per portare a termine il proprio mandato.

Restando a Vs disposizione per ogni eventuale chiarimento, porge distinti saluti.

Napoli 20/05/2019

I Consulenti Tecnici d'Ufficio
Dott. Lorenzo Laurato

3.13.7 Incarico di perizia

Nr. Procedimento Penale	N. XXXX/19 R.G. P.M. N. YYYYY/19 R.G. Tribunale	COLLEGIO C	Nr. Registro Tribunale
Composizione del Collegio	TRIBUNALE DI NOLA SEZIONE PENALE VERBALE DI UDIMENTO (ART. 480 E SEGUENTI C.P.P.) Il giorno MERCOLEDÌ 4 DICEMBRE 2019 alle ore 10.30, in Nola dinanzi al Tribunale di Nola in composizione collegiale, composto da: Presidente DOTT. SRA. GIUDICE A Giudice DOTT. SRA. GIUDICE B Giudice DOTT. GIUDICE C con l'autorizzazione del Consigliere dott. OMISISS., che espressamente autorizzato si avvale, ove necessario, di personale tecnico per la redazione del verbale con la stenografia sig. sig.ra _____. La riproduzione fotografica sig. / sig.ra _____ sono presenti: il Pubblico Ministero P.M. Imputati: IMPUTATO A _____ (REMO/ A/GIA PRESENTE) IMPUTATO B _____ (REMO/ A/GIA PRESENTE) Persona Offesa: PERSONA OFFESA _____ (REMO/ A/GIA PRESENTE) Difensori: DOTT. AVV. DIFENSORE A _____ (REMO/ A/GIA PRESENTE) DOTT. AVV. DIFENSORE B _____ (REMO/ A/GIA PRESENTE) Testi presenti: _____ Si da atto ai fini della pratica formale della presenza del(i) dott(s). UNIVERSITÀ DEGLI STUDI DI CAMPANIA		
Imputati	Difensore Imputati		
Persona Offesa	Difensore Persona Offesa		
Perito			

*Il Presidente controlla la regolare costituzione delle parti.
 Comprato l'accertamento della costituzione delle parti.
 Preliminarmente, si è de' circa che c'è l'avv. Laurato e
 l'ing. Consulente di Parte. Il presidente procede alla constatazione
 dell'accertamento della costituzione delle parti, e quindi
 procede alla verifica della costituzionalità.
 L'avv. [difensore imputati] procede alla nomina di n. 2 consulenti propri
 già indicato nella lista Teste nonché l'ing. [Consulente di Parte].
 A questo punto il Tribunale rinvia al 4-3-20 ore 12.00 per il deposito
 della perizia.
 Perito presente edotto.
 Chiusura 10,45.*

Dal verbale stenotipico:

CONFERIMENTO DELL'INCARICO PRESIDENTE: No. Va bene, allora, senta, in questo caso noi abbiamo un telefono cellulare con il quale è stata registrata una conversazione tra presenti, cioè, una conversazione tra la Persona Offesa e gli Imputati di questo procedimento, ora, l'accertamento che lei dovrebbe fare, innanzitutto, quello di estrarre questa conversazione e anche di verificare se la genuinità della stessa, cioè, eventualmente se ci sono delle alterazioni, degli stop and go, qualche elemento che possa far dubitare della genuinità della conversazione, questo è il quesito, adesso, magari, lo specifichiamo un poco meglio, c'è qualcosa' altro che le Parti vogliono aggiungere?

3.13.8 Verbale operazioni compiute

Tribunale di Nola
Coll. C

4 Lezione 4

4.1 Il Reato

È quell'**illecita azione** o **omissione**, tesa a ledere un bene tutelato giuridicamente e a cui viene corrisposta una **pena**.

Con il termine **illecito** intendiamo identificare qualcosa che sia contrario all'ordinamento giuridico. Tutto ciò che risulta illecito, tramite un **azione** può essere identificato in tre tipologie di azioni:

- **Doloso**: un azione con consapevolezza e volontà di commettere un reato.
- **Preterintenzionale**: un azione con conseguenze più gravi di quelle volute.
- **Colposo**: quando manca la volontà di determinare un qualsiasi evento costituente reato, ma l'evento si è verificato ugualmente per negligenza, imprudenza, imperizia o per inosservazione di leggi/regolamenti, ordini o discipline.

Parliamo di **omissione** quando non si impedisce un evento che si aveva l'obbligo di impedire, questo equivale a cagionarlo.

La **pena** invece è una sanzione predisposta per la violazione di un precetto penale.

4.2 Il Reato Informatico

In ambito penale il **reato informatico** è un illecito in cui è coinvolto un computer come strumento o come oggetto, è un illecito per il quale sono necessarie delle competenze informatiche. Queste sono tuttavia delle definizioni troppo ampie e non riescono a delimitare un reato informatico, infatti a livello internazionale si è deciso di concordare una tipologia di comportamenti etichettati come **reati informatici**, non esiste quindi una definizione precisa, bensì ci sono delle categorie di reati. La pietra miliare tra le regolamentazioni sui reati informatici è la legge **547/1993** che introduce nel codice penale le prime categorie per questi reati. Nel **2001** invece, con il consiglio d'europa viene proposta quella che sarà poi approvata in italia nel 2008, ovvero la legge **48/2008** che è l'adozione di misure tecniche atte a preservare i dati originali per un reato informatico, qui viene introdotta la **scena del crimine digitale** e la **perquisizione informatica**.

4.3 Consiglio d'Europa del 1989

Qui vennero elaborate e stilate due liste di abusi per la categoria informatica, queste sono:

- **Lista Minima:** ci sono le condotte criminose che gli stati devono reprimere con delle sanzioni penali.
 - frode informatica;
 - falso in documenti informatici;
 - danneggiamento di dati e programmi;
 - sabotaggio informatico;
 - accesso non autorizzato ad un sistema informatico;
 - intercettazione non autorizzata di comunicazioni informatiche;
 - riproduzione non autorizzata di un programma protetto;
 - riproduzione non autorizzata della topografia di un prodotto a semiconduttori;
- **Lista Facoltativa:** ci sono comportamenti non ritenuti eccessivamente offensivi la cui repressione è rimandata alla valutazione dei singoli stati.
 - alterazione di dati o di programmi (*senza danneggiamento*);
 - spionaggio informatico;
 - utilizzazione non autorizzata di un elaboratore;
 - utilizzazione non autorizzata di un programma informatico;

4.4 Legge 547/1993

L'Italia dopo il consiglio d'europa del 1989 introduce, con la legge **547/1993**, nel codice penale diverse figure di reato informatico collocando questi reati al fianco di reati già esistenti valutando quel reato come se fosse commesso sfruttando la tecnologia informatica. Ad esempio abbiamo:

4.4.1 Art. 392 C.P. - Esercizio arbitrario delle proprie ragione con violenza sulle cose

- ▶ Chiunque, al fine di esercitare un preteso diritto, potendo ricorrere al giudice, si fa arbitrariamente ragione da sé medesimo, mediante violenza sulle cose, è punito, a querela della persona offesa [120; c.p.p. 336, 340], con la multa fino a euro 516.
- ▶ Agli effetti della legge penale, si ha violenza sulle cose allorché la cosa viene danneggiata o trasformata, o ne è mutata la destinazione.
- ▶ *Si ha altresì, violenza sulle cose allorché un programma informatico viene alterato, modificato o cancellato in tutto o in parte ovvero viene impedito o turbato il funzionamento di un sistema informatico o telematico.*

La parte

sottolineata rappresenta l'affiancamento del reato informatico ad un articolo già presente.

4.4.2 Art. 420 C.P. - Attentato ad impianti di pubblica utilità

- ▶ Chiunque commette un fatto diretto a danneggiare o distruggere impianti di pubblica utilità, è punito, salvo che il fatto costituisca più grave reato, con la reclusione da uno a quattro anni.
- ▶ *La pena di cui al primo comma si applica anche a chi commette un fatto diretto a danneggiare o distruggere sistemi informatici o telematici di pubblica utilità, ovvero dati, informazioni o programmi in essi contenuti o a essi pertinenti. (*)*
- ▶ *Se dal fatto deriva la distruzione o il danneggiamento dell'impianto o del sistema, dei dati, delle informazioni o dei programmi ovvero l'interruzione anche parziale del funzionamento dell'impianto o del sistema, la pena è della reclusione da tre a otto anni. (*)*

(*) abrogati con la legge n. 48 del 18/03/2008

4.4.3 Art. 491-bis C.P. - Documenti Informatici

- ▶ Se alcuna delle falsità previste dal presente capo riguarda un documento informatico pubblico [o privato] {avente efficacia probatoria}, si applicano le disposizioni del capo stesso concernenti [rispettivamente] agli atti pubblici [e le scritture private]
- ▶ A tal fine per documento informatico si intende qualunque supporto informatico contenente dati o informazioni aventi efficacia probatoria o programmi specificamente destinati ad elaborarli (*)

{} termini soppressi dal d.lgs. n. 7 del 15/01/2016

{} termini aggiunti con la legge n. 48 del 18/03/2008

(*) abrogato con la legge n. 48 del 18/03/2008

4.4.4 Art. 615-ter C.P. - Accesso abusivo ad un sistema informatico o telematico

- Chiunque 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, è punito con la reclusione fino a tre anni.
- La pena è della reclusione da uno a cinque anni:
 - 1) se il fatto è commesso da un pubblico ufficiale o da un incaricato di un pubblico servizio, con abuso dei poteri, o con violazione dei doveri inerenti alla funzione o al servizio, o da chi esercita anche abusivamente la professione di investigatore privato, o con abuso della qualità di operatore del sistema;
 - 2) se il colpevole per commettere il fatto usa violenza sulle cose o alle persone, ovvero se è palesemente armato;
 - 3) se dal fatto deriva la distruzione o il danneggiamento del sistema o l'interruzione totale o parziale del suo funzionamento, ovvero la distruzione o il danneggiamento dei dati, delle informazioni o dei programmi in esso contenuti.
- Qualora i fatti di cui ai commi primo e secondo riguardino sistemi informatici o telematici di interesse militare o relativi all'ordine pubblico o alla sicurezza pubblica o alla sanità o alla protezione civile o comunque di interesse pubblico, la pena è, rispettivamente, della reclusione da uno a cinque anni e da tre a otto anni.
- Nel caso previsto dal primo comma il delitto è punibile a querela della persona offesa; negli altri casi si procede d'ufficio

4.4.5 Art. 615-quater C.P. - Detenzione e diffusione abusiva di codici di accesso a sistemi informatici o telematici

- Chiunque, al fine di procurare a sé o ad altri un profitto o di arrecare ad altri un danno, abusivamente si procura, riproduce, diffonde, comunica o consegna codici, parole chiave o altri mezzi idonei all'accesso ad un sistema informatico o telematico, protetto da misure di sicurezza, o comunque fornisce indicazioni o istruzioni idonee al predetto scopo, è punito con la reclusione sino a un anno e con la multa sino a lire 10 milioni (*cinquemilacentosessantaquattro euro*).
- La pena è della reclusione da uno a due anni e della multa da lire 10 milioni a 20 milioni (*cinquemilacentosessantaquattro euro a diecimilatrecentoventinove euro*) se ricorre taluna delle circostanze di cui ai numeri 1) e 2) del quarto comma dell'articolo 617quater.

4.4.6 Art. 615-quinquies C.P. - Diffusione di programmi diretti a danneggiare o interrompere un sistema informatico

- Chiunque diffonde, comunica o consegna un programma informatico da lui stesso o da altri redatto, avente per scopo o per effetto il danneggiamento di un sistema informatico o telematico, dei dati o dei programmi in esso contenuti o a esso pertinenti, ovvero l'interruzione, totale o parziale, o l'alterazione del suo funzionamento, è punito con la reclusione sino a due anni e con la multa sino a lire 20 milioni.

4.4.7 Art. 616 C.P. - Violazione, sottrazione e soppressione di corrispondenza

- Chiunque prende cognizione del contenuto di una corrispondenza chiusa, a lui non diretta, ovvero sottrae o distrae, al fine di prenderne o di farne da altri prender cognizione, una corrispondenza chiusa o aperta, a lui non diretta, ovvero, in tutto o in parte, la distrugge o sopprime, è punito, se il fatto non è preveduto come reato da altra disposizione di legge, con la reclusione fino a un anno o con la multa da trenta euro a cinquecentosedici euro.
- Se il colpevole, senza giusta causa, rivela, in tutto o in parte, il contenuto della corrispondenza, è punito, se dal fatto deriva documento ed il fatto medesimo non costituisce un più grave reato, con la reclusione fino a tre anni [618].
- Il delitto è punibile a querela della persona offesa.
- *Agli effetti delle disposizioni di questa sezione, per "corrispondenza" s'intende quella epistolare, telegrafica o telefonica, informatica o telematica ovvero effettuata con ogni altra forma di comunicazione a distanza.*

4.4.8 Art. 617-quater C.P. - Intercettazione, impedimento o interruzione illecita di comunicazioni informatiche

- Chiunque fraudolentemente intercetta comunicazioni relative a un sistema informatico o telematico o intercorrenti tra più sistemi, ovvero le impedisce o le interrompe, è punito con la reclusione da sei mesi a quattro anni.
- Salvo che il fatto costituisca più grave reato, la stessa pena si applica a chiunque rivela, mediante qualsiasi mezzo d'informazione al pubblico, in tutto o in parte, il contenuto delle comunicazioni di cui al primo comma.
- I delitti di cui ai commi primo e secondo sono punibili a querela della persona offesa.
- Tuttavia si procede d'ufficio e la pena è della reclusione da uno a cinque anni se il fatto è commesso:
 - 1) in danno di un sistema informatico o telematico utilizzato dallo stato o da altro ente pubblico o da impresa esercente servizi pubblici o di pubblica necessità;
 - 2) da un pubblico ufficiale o da un incaricato di un pubblico servizio, con abuso dei poteri e con violazione dei doveri inerenti alla funzione o al servizio, ovvero con abuso della qualità di operatore del sistema;
 - 3) da chi esercita anche abusivamente la professione di un investigatore privato

4.4.9 Art. 617-quinquies C.P. - Installazione di apparecchiature atte a intercettare, impedire o interrompere comunicazioni informatiche o telematiche

- Chiunque, fuori dai casi consentiti dalla legge, installa apparecchiature atte a intercettare, impedire o interrompere comunicazioni relative a un sistema informatico o telematico ovvero intercorrenti tra più sistemi, è punito con la reclusione da uno a quattro anni.
- La pena è della reclusione da uno a cinque anni nei casi previsti dal quarto comma dell'art. 617-quater.

4.4.10 Art. 617-sexies C.P. - falsificazione, alterazione o soppressione del contenuto di comunicazioni informatiche o telematiche

- ▶ Chiunque, al fine di procurare a sé o ad altri un vantaggio o di arrecare ad altri un danno, forma falsamente ovvero altera o sopprime, in tutto o in parte, il contenuto, anche occasionalmente intercettato, di taluna delle comunicazioni relative a un sistema informatico o telematico o intercorrenti tra più sistemi, è punito, qualora ne faccia uso o lasci che altri ne facciano uso, con la reclusione da uno a quattro anni.
- ▶ La pena è della reclusione da uno a cinque anni nei casi previsti dal quarto comma del l'art. 617-quater.
- ▶ Nel caso previsto dal primo comma il delitto è punibile a querela della persona offesa.(*)

() aggiunto con il D.Lgs. n. 36 del 10/04/2018*

4.4.11 Art. 621 C.P. - Rivelazione del contenuto di documenti segreti

- ▶ Chiunque, essendo venuto abusivamente a cognizione del contenuto, che debba rimanere segreto, di altri atti o documenti, pubblici o privati, non costituenti corrispondenza, lo rivela, senza giusta causa, ovvero l'impiega a proprio o altrui profitto, è punito, se dal fatto deriva documento, con la reclusione fino a tre anni o con la multa da centotré euro a milletrentadue euro.
- ▶ ***Agli effetti della disposizione di cui al primo comma è considerato documento anche qualunque supporto informatico contenente dati, informazioni o programmi.***
- ▶ Il delitto è punibile a querela della persona offesa.

4.4.12 Art. 623-bis C.P. - Altre comunicazioni e conversazioni

- ▶ Le disposizioni contenute nella presente sezione, relative alle comunicazioni e conversazioni telegrafiche, telefoniche, informatiche o telematiche, si applicano a qualunque altra trasmissione a distanza dei suoni, immagini o altri dati

4.4.13 Art. 635-bis C.P - Danneggiamento di sistemi informatici o telematici

- ▶ Chiunque distrugge, deteriora o rende, in tutto o in parte, inservibili sistemi informatici o telematici altrui, ovvero programmi, informazioni o dati altrui, è punito, salvo che il fatto costituisca più grave reato, con la reclusione da sei mesi a tre anni.
- ▶ Se ricorre una o più delle circostanze di cui al secondo comma dell'art. 635, ovvero se il fatto è commesso con abuso della qualità di operatore del sistema, la pena è della reclusione da uno a quattro anni.

4.4.14 Art. 640-ter C.P. - Frode informatica

- ▶ Chiunque, alterando in qualsiasi modo il funzionamento di un sistema informatico o telematico o intervenendo senza diritto con qualsiasi modalità sui dati, informazioni o programmi contenuti in un sistema informatico o telematico o ad esso pertinenti, procura a sé o ad altri un ingiusto profitto con altrui danno, è punito con la reclusione da sei mesi a tre anni e con la multa da lire 100 mila a 2 milioni (*cinquantuno euro a millecentrentadue euro*).
- ▶ La pena è della reclusione da uno a cinque anni e della multa da lire 600 mila a 3 milioni (*euro 600 a euro 3.000*) se ricorre una delle circostanze previste dal numero 1) del secondo comma dell'art. 640, ovvero se il fatto è commesso con abuso della qualità di operatore del sistema.
- ▶ Il delitto è punibile a querela della persona offesa, salvo che ricorra taluna delle circostanze di cui al secondo comma o un'altra circostanza aggravante

4.5 Frode Informatica

Essa è composta da una manipolazione di:

- **Dati**, sia in input che in output.
- **Programmi**.
- **Hardware**.

Il risultato di queste manipolazioni sarà sempre un falso. Esempio:

Manipolazioni di dati: *input*

Esempio

Il funzionario di una banca modifica i dati relativi ai bonifici effettuati a favore dei clienti, aumentandone l'importo; provvede poi a stornare la somma in eccesso sul proprio conto corrente.

(Modifica di dati veri)

4.6 Evoluzione Normativa - Legge n°48 del 18/03/2008

4.6.1 Convenzione sulla criminalità informatica

È il primo trattato internazionale sulle infrazioni penali commesse via internet o su altre reti informatiche, ad esempio:

- **Violazione dei diritti d'autore.**
- **Frode Informatica.**
- **Pornografia Infantile.**
- **Violazione della sicurezza della rete.**

Contiene inoltre una serie di misure e procedure appropriate, quali la **perquisizione** dei sistemi di reti informatiche e l'**intercettazione** dei dati. Lo scopo principale è perseguire una politica penale comune per la protezione della società contro la cybercriminalità, in particolar modo adottando legislazioni appropriate e promuovendo la cooperazione internazionale.

4.6.2 Legge 48/2008

Dopo il consiglio d'europa del 2001, l'italia ne recepisce le direttive solamente nel 2008, introducendo la legge n°48 il 18/03/2008, questa definisce:

- **Danneggiamento Informatico**[C.P. Artt. 635-bis, 635-ter, 635-quater, 635-quinquies]:
 - Distinzione tra danneggiamento dell'integrità dei dati e il danneggiamento dell'integrità del sistema.
 - Distinzione a seconda che l'oggetto della tutela abbia, o meno, rilevanza a fini pubblicistici.
- **Ridefinizione di Documento Informatico**[Art. 491-bis C.P.]
- **Gestione della scena del crimine informatica**[C.P.P. Artt. 244, 247, 248, 254-bis, 256, 259, 260, 352, 353, 354].

Questa legge apporta alcune modifiche a determinati articoli correggendone il contenuto, ad esempio:

4.7 Differenza tra Ispezione e Perquisizione

4.7.1 Art. 244 C.P.P. - Casi e forme delle ispezioni

1. L'ispezione delle persone, dei luoghi e delle cose è disposta con decreto motivato quando occorre accertare le tracce e gli altri effetti materiali del reato.
2. Se il reato non ha lasciato tracce o effetti materiali, o se questi sono scomparsi o sono stati cancellati o dispersi, alterati o rimossi, l'autorità giudiziaria descrive lo stato attuale e, in quanto possibile, verifica quello preesistente, curando anche di individuare modo, tempo e cause delle eventuali modificazioni. L'autorità giudiziaria può disporre rilievi segnaletici, descrittivi e fotografici e 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.*

4.7.2 Art. 247 C.P.P. - Casi e forme delle perquisizioni

1. Quando vi è fondato motivo di ritenere che taluno occulti sulla persona il corpo del reato o cose pertinenti al reato, è disposta perquisizione personale. Quando vi è fondato motivo di ritenere che tali cose si trovino in un determinato luogo ovvero che in esso possa eseguirsi l'arresto dell'imputato o dell'evaso, è disposta perquisizione locale.
Ibis 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, ancorché 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.
2. La perquisizione è disposta con decreto motivato.
3. L'autorità giudiziaria può procedere personalmente ovvero disporre che l'atto sia compiuto da ufficiali di polizia giudiziaria delegati con lo stesso decreto.

4.7.3 Art. 259 C.P.P. - Custodia delle cose sequestrate

1. Le cose sequestrate sono affidate in custodia alla cancelleria o alla segreteria. Quando ciò non è possibile o non è opportuno, l'autorità giudiziaria dispone che la custodia avvenga in luogo diverso, determinandone il modo e nominando un altro custode, idoneo a norma dell'articolo 120.
2. All'atto della consegna, il custode è avvertito dell'obbligo di conservare e di presentare le cose a ogni richiesta dell'autorità giudiziaria nonché delle pene previste dalla legge penale per chi trasgredisce ai doveri della custodia. *Quando la custodia riguarda dati, informazioni o programmi informatici, il custode è altresì avvertito dell'obbligo di impedirne l'alterazione o l'accesso da parte di terzi, salva, in quest'ultimo caso, diversa disposizione dell'autorità giudiziaria.* Al custode può essere imposta una cauzione. Dell'avvenuta consegna, dell'avvertimento dato e della cauzione imposta è fatta menzione nel verbale. La cauzione è ricevuta, con separato verbale, nella cancelleria o nella segreteria.

4.7.4 Art. 260 C.P.P. - Apposizione di sigilli alle cose sequestrate. Cose deperibili. Distruzione di cose sequestrate.

1. Le cose sequestrate si assicurano con il sigillo dell'ufficio giudiziario e con le sottoscrizioni dell'autorità giudiziaria e dell'ausiliario che la assiste ovvero, in relazione alla natura delle cose, con altro mezzo, *anche di carattere elettronico o informatico*, idoneo a indicare il vincolo imposto a fini di giustizia.
2. 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'articolo 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.*

Con questo articolo si introduce la modalità in cui deve essere fatta una copia forense e alla quale poi dovrà essere posto un sigillo. Come ?

Durante la copia del sorgente viene applicato un algoritmo di cifratura, detto **HASH**, sia sulla copia che all'originale, in questo modo le due chiavi di cifratura risultano uguali garantendo così:

1. Che la copia è speculare all'originale.
2. Che se ci fossero alterazioni ai dati la cifratura non risulterebbe più la stessa ed il sigillo sarebbe violato e non più valido.

4.8 Altri articoli

4.8.1 Art. 248 C.P.P. - Richiesta di consegna

1. Se attraverso la perquisizione si ricerca una cosa determinata, l'autorità giudiziaria può invitare a consegnarla. Se la cosa è presentata, non si procede alla perquisizione, salvo che si ritenga utile procedervi per la completezza delle indagini .
2. Per rintracciare le cose da sottoporre a sequestro o per accertare altre circostanze utili ai fini delle indagini, l'autorità giudiziaria o gli ufficiali di polizia giudiziaria da questa delegati possono esaminare *atti, documenti e corrispondenza presso banche atti, documenti e corrispondenza nonché dati, informazioni e programmi informatici*. In caso di rifiuto, l'autorità giudiziaria procede a perquisizione.

4.8.2 Art. 254 C.P.P. - Sequestro di corrispondenza

1. *Presso coloro che forniscono servizi postali, telegrafici, telematici o di telecomunicazioni è consentito procedere al sequestro di lettere, pieghi, pacchi, valori, telegrammi e altri oggetti di corrispondenza, anche se inoltrati per via telematica, che l'autorità giudiziaria abbia fondato motivo di ritenere spediti dall'imputato o a lui diretti, anche sotto nome diverso o per mezzo di persona diversa o che comunque possono avere relazione con il reato.*
2. Quando al sequestro procede un ufficiale di polizia giudiziaria, questi deve consegnare all'autorità giudiziaria gli oggetti di corrispondenza sequestrati, senza aprirli *o alterarli* e senza prendere altrimenti conoscenza del loro contenuto.
3. Le carte e gli altri documenti sequestrati che non rientrano fra la corrispondenza sequestrabile sono immediatamente restituiti all'avente diritto e non possono comunque essere utilizzati.

4.8.3 Art. 254-bis C.P.P. - Sequestro di dati informatici presso fornitori di servizi informatici, telematici e di telecomunicazioni

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 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

4.8.4 Art. 256 C.P.P. - Dovere di esibizione e segreti

1. Le persone indicate negli articoli 200 e 201 devono consegnare immediatamente all'autorità giudiziaria, che ne faccia richiesta, gli atti e i documenti, anche in originale se così è ordinato, *nonché i dati, le informazioni e i programmi informatici, anche mediante copia di essi su adeguato supporto*, e ogni altra cosa esistente presso di esse per ragioni del loro ufficio, incarico, ministero, professione o arte, salvo che dichiarino per iscritto che si tratti di segreto di Stato ovvero di segreto inerente al loro ufficio o professione.

4.8.5 Art. 352 C.P.P. - Perquisizioni

- 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, ancorché 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.

4.8.6 Art. 353 C.P.P. - Acquisizione di plichi o di corrispondenza

1. Quando vi è necessità di acquisire plichi sigillati o altrimenti chiusi, l'ufficiale di polizia giudiziaria li trasmette intatti al pubblico ministero per l'eventuale sequestro.
2. Se ha fondato motivo di ritenere che i plichi contengano notizie utili alla ricerca e all'assicurazione di fonti di prova che potrebbero andare disperse a causa del ritardo, l'ufficiale di polizia giudiziaria informa col mezzo più rapido il pubblico ministero il quale può autorizzarne l'apertura immediata *e l'accertamento del contenuto*.
3. Se si tratta di lettere, pieghi, pacchi, valori, telegrammi o altri oggetti di corrispondenza, *anche se in forma elettronica o se inoltrati per via telematica*, per i quali è consentito il sequestro a norma dell'articolo 254, gli ufficiali di polizia giudiziaria, in caso di urgenza, ordinano a chi è preposto al servizio postale, *telegrafico, telematico o di telecomunicazione* di sospendere l'inoltro. Se entro quarantotto ore dall'ordine della polizia giudiziaria il pubblico ministero non dispone il sequestro, gli oggetti di corrispondenza sono inoltrati.

4.8.7 Art. 354 C.P.P. - Accertamenti urgenti sui luoghi, sulle cose e sulle persone. Sequestro.

1. Gli ufficiali e gli agenti di polizia giudiziaria curano che le tracce e le cose pertinenti al reato siano conservative e che lo stato dei luoghi e delle cose non venga mutato prima dell'intervento del pubblico ministero.
2. Se vi è pericolo che le cose, le tracce e i luoghi indicati nel comma 1 si alterino o si disperdano o comunque si modifichino e il pubblico ministero non può intervenire tempestivamente, ovvero non ha ancora assunto la direzione delle indagini, gli ufficiali di polizia giudiziaria compiono i necessari accertamenti e rilievi sullo stato dei luoghi e delle cose. *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.* Se del caso, sequestrano il corpo del reato e le cose a questo pertinenti.

4.8.8 Art. 420 C.P. - Attentato a impianti di pubblica utilità

- ▶ Chiunque commette un fatto diretto a danneggiare o distruggere impianti di pubblica utilità, è punito, salvo che il fatto costituisca più grave reato, con la reclusione da uno a quattro anni.
- ▶ *La pena di cui al primo comma si applica anche a chi commette un fatto diretto a danneggiare o distruggere sistemi informatici o telematici di pubblica utilità, ovvero dati, informazioni o programmi in essi contenuti o a essi pertinenti. (*)*
- ▶ *Se dal fatto deriva la distruzione o il danneggiamento dell'impianto o del sistema, dei dati, delle informazioni o dei programmi ovvero l'interruzione anche parziale del funzionamento dell'impianto o del sistema, la pena è della reclusione da tre a otto anni. (*)*

(*) abrogati con la legge n. 48 del 18/03/2008

4.8.9 Art. 491-bis C.P. - Documenti Informatici

- ▶ Se alcuna delle falsità previste dal presente capo riguarda un documento informatico pubblico [o privato] {avente efficacia probatoria}, si applicano le disposizioni del capo stesso concernenti [rispettivamente] agli atti pubblici [e le scritture private]
- ▶ A tal fine per documento informatico si intende qualunque supporto informatico contenente dati o informazioni aventi efficacia probatoria o programmi specificamente destinati ad elaborarli (*)

[] termini soppressi dal d.lgs. n. 7 del 15/01/2016

{ } termini aggiunti con la legge n. 48 del 18/03/2008

(*) abrogato con la legge n. 48 del 18/03/2008

4.8.10 Art. 495-bis C.P. - Falsa dichiarazione o attestazione al certificatore di firma elettronica sull'identità o su qualità personali proprie o di altri

- ▶ Se alcuna delle falsità previste dal presente capo riguarda un documento informatico pubblico [o privato] {avente efficacia probatoria}, si applicano le disposizioni del capo stesso concernenti [rispettivamente] agli atti pubblici [e le scritture private]
- ▶ A tal fine per documento informatico si intende qualunque supporto informatico contenente dati o informazioni aventi efficacia probatoria o programmi specificamente destinati ad elaborarli (*)

[I] termini soppressi dal d.lgs. n. 7 del 15/01/2016

{ } termini aggiunti con la legge n. 48 del 18/03/2008

() abrogato con la legge n. 48 del 18/03/2008*

4.8.11 Art. 615-quinquies C.P. - Diffusione di apparecchiature, dispositivi o programmi informatici diretti a danneggiare o interrompere un sistema informatico o telematico

- ▶ Chiunque diffonde, comunica o consegna un programma informatico da lui stesso o da altri redatto, avente per scopo o per effetto il danneggiamento di un sistema informatico o telematico, dei dati o dei programmi in esso contenuti o a esso pertinenti, ovvero l'interruzione, totale o parziale, o l'alterazione del suo funzionamento, è punito con la reclusione sino a due anni e con la multa sino a lire 20 milioni.

4.8.12 Art. 635-bis C.P. - Danneggiamento di informazioni, dati e programmi informatici

- ▶ Chiunque distrugge, deteriora o rende, in tutto o in parte, inservibili sistemi informatici o telematici altrui, ovvero programmi, informazioni o dati altrui, è punito, salvo che il fatto costituisca più grave reato, con la reclusione da sei mesi a tre anni.
- ▶ Se ricorre una o più delle circostanze di cui al secondo comma dell'art. 635, ovvero se il fatto è commesso con abuso della qualità di operatore del sistema, la pena è della reclusione da uno a quattro anni.

4.8.13 Art. 635-ter C.P. - Danneggiamento di informazioni, dati e programmi informatici utilizzato dallo Stato o da altro ente pubblico o comunque di pubblica utilità

- ▶ Salvo che il fatto costituisca più grave reato, chiunque commette un fatto diretto a distruggere, deteriorare, cancellare, alterare o sopprimere informazioni, dati o programmi informatici utilizzati dallo Stato o da altro ente pubblico o ad essi pertinenti, o comunque di pubblica utilità, è punito con la reclusione da uno a quattro anni.
- ▶ Se dal fatto deriva la distruzione, il deterioramento, la cancellazione, l'alterazione o la soppressione delle informazioni, dei dati o dei programmi informatici, la pena è della reclusione da tre a otto anni.
- ▶ Se ricorre la circostanza di cui al numero 1) del secondo comma dell'articolo 635 ovvero se il fatto è commesso con abuso della qualità di operatore del sistema, la pena è aumentata.

4.8.14 Art. 635-quater C.P. - Danneggiamento di sistemi informatici o telematici

- ▶ Salvo che il fatto costituisca più grave reato, chiunque, mediante le condotte di cui all'articolo 635-bis, ovvero attraverso l'introduzione o la trasmissione di dati, informazioni o programmi, distrugge, danneggia, rende, in tutto o in parte, inservibili sistemi informatici o telematici altrui o ne ostacola gravemente il funzionamento è punito con la reclusione da uno a cinque anni.
- ▶ Se ricorre la circostanza di cui al numero 1) del secondo comma dell'articolo 635 ovvero se il fatto è commesso con abuso della qualità di operatore del sistema, la pena è aumentata.

4.8.15 Art. 635-quinquies C.P. - Danneggiamento di sistemi informatici o telematici di pubblica utilità

- ▶ Se il fatto di cui all'articolo 635-quater è diretto a distruggere, danneggiare, rendere, in tutto o in parte, inservibili sistemi informatici o telematici di pubblica utilità o ad ostacolarne gravemente il funzionamento, la pena è della reclusione da uno a quattro anni.
- ▶ Se dal fatto deriva la distruzione o il danneggiamento del sistema informatico o telematico di pubblica utilità ovvero se questo è reso, in tutto o in parte, inservibile, la pena è della reclusione da tre a otto anni.
- ▶ Se ricorre la circostanza di cui al numero 1) del secondo comma dell'articolo 635 ovvero se il fatto è commesso con abuso della qualità di operatore del sistema, la pena è aumentata

4.8.16 Art. 640-quinquies C.P. - Frode informatici del soggetto che presenta servizi di certificazione di firma elettronica

- ▶ Il soggetto che presta servizi di certificazione di firma elettronica, il quale, al fine di procurare a sé o ad altri un ingiusto profitto ovvero di arrecare ad altri danno, viola gli obblighi previsti dalla legge per il rilascio di un certificato qualificato, è punito con la reclusione fino a tre anni e con la multa da 51 a 1.032 euro.

5 Lezione 5

5.1 Ricorda, cos'è la computer forensics?

Essa è l'insieme delle metodologie, scientificamente provate, finalizzate alla ricostruzione di eventi ai fini probatori che coinvolgono direttamente o indirettamente un supporto digitale.

5.2 Fasi del Trattamento - Identificazione e Raccolta.

5.2.1 Identificazione

Identificare il reperto che contiene dati che possono essere utili alla svolta delle indagini, quindi cerchiamo **dove** il dato è conservato. Vanno individuati tutti quei dispositivi che possono contenere dati rilevanti. Oltre ai dispositivi fisici che troviamo sulla scena del crimine, potrebbero esserci dati rilevanti all'interno di dispositivi remoti, attraverso il cloud magari, è quindi fondamentale rendersi conto di ciò e in caso riferire al **P.M.** che si vuole approfondire l'indagine cercando sul cloud/rete. Verrà poi disposto il sequestro per gli accessi al cloud e verrà poi fatta una copia di quei dati. Una volta individuati i possibili dispositivi di nostro interesse è bene capire se contengono davvero dati rilevanti **perquisendo** quei dispositivi, ma come farlo correttamente?

La così detta **Preview** è un'analisi di primo livello delle memorie dei dispositivi al fine di individuare possibili elementi di interesse investigativo, essa viene eseguita utilizzando dei **Write Blocker**, che possono essere sia di tipo hardware che di tipo software. Durante la perquisizione bisogna fare attenzione a non alterare i contenuti e perdere così delle possibili prove.

5.2.2 La Preview

Questa tipologia di analisi preliminare può essere effettuata in due modi:

- **DEAD**: questa è un'analisi eseguita a sistema operativo spento e viene fatto uso di un **Write Block**, quando questo è di tipo hardware viene montato il disco che si deve analizzare attraverso il Write Block, in modo tale da poter evitare accessi in scrittura su quel disco. Quando invece utilizziamo un Write Block di tipo software, perché magari non si ha la possibilità di smontare il disco perché saldato sulla macchina, possiamo fare uso di sistemi operativi, come **Caine**, che permettono di essere avviati **LIVE** tramite un CD/USB sulla macchina avente

quel disco che si vuole analizzare, il sistema operativo farà da Write Block evitando gli accessi in scrittura mentre si analizza il disco desiderato.

Quali sono i pro ed i contro di questa modalità?

- **PRO:** Inalterazione del dispositivo, in quanto spento, inoltre permette di utilizzare diversi strumenti per analizzare la memoria del dispositivo.
- **CONTRO:** Necessaria una buona conoscenza del sistema e dei sistemi software da analizzare. Non è poi sempre applicabile, ad esempio per i sistemi embedded.
- **LIVE:** questa invece è un'analisi eseguita impiegando il sistema operativo presente sul dispositivo da analizzare. Tutta la procedura deve essere documentata e verbalizzata.
Quali sono i pro ed i contro di questa modalità?
 - **PRO:** Si ha una visione dell'ambiente in cui opera l'utente e si ha maggiore velocità nell'analisi dei software installati.
 - **CONTRO:** Alterazione inevitabile del reperto. Inoltre bisognerebbe utilizzare strumenti adeguati per sfogliare i dischi del sistema in modo da non alterarne il contenuto.



5.2.3 Cambiamento di stato del dispositivo

Se il dispositivo che si deve analizzare è acceso ma si preferisce effettuare un'analisi di tipo **DEAD**, cosa si fa? Si spegne il dispositivo? O in caso contrario, lo si riaccende? Assolutamente no, ci sono alcune accortezze da fare, ovvero:

- **Shutdown** → se si vuole effettuare lo spegnimento del device, bisogna fare delle valutazioni prima, considerando la possibilità di eventuali cifrature ad esempio, oppure la possibilità che ci siano software in esecuzione e di conseguenza dei dump della RAM. Presi in considerazione questi fattori bisogna considerare le proprie necessità di analisi e scegliere tra:
 - Staccare la spina in modo da non alterare la memoria, perdendo però i dump della RAM e compromettendo il funzionamento del sistema.
 - Spegnimento attraverso il S.O in modo che i dump non vengano persi del tutto, ma in questo modo vengono eseguite diverse operazioni sul disco.
- **Accensione** → anche qui bisogna fare le dovute valutazioni, considerando se le informazioni che andremo a perdere sono meno importanti dell'urgenza di accertamento. Potremmo perdere l'ultimo accesso al sistema in questo modo, ed inoltre durante l'accensione vengono eseguite diverse operazioni sul disco.

5.2.4 La Raccolta

Dopo aver concluso quindi la fase di **perquisizione**, ed avendo un riscontro positivo dei dati ritrovati, dobbiamo procedere con la **raccolta** di queste **evidence** trovate, quindi attueremo una fase di **sequestro**.

5.2.5 Sequestro

La fase di **sequestro** può essere finalizzata in due modi:

- Sequestro **Fisico**: prendere quindi fisicamente il supporto su cui il dato interessato si trova. Durante il sequestro fisico bisogna preoccuparsi di identificare e verbalizzare tutti i reperti su un vero e proprio documento chiamato **catena di custodia**, che non è altro che un verbale che può essere redatto dalla polizia giudiziaria e nel quale devono essere specificate tutte le informazioni sui dispositivi sottoposti a sequestro. Cosa comprende questo documento?

- Luogo, data e operatore che ha reperito e collezionato la fonte di prova.
- Luogo, data e operatore che ha gestito e/o esaminato la fonte di prova.
- Coloro che hanno la possibilità di custodia delle varie **digital evidence**.
- Metodo di conservazione del reperto.
- Eventuali trasferimenti.

Non sempre il sequestro fisico è attuabile, nei casi in cui ci sono sistemi distribuiti o sistemi vitali ad esempio si ricorre al sequestro di tipo **logico**.

- Sequestro **Logico**: eseguire una copia totale o parziale del dispositivo. Durante il sequestro logico viene effettuata una duplicazione dei dati di possibile interesse e viene fatto attraverso la **Copia Forense**, che non è altro che una copia che ci dà la garanzia di ripetibilità dei successivi accertamenti che verranno eseguiti sulla **Copia Forense**.

Ricorda che **VALIDAZIONE + PRESERVAZIONE + COPIA**
⇒ COPIA FORENSE

Nota Bene: Qualora venisse trovato e copiato un singolo file costituente il motivo della perquisizione/sequestro, quel file non è sufficiente ad incolpare il proprietario del dispositivo, bensì c'è bisogno di un corredo di informazioni, come per esempio il come quel file sia arrivato su quel dispositivo, oppure se è stato condiviso altrove, gli accessi a quel file, ecc.

5.2.6 Acquisizione Fisica

Per effettuare una **Copia Forense** si può procedere effettuando la copia **bit a bit** dell'interno supporto di memoria permettendo di memorizzare dati e tutte le informazioni sulla gestione dei dati. Possiamo effettuare una copia completa di un supporto in due modi:

- **Clonazione**: questa ha come risultato un supporto pressoché identico all'originale. Viene utilizzata in casi particolari come quando bisogna analizzare il supporto reinserendolo nel proprio habitat.
- **File Immagine**: questa ha come risultato un file rappresentante il supporto originale. Risulta molto più maneggevole un singolo file e poi dal file immagine su può generare un disco clone. Posso inoltre manipolare più file immagine insieme.

5.2.7 Strumenti per la Copia Forense

Essi possono essere di tipo:

- **Hardware:** detti duplicatori forensi. Sono degli strumenti certificati, prestanti ma costosi, per questo molto spesso possono essere noleggiati.
- **Software:** sono delle distro linux LIVE forensi. Sono strumenti gratuiti ed opensource e sono molto versatili, bisogna però saperli utilizzare.

6 Lezione 6

6.1 Copia Forense

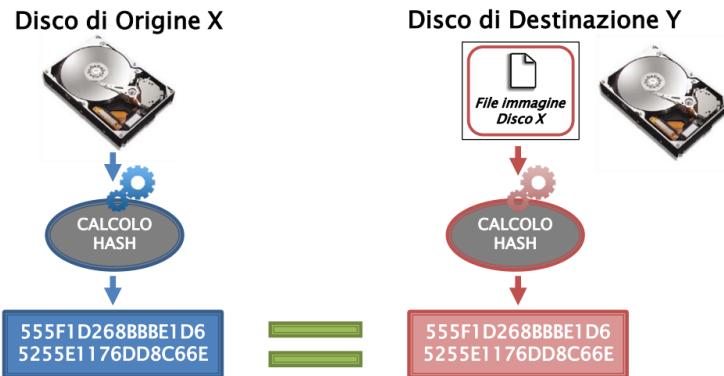
6.1.1 Hash

L'algoritmo di **HASH** prende in input un flusso di dati di dimensione qualsiasi e restituisce una stringa a lunghezza fissa di esadecimale. Questa stringa prodotta in output è univoca per ogni flusso di bit e funge da identificatori. Questo algoritmo risulta essere **non invertibile**, ovvero non possiamo ricostruire il dato a partire dalla stringa esadecimale.

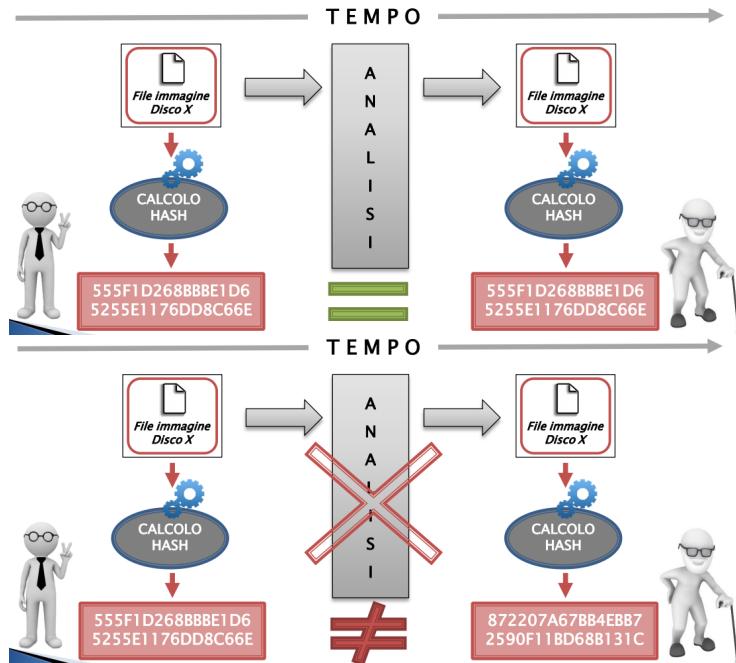
Nota Bene: sembrerebbe impossibile avere lo stesso hash per flussi di bit diversi, invece è una cosa che può accadere quando l'algoritmo non è troppo complesso.

Gli algoritmi di hash sono fondamentali in quanto ci forniscono:

- **Validazione**, ovvero ci garantiscono che la copia eseguita sia identica all'originale.

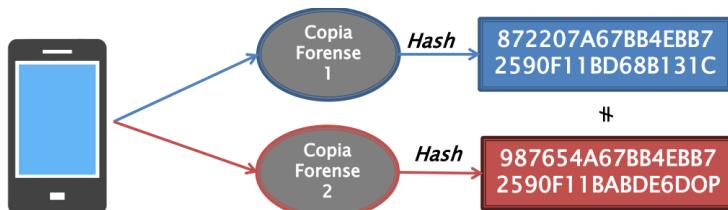


- **Preservazione**, ovvero abbiamo la garanzia che non vengano eseguite modifiche/alterazioni, se ciò dovesse accadere gli hash non combacerebbero più.



6.1.2 Accertamenti Ripetibili VS Irripetibili

- Per quanto riguarda gli accertamenti ripetibili abbiamo una o più copie dell'originale ed ognuna delle copie avrà sempre lo stesso hash in modo da validare le copie e rendere ripetibile il processo di analisi. I dischi copia sono sempre disponibili a differenza dell'originale. Possiamo applicare accertamenti ripetibili su supporti in buono stato.
- Per accertamenti di tipo irripetibile il supporto viene poi restituito, quindi non è sempre disponibile, oppure le memorie non sono in buono stato. Quando il supporto necessita di essere avviato per realizzare una copia forense (Live Acquisition) si ricorre all'[art. 360 C.P.P.](#) (accertamento irripetibile).



6.1.3 File di Log

Come visto durante l'introduzione agli strumenti per effettuare una copia forense, abbiamo visto un duplicatore forense creare un file immagine, spilitto in più file, il quale oltre al file immagine generava anche un file **file di log**. Questo file di log non è altro che un file descrittivo in cui sono riportate le informazioni sulla copia realizzata, come ad esempio:

- Informazioni sullo strumento impiegato per la copia → nome, versione, ...
 - Informazioni del disco origine → modello, capacità, numero di serie, ...
 - Informazioni dell'immagine forense → numero di file in cui è stata divisa, dimensioni, ...
 - Altre informazioni → data e ora, numero di settori saltati, ...
 - HASH → l'algoritmo usato per la cifratura [MD5, SHA1, SHA256, SHA512, ...]

```

*** Forensic Dossier -- Serial No.:78265 --
Software: V3.3.3RC16 Firmware: V1.14.2 fs:NTFS

Nome e Versione
dello strumento

***** Acquired by _____ Location _____
Acquired on _____ AT _____
***** SESSION SETTINGS *****
* Operating Mode: 4G E01:S2=>D2 Address Mode: LBA
* Verify : Hash-Dsk+v Speed :
* Connection : Direct

* E01 CAPTURE OF S2 HAS BEEN ACHIEVED.

***** SOURCE DRIVE(S) ***** DESTINATION DRIVE(S) *****
* S1 Model : ST380815AS Model : ST2000DM008-2FR102
Serial: 5RW2FPPX Serial: WFLC18EV
* C: 155000 H: 16 S: 63
Total Sectors Drive Size
156250000 74.0GB
* C: 3876021H: 16 S: 63
Total Sectors Drive Size
3907029168 1863.0GB

***** PC_OLI_E01: S1: 0 To:8667135
* start MD5: 67452301 EFCDBA89 98BADCFE 10325476
* end MD5: A18B0EE6 C7E7192A EEAE683F 88ADF742
* Verified : A18B0EE6 C7E7192A EEAE683F 88ADF742
* PC_OLI_E02: S1: 18667136 To:18759679
* start MD5: DEB75F20 10AA171F 9B05B385 AF4EEC01
* end MD5: DEB75F20 10AA171F 9B05B385 AF4EEC01
* Verified : DEB75F20 10AA171F 9B05B385 AF4EEC01
* PC_OLI_E03: S1: 18759680 To:27312127
* start MD5: DEB75F20 10AA171F 9B05B385 AF4EEC01
* end MD5: 46FA2899 B2528064 2BB2604D B9E6F5EF
* Verified : 46FA2899 B2528064 2BB2604D B9E6F5EF

Hash MD5
*** PC_OLI_E16: S1: 128294912 To:156249999
* start MD5: BBD79829 0F46CC3 296CBDB9 729804F2
* end MD5: 561B05BD 398160AB 2376C70F 383D744E
* Verified : 561B05BD 398160AB 2376C70F 383D744E
* S1 From: 0, To: 156249999, Size: 156250000
* Source MD5:
* ...EF184313 9669170C 593356DD A8849F1B...
* ...EF184313 9669170C 593356DD A8849F1B...
* Skipped Sectors: 0 Recovered Sectors: 0
* ****
* Compression Ratio is: 4.47 : 1
* Completion Time: 08/08/2008 08:08:00
* Audit Trail checksum: 077C3058 5E1293AB DD8BB848 43EE6E86

```

6.2 Comandi per eseguire la Copia Forense

6.2.1 Comando DD

```
DD(1)                               User Commands
NAME
dd - convert and copy a file
SYNOPSIS
dd [OPERAND]...
dd OPTION
```

Non si sa con precisione a cosa facesse riferimento la dicitura **DD**, ma il suo scopo è quello di copiare bit a bit il contenuto di un supporto su un altro supporto di archiviazione, leggendo sequenzialmente e poi copiando ogni bit. Nella maggior parte dei sistemi **Unix** tutto è visto come un file. Esiste una directory chiamata **/dev** dove sono raccolti tutti i dispositivi, essi sono visti tutti come file e possono essere di due tipi:

- **Character Device**: sono i dispositivi che trasmettono/trasferiscono dati [dispositivi audio, ...].
- **Block Device**: sono i dispositivi che memorizzano/raccolgono dati [hdd, sdd, ...].

Per eseguire il comando **DD** avviamo la nostra distro linux live, come ad esempio **caine**, che sia sul nostro personal computer oppure sul dispositivo con i supporti da copiare, ed una volta avviato senza montare i dischi target e il disco di destinazione, possiamo eseguire le nostre operazioni, come per esempio analizzare tutti i dischi collegati al sistema con il comando **fdisk -l**, il risultato sarà il seguente:

```
root@caine:/# fdisk -l

Disk /dev/sda: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72a3c36c

Device      Boot Start   End Sectors Size Id Type
/dev/sda1          2048 2099199 2097152   1G b W95 FAT32
/dev/sda2        2099200 8388607 6289408   3G b W95 FAT32

Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x9a847d68

Device      Boot Start   End Sectors Size Id Type
/dev/sdc1          2048 16777215 16775168   8G 7 HPFS/NTFS/exFAT
```

The screenshot shows the terminal output of the `fdisk -l` command. It lists two disks: `/dev/sda` (target disk) and `/dev/sdb` (destination disk). Disk `/dev/sda` is a 4GB disk with two partitions: `/dev/sda1` (FAT32, 1GB) and `/dev/sda2` (FAT32, 3GB). Disk `/dev/sdb` is a 20GB disk with one partition: `/dev/sdc1` (NTFS, 8GB).

Vediamo differenti informazioni riguardo i supporti agganciati. Tra i vari device che troviamo in `/dev` abbiamo il nostro **disco target** ovvero `/dev/sda`, esso rappresenta un hard disk da 4GB ed in questo caso è diviso in due partizioni chiamate rispettivamente `/dev/sda1` ed `/dev/sda2`. Dato che sono riportate informazioni su tutti i supporti agganciati, avremo quindi anche il nostro **disco di destinazione** rappresentato da `/dev/sdc` che è un disco da 8GB composto da un'unica partizione `/dev/sdc1`, ora sappiamo cosa fare, ovvero copiare `/dev/sda` in `/dev/sdc`, ma come lo facciamo ? Dobbiamo per prima cosa preparare il nostro disco di destinazione, che in questo momento risulta solo agganciato e non montato, affinché possa essere eseguita la copia forense. Utilizziamo una serie di comandi per effettuare questa operazione.

```
root@caine:/# mkdir /mnt/dest
root@caine:/# mount /dev/sdc1 /mnt/dest/
root@caine:/# mkdir /mnt/dest/dd_image
```

Sempre grazie all'utilizzo di **caine** possiamo montare il disco di destinazione. Sotto la directory **/mnt** vengono montati tutti i supporti. Con il comando **mkdir** creo una cartella di nome **dest** all'interno di **/mnt**. Qui è dove monteremo il disco, quello da 8GB per intenderci, per effettuarci poi la copia. Con il secondo comando, ovvero **mount** indico al sistema di montare il supporto **/dev/sdc1** che sarebbe il nostro disco di destinazione, nella cartella che ho creato poco fa. Con la terza istruzione invece vado a creare, dove ho appena montato il supporto da 8GB, un'ulteriore cartella **/dd_image**. Non ci resta che eseguire la copia forense vera e propria con il comando **dd**:

```
root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.dd bs=2048 conv=noerror,sync
```

Le varie istruzioni in questo comando stanno ad indicare:

- **if [input file]** → il disco sorgente che vogliamo copiare, nel nostro caso **/sda**.
- **of [output file]** → il file immagine, che sarà il nostro output, in questo caso **sda.dd**, che troveremo in **/mnt/dest/dd_image**.
- **bs [block size]** → espresso in byte (default 512), rappresenta la dimensione del blocco di lettura.
- **conv** → esegue l'elaborazione in base ai parametri indicati, in questo caso:
 - **noerror** → continua ad elaborare la copia anche se ci sono errori di lettura.
 - **sync** → sostituisce i blocchi di memoria non letti nella destinazione con *NULLS*, in questo modo da avere le dimensioni dei due dischi sincronizzate.

Dopo aver lanciato il comando possiamo dare un occhio al risultato di questo:

```
root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.dd bs=2048 conv=noerror,sync
2097152+0 records in
2097152+0 records out
4294967296 bytes (4,3 GB, 4,0 GiB) copied, 302,094 s, 14,2 MB/s
```

Vengono riportati i blocchi letti e quelli scritti, inoltre la dimensione dei dati riportati. Ora se provassimo a listare i file nella cartella

`/mnt/dest/dd_image` troveremo nient'altro che il file `sda.dd`, ovvero il nostro file immagine. Non abbiamo ancora la nostra copia forense però, bisogna prima validare e poi preservare il file di copia.

```
root@caine:/# ls -l /mnt/dest/dd_image/
total 4194304
-rwxrwxrwx 1 root root 4294967296 apr  7 23:26 sda.dd
```

Altri parametri utili per il comando `dd` potrebbero essere:

- **skip = [n]**: permette di saltare la lettura dei primi *n* blocchi di memoria.
- **count = [n]**: indica di terminare l'elaborazione dopo aver letto *n* blocchi.

Alcuni dei casi d'uso per questi parametri possono essere ad esempio quando vogliamo acquisire la copia di una singola partizione invece che del disco intero, procedo come segue:

```
Disk /dev/sda: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72a3c36c
Device     Boot   Start   End Sectors Size Id Type
/dev/sda1        2048 2099199 2097152    1G  b W95 FAT32
/dev/sda2      2099200 8388607 6289408    3G  b W95 FAT32

root@caine:/# dd if=/dev/sda2 of=/mnt/dest/dd_image/sda_p2.dd bs=2048
572352+0 records in
1572352+0 records out
3220176896 bytes (3,2 GB, 3,0 GiB) copied, 238,845 s, 13,5 MB/s

root@caine:/# ls -l /mnt/dest/dd_image/
total 3144704
-rwxrwxrwx 1 root root 3220176896 apr  7 23:36 sda_p2.dd
```

Eseguo gli stessi passaggi di prima solo che specifico di eseguire la copia solo della partizione `sda2`, in modo da avere un file immagine solo di quella partizione. In altri casi invece è utile adoperare questi parametri quando la partizione non viene riconosciuta ed utilizzando `skip` e `count` specifico i settori interessati dalla partizione `sda2` della quale conosciamo inizio e fine grazie al comando `fdisk -l`.

```

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda_p2.dd skip=2099199 count=6289408
6289408+0 records in
6289408+0 records out
3220176896 bytes (3,2 GB, 3,0 GiB) copied, 764,928 s, 4,2 MB/s

root@caine:/# ls -l /mnt/dest/dd_image/
total 3144704
-rwxrwxrwx 1 root root 3220176896 apr  7 23:55 sda_p2.dd

```

Possiamo utilizzare questi due parametri anche per suddividere un file immagine in più file, per esempio sezionando in più file ognuno da 1GB:

```

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.000 bs=1024 count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1,0 GB, 977 MiB) copied, 200,268 s, 5,1 MB/s

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.001 bs=1024 skip=1000000
count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1,0 GB, 977 MiB) copied, 226,651 s, 4,5 MB/s

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.002 bs=1024 skip=2000000
count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1,0 GB, 977 MiB) copied, 213,783 s, 4,8 MB/s

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.003 bs=1024 skip=3000000
count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1,0 GB, 977 MiB) copied, 220,863 s, 4,6 MB/s

root@caine:/# dd if=/dev/sda of=/mnt/dest/dd_image/sda.004 bs=1024 skip=4000000
194304+0 records in
194304+0 records out
198967296 bytes (194,3 MB, 185 MiB) copied, 220,863 s, 3,7 MB/s

root@caine:/# ls -l /mnt/dest/dd_image/
total 4194304
-rwxrwxrwx 1 root root 1024000000 apr  8 00:03 sda.000
-rwxrwxrwx 1 root root 1024000000 apr  8 00:04 sda.001
-rwxrwxrwx 1 root root 1024000000 apr  8 00:04 sda.002
-rwxrwxrwx 1 root root 1024000000 apr  8 00:05 sda.003
-rwxrwxrwx 1 root root 198967296 apr  8 00:06 sda.004

```

Con il ripetersi dello stesso comando più volte riusciamo ad arrivare alla suddivisione completa del supporto in diversi file più maneggevoli e leggeri. Diciamo che però dover ripetere lo stesso comando svariate volte è oneroso, viene introdotto così il comando **split** che svolge la suddivisione in un singolo comando.

```
root@caine:/# dd if=/dev/sda bs=2048 | split -d -b 2G - mnt/dest/dd_image/sda.
2097152+0 records in
2097152+0 records out
4294967296 bytes (4,3 GB, 4,0 GiB) copied, 157,836 s, 27,2 MB/s
root@caine:/# ls -l /mnt/dest/dd_image/
total 4194304
-rwxrwxrwx 1 root root 2147483648 apr  8 00:12 sda.00
-rwxrwxrwx 1 root root 2147483648 apr  8 00:13 sda.01
```

Dove i parametri:

- **-d** → indica di “appendere” al nome del file un contatore, in questo modo ogni file avrà lo stesso nome ma cambierà solo per un numero.
- **-b** → specifica la dimensione massima di ogni file, in questo caso 2GB.

6.2.2 Patch del comando DD

```
root@caine:/# dc3dd if=/dev/sda ofs=/mnt/dest/dd_image/sda.000 ofsz=2G bufsz=2k hash=md5
hash=sha256 log=/mnt/dest/dd_image/sda.log verb=on
```

Il vecchio comando **dd** è ormai stato sostituito dal nuovo comando **dc3dd** che ne rappresenta la nuova versione, fornendo queste tipologie di parametri:

- **ofs** → output diviso in più file, iniziando dal file immagine **sda.000**.
- **ofsz** → dimensione massima di ogni file, in questo caso 2 GB.
- **bufsz** o **bs** → block size in byte (default 512), in questo caso la dimensione del blocco di lettura è 2048 byte.
- **hash** → calcola l'hash con l'algoritmo indicato (MD5—SHA1—SHA256—SHA512), in questo caso sono stati scelti sia *MD5* che *SHA256*.
- **log** → salva il report dell'elaborazione in un file, in questo caso nel file **sda.log**
- **verb** → indica di generare un report dettagliato, si può decidere di attivare (*verb=on*) oppure disattivare (*verb=off*).

- **rec** → quando settato ad *off* interrompe l'esecuzione in caso di un errore di lettura.
- **hofs** → output diviso in molteplici file e per ognuno viene calcolato l'hash.

6.3 Calcolare l'Hash

6.3.1 Metodo n°1

Calcolare l'hash di un supporto è facile, in questo primo caso useremo **MD5**, come algoritmo, per calcolare l'hash del disco sorgente **sda** ed infine memorizzarlo in un file che chiameremo **sda.orig.hash**.

```
root@caine:/# md5sum /dev/sda > /mnt/dest/dd_image/sda.log
root@caine:/# cat /mnt/dest/dd_image/sda.log
d7a09df1018710f2b40744ba62445c7b  /dev/sda
```

Con il primo comando, **md5sum** calcolo l'hash e redirigo l'output sul file **sda.orig.hash**. Dopo aver fatto ciò si usa il comando **cat** per mostrare a schermo il contenuto del file in cui abbiamo inserite l'hash.

Una volta fatto questo calcoliamo allo stesso modo l'hash per il file immagine che abbiamo creato prima, inserendo il risultato dell'hash in un altro file di testo chiamato **sda.dd.hash**:

```
root@caine:/# md5sum /mnt/dest/dd_image/sda.dd >> /mnt/dest/dd_image/sda.log
root@caine:/# cat /mnt/dest/dd_image/sda.log
d7a09df1018710f2b40744ba62445c7b  /dev/sda
d7a09df1018710f2b40744ba62445c7b  /mnt/dest/dd_image/sda.dd
```

Come procediamo se invece abbiamo avuto la necessità di splittare il file immagine in diversi file ?

```
root@caine:/# md5 /mnt/dest/dd_image/sda.* >> /mnt/dest/dd_image/sda.log
root@caine:/# cat /mnt/dest/dd_image/sda.log
d7a09df1018710f2b40744ba62445c7b  /dev/sda
025c062e137800811d6339759360f4c1  /mnt/dest/dd_image/sda.00
9faa25372a0b7176cd889cbf51e85bf5  /mnt/dest/dd_image/sda.01
```

Attraverso l'uso del comando **cat** redirigo il contenuto di tutti i file, che ovviamente saranno splittati, quindi useremo la dicitura **sda.***, con la pipeline, verso il comando **md5sum** che metterà il risultato dell'hash in un file di testo, avendo così lo stesso risultato.

6.3.2 Metodo n°2

Con questo metodo abbiamo la possibilità di calcolare l'hash *durante* l'elaborazione della copia:

```
root@caine:/# dd if=/dev/sda bs=2048 | tee /mnt/dest/dd_image/sda.dd |  
md5sum > /mnt/dest/dd_image/sda.log
```

Usando il comando `dd`, come al solito, leggiamo dal disco sorgente e mettiamo l'output, ovvero ciò che è stato letto sul disco, in pasto al comando `tee` grazie alla pipeline, al contempo lo diamo in pasto al comando `md5sum` per il calcolo dell'hash. Il comando `tee` ci permette di biforcire lo stream che sarà diviso tra la creazione del file immagine e la cifratura dello stream.

7 Lezione 7

7.1 Disk Image

Abbiamo visto che una copia forense può essere acquisita fisicamente in due modi:

- clonando il disco su un altro supporto
- oppure creando un file immagine, detto **disk image**.

In entrambi i casi la copia risulterà essere bit a bit, comprendendo quindi tutto, anche informazioni sulla gestione dei dati.

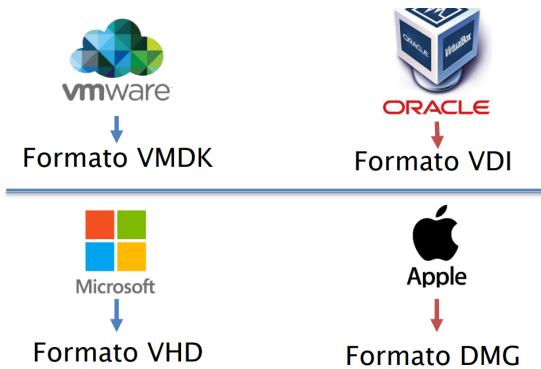
Il **disk image** nasce negli anni 60, in ambito aziendale indicava il **backup** o un **disaster recovery**. In ambito consumer invece venne introdotto per la duplicazione di supporti ottici e per eseguire backup, facilitare la masterizzazione e per la diffusione di software ed utility.

7.1.1 Supporti Ottici

Il formato **.ISO** è quello più comune, esso rappresentava il file system dei supporti ottici ed era definito nella **ISO 9660**. Questo formato era quello più comune, da esso nacque una sua evoluzione che prevedeva la suddivisione in due file, un file **.BIN** che rappresentava la copia grezza binaria, ed un file **.CUE** dove erano salvati i metadati della copia *raw*.

7.1.2 Dischi Virtuali

Un'ulteriore evoluzione avvenne con i **dischi virtuali**, finita l'era dei supporti fisici e dei disk image, le aziende iniziarono quindi a sviluppare i loro formati proprietari.



7.1.3 Formato DD/Raw

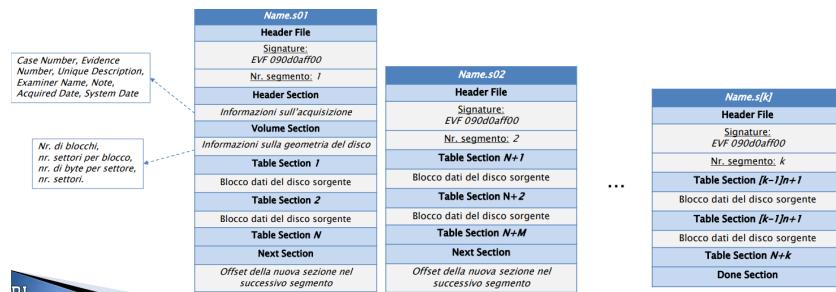
Il formato che abbiamo già visto e che utilizzeremo noi è un formato grezzo e non contiene molte informazioni utili. Il formato **.dd** è un container dello stream ma esso non conserva i metadati dell'evidenze come ad esempio il **modello**, **seriale**, **dimensioni**, ecc, ne tanto meno conserva gli **hash** calcolati, non esegue compressioni e non può contenere più di un file/stream.

Viene creato un disk image che ci permette di migliorare la nostra copia forense, ovvero...

7.1.4 Expert Witness Disk Image Format - (E.W.F)

In questa nuova famiglia di formati viene introdotta di default la possibilità di spartire il file immagine in sezioni, la compressione può essere applicata alla copia che stiamo realizzando ed infine l'immagine può essere segmentata. Uno dei formati **E.W.F.** è il formato **SMART** il cui obiettivo è quello di avere un accesso veloce ad una parte dell'immagine, ovvero si può avere accesso ai singoli segmenti del file spartito senza doverlo caricare per intero. Ogni segmento è così formato:

- **Header File:** Signature e nr° di segmento.
- Una o più sezioni, ce ne sono quattro tipi:
 - Header Section
 - Volume Section
 - Table Section
 - Next/Done Section



Un altro formato della famiglia **E.W.F.** è il formato **Encase E01 Bitstream**, questo è basato sul formato **SMART**, ha la possibilità di segmentare l'immagine ed in più ha ben 3 livelli di compressione, che sono:

- **NO.**
- **GOOD.**
- **BEST.**

Infine questo formato impiega 13 sezioni, 9 in più rispetto a **SMART**, il che lo rende molto più strutturato e dettagliato come formato.

► **Impiega nr. 13 sezioni (+ 9 al formato SMART):**

- Header2 section;
- Errors2 section;
- Disk section;
- Session section;
- Sectors section;
- Hash section;
- Table2 section;
- Digest section;
- Data section;

Un altro formato della stessa famiglia è **Encase L01 Logical** che intraprende una rivoluzione, ovvero l'acquisizione dei file logici, in pratica permette di acquisire una cartella e tutti i suoi file all'interno con un singolo comando. Questo formato permette come sempre la segmentazione dell'immagine ed infine, a differenza del formato **E01** qui abbiamo 15 sezioni, queste due sezioni in più sono:

- **Ltree Section.**
- **Ltypes Section.**

Entrambe le due nuove sezioni forniscono informazioni riguardo il percorso di acquisizione del file logico.

Prima della realizzazione della libreria **LIBEWF**, venne creato un ulteriore formato open ed estendibile detto **AFF/AFF4** dove ogni disco veniva separato in due layer che erano:

- **Disk-Rappresentation Layer** (Metadato).
- **Data-Storage Layer** (Dato).



Il metadato viene salvato come file **XML**.

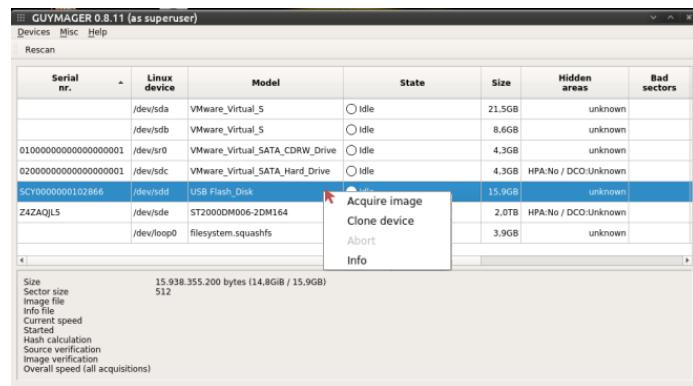
7.2 Software di Acquisizione

7.2.1 Guymager

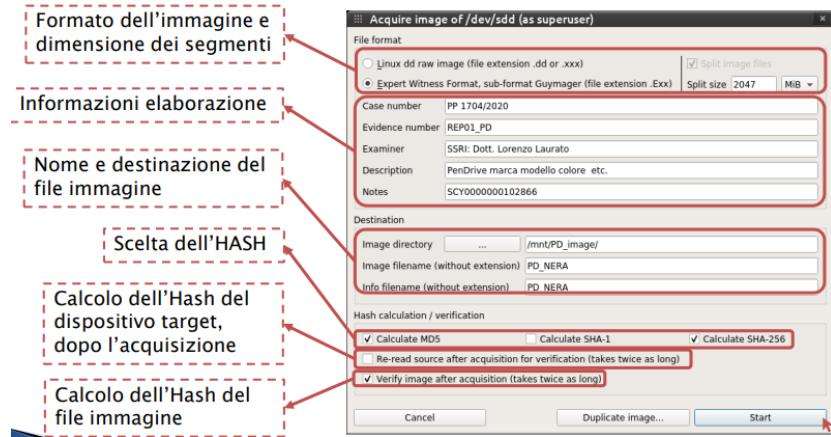
Questo è un software free ed opensource sviluppato per linux e che si basa sulla libreria **LIBEWF**, questo software permette due tipologie di acquisizione:

- Clone della memoria bit a bit.
- Disk Image, ovvero la produzione di un file immagine del disco.

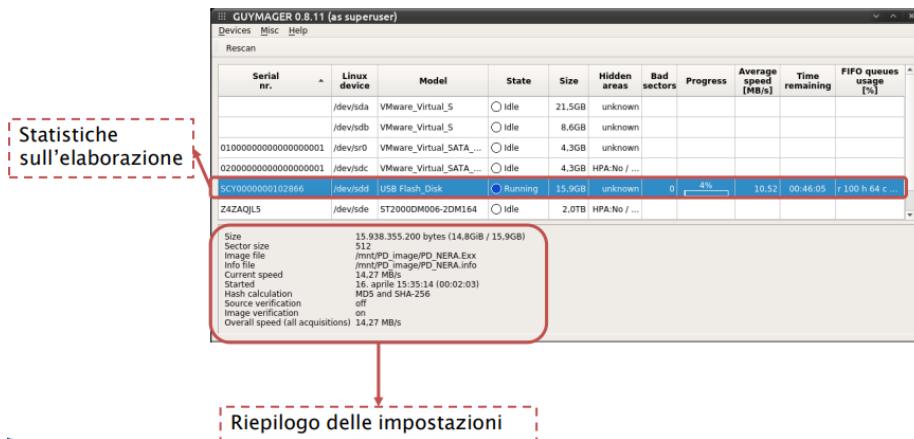
Guymager permette esclusivamente l'acquisizione *full disk*.



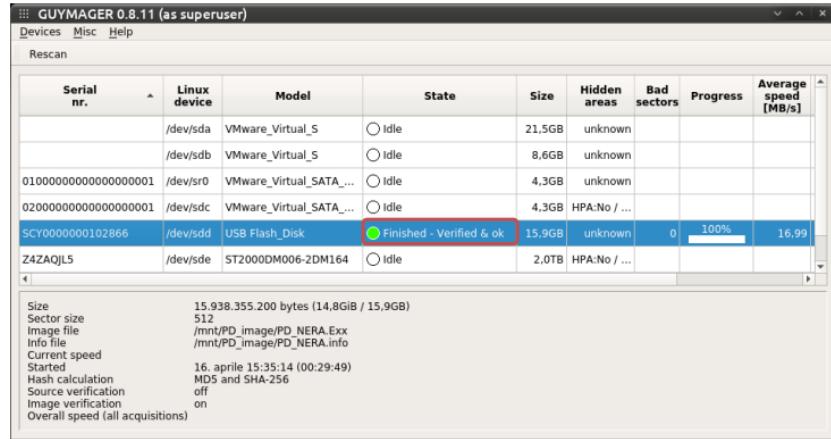
La GUI principale rappresenta l'elenco dei possibili dispositivi da cui effettuare un acquisizione, equivalente ad `fdisk -l` in pratica. Supponiamo di voler acquisire il device **sdd**, con il tasto destro del mouse si può scegliere se fare una *clonazione* o un disk image, procediamo in questo caso con il *disk image*.



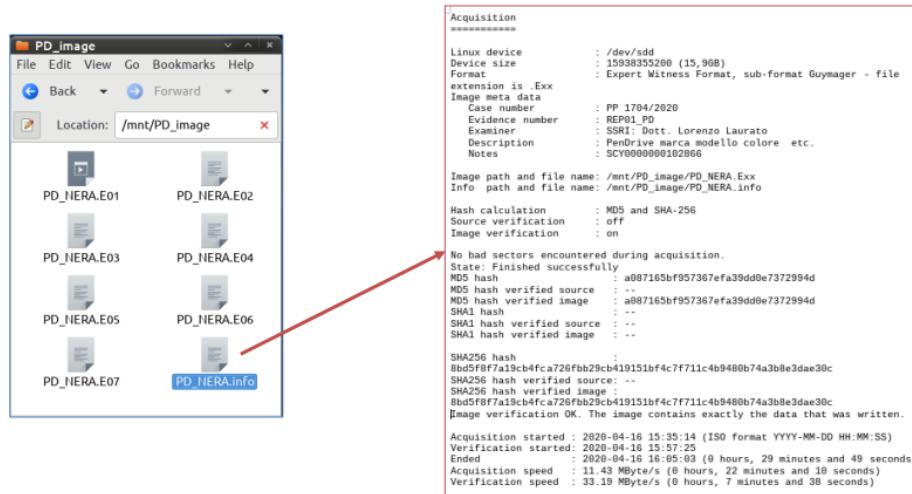
La finestra che verrà aperta una volta selezionata la tipologia di acquisizione sarà quella predisposta al settaggio dei parametri per l'elaborazione.



Una volta avviata l'elaborazione viene mostrata una schermata con gli avanzamenti dell'elaborazione e qualche statistica sul processo, insieme ad un riepilogo dei parametri.



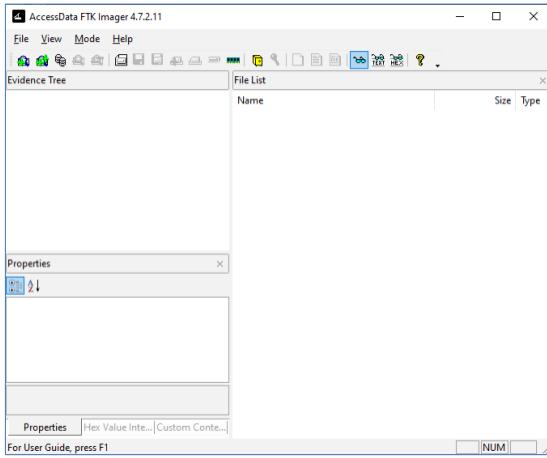
Possiamo infine vedere, ad elaborazione terminata, un indicatore che ci indica che l'elaborazione è conclusa.



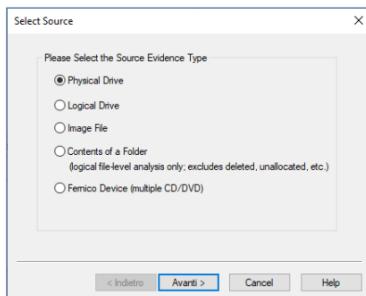
Una volta terminato il tutto possiamo recarci nella cartella di destinazione del disk image dove troveremo i 7 segmenti del file splitto ed un file di log molto dettagliato contenente i dati dell'elaborazione.

7.2.2 FTK Imager

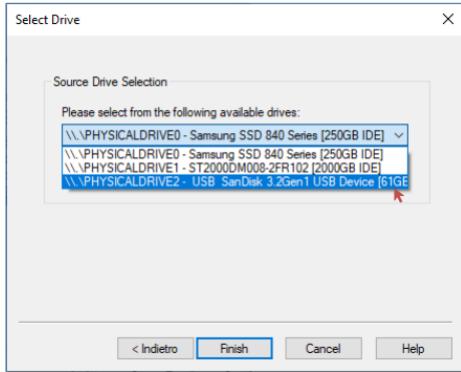
Software con licenza freeware per piattaforme Windows e ci sono due versioni del prodotto, una **Lite Version** ed una **Portable Version**, quest'ultima è possibile utilizzarla senza bisogno di installazione, è quindi possibile scaricarla su una USB e utilizzarla dove di vuole, oppure in modo più comune si può scegliere la **Install Version** scaricabile sulla propria macchina. **FTK Imager** risulta utile anche per le perquisizioni informatiche, offrendo la possibilità di analizzare un disco. Vediamo nel dettaglio cosa offre l'interfaccia di questo tool.



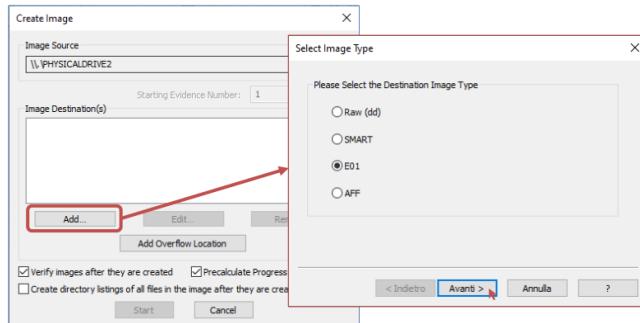
Nella schermata home dell'applicativo possiamo trovare diverse voci, per iniziare una nuova acquisizione ci rechiamo su *File* in alto a sinistra e poi sulla voce *Create Disk Image*.



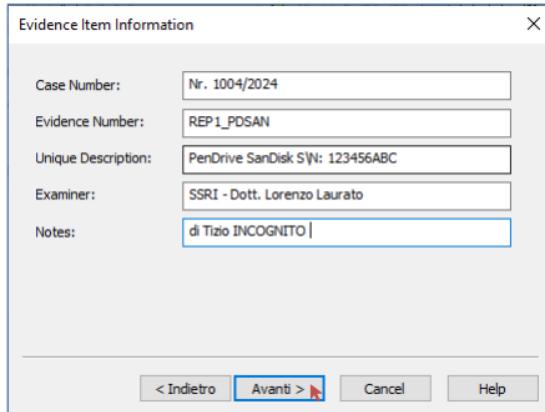
Possiamo notare come ci siano molte più opzione rispetto a Guy-imager per la scelta della tipologia di acquisizione.



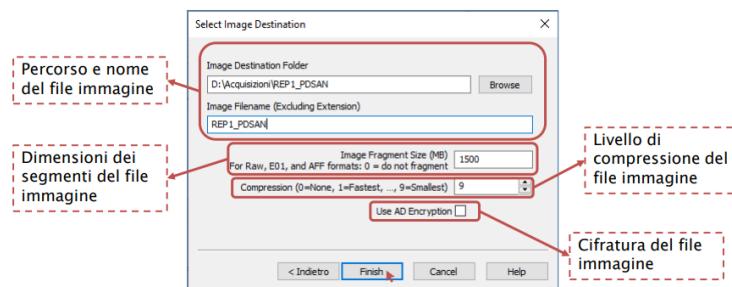
Una volta scelta la tipologia di acquisizione, nel nostro caso abbiamo scelto **Physical Drive** possiamo passare alla scelta del disco da acquisire, in questo caso è stato scelto il disco denominato **PHYSICALDRIVE2** una pendrive da 61GB.



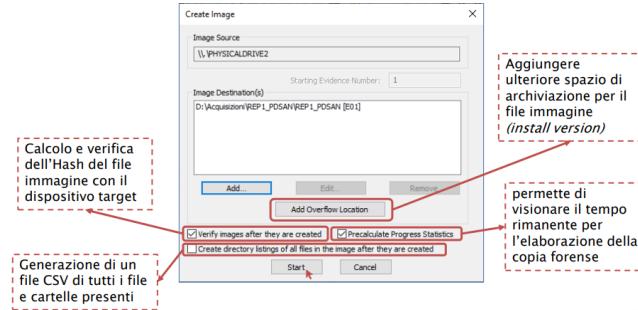
Selezionato il disco sorgente andiamo a settare i parametri per l'e-laborazione, il primo parametro sarà proprio il tipo di formato del file immagine, scegliamo il formato **E01**.



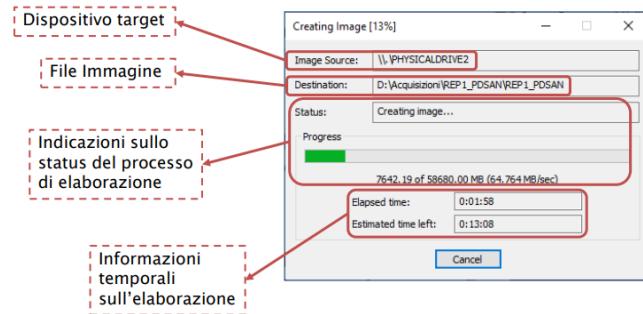
Inseriamo tutte le informazioni rilevanti del caso, in questo modo sarà più facile lasciare traccia di ciò che si sta facendo.



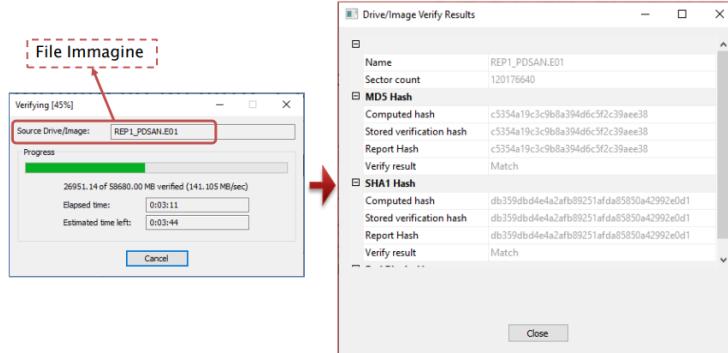
Nella schermata successiva ci viene chiesto di selezionare la cartella in cui salvare il disk image, il nome che gli vogliamo dare ed altri parametri utili.



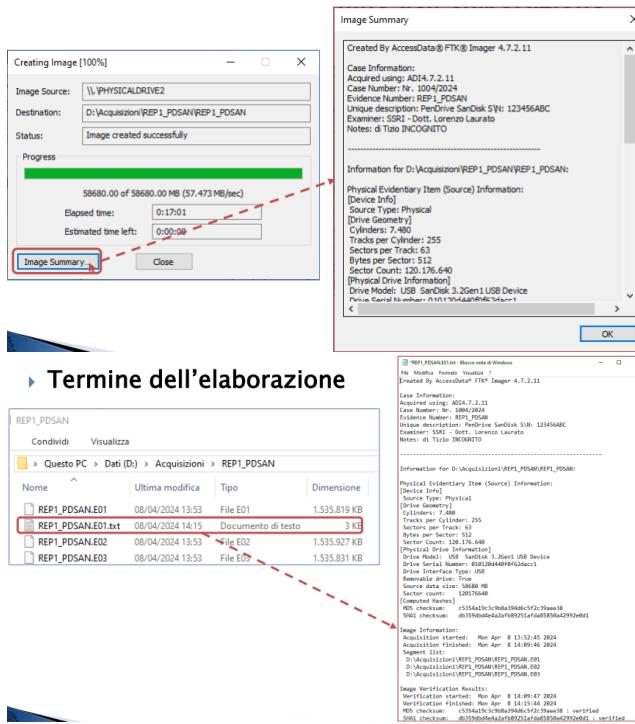
Informazioni aggiuntive possono essere specificate in questa pagina, per esempio, in caso di overflow della prima destinazione, si può decidere una destinazione differente.



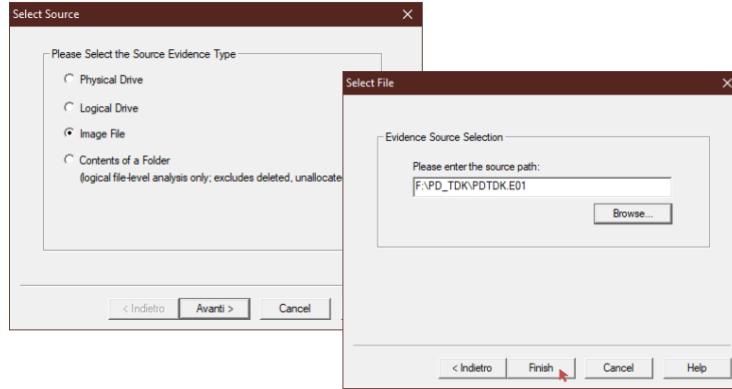
Qui possiamo osservare l'avanzamento dell'elaborazione e varie statistiche di quest'ultima.



Dopo aver terminato la fase di validazione il nostro tool ci mostrerà gli hash del **Computed** e del **Report** ed inoltre lo **Store Hash**, in quanto con il formato **E01** l'hash viene salvato nell'header del disk image.



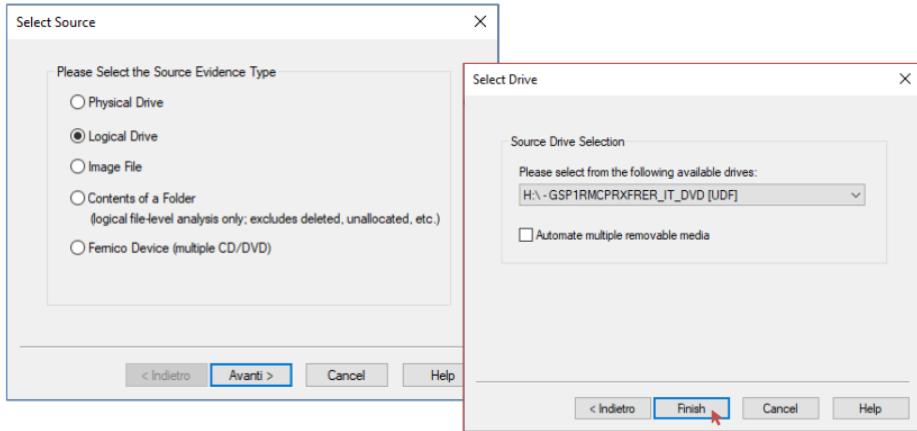
Una volta terminata l'elaborazione si può accedere ad una preview del file di log.



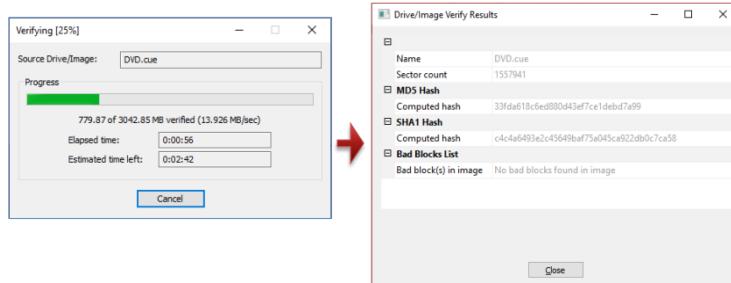
In FTK Imager, selezionando invece, *File* e poi *Add Evidence Item* possiamo andare ad analizzare i file immagine che abbiamo generato. Una volta aperto il disk image avremo diverse informazioni.

Evidence Source Path	
Evidence Type	Forensic Disk Image
Disk	
Verification Hashes	
MD5 verification hash	c5354a19c3c9b8a394d6c5f2c39aee38
SHA1 verification hash	db359dbd4e4a2afb89251afda85850a42992e0d1
Drive Geometry	
Bytes per Sector	512
Sector Count	120,176,640
Image	
Image Type	E01
Case number	Nr. 1004/2024
Evidence number	REP1_PDSAN
Examiner	SSRI - Dott. Lorenzo Laurato
Notes	di Tizio INCOGNITO
Acquired on OS	Win201x
Acquired using	AD4:7.2.11
Acquire date	08/04/2024 11:52:45
System date	08/04/2024 11:52:45
Unique description	PenDrive SanDisk S:N: 123456ABC

Informazioni come ad esempio l'hash, oppure il sistema operativo. Tutto questo grazie al formato **E01**



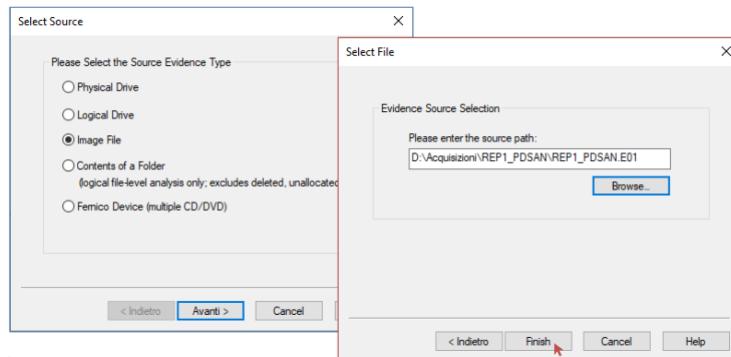
È inoltre possibile andare ad eseguire una copia sulle singole partizioni grazie ad **FTK Imager**, in più con l'opzione **Logical Drive** saremo in grado di vedere anche i supporti ottici. In questo caso sceglieremo un *DVD* come supporto.



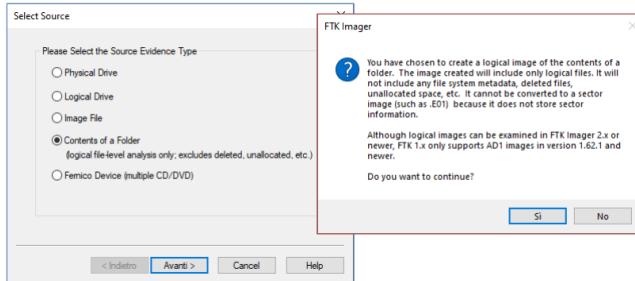
Una volta selezionato il supporto viene eseguita direttamente l'elaborazione, questo perché non è possibile aggiungere parametri o specifiche varie, di default viene scelto il formato **CUE/BIN** e viene fatta partire l'elaborazione.

The screenshot shows two windows from AccessData FTK Imager. On the left is a file browser window titled 'Nome' (Name) with columns for 'Ultima modifica' (Last modified), 'Tipo' (Type), and 'Dimensione' (Size). It lists several files: 'DVD.cue' (CUE Other File (VLC)), 'DVD.cue.txt' (Documento di testo), 'DVD.iso' (File immagine disco), 'DVD.iso01' (File ISO01), and 'DVD.iso02' (File ISO02). The 'DVD.cue.txt' file is selected. On the right is a detailed file information window titled 'Created By AccessData® FTK® Imager 4.7.2.11'. It provides technical details for the selected file, including the device name (H-DT-ST DVD+RW GH70N), media type (DVD-ROM), bytes per sector (2.848), session count (792), session count (1), UTC timestamps (True), and various file class, size, physical size, actual file, LBA, and logical block count values.

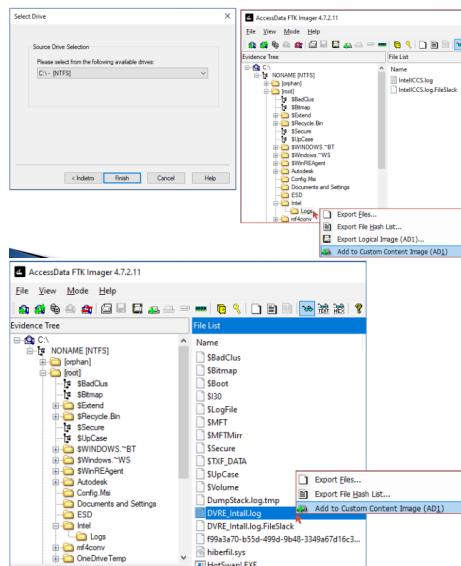
Ed ecco che ci ritroviamo con il file *.ISO* spartito ed il file *.CUE* dei metadati. Abbiamo come al solito un file di log di cui possiamo vedere la preview.



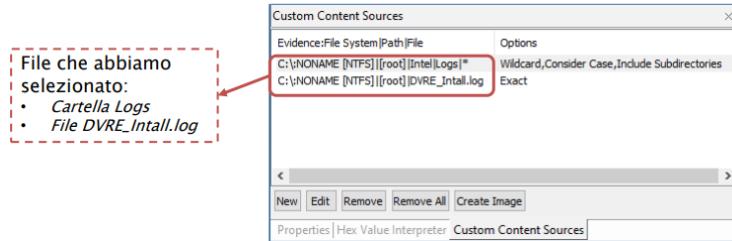
Tra le varie opzione è anche possibile utilizzare *Image File* per eseguire delle conversioni di formato per i file immagine. Qualora ad esempio il formato **E01** non fosse supportato ma il **DD/RAW** si, allora si può usufruire di questa opzione dando il file di input e avendo come un output un file con un formato diverso.



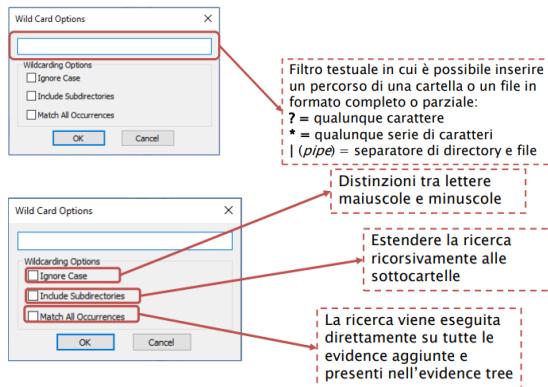
È possibile inoltre scegliere di acquisire un singolo file oppure una singola cartella al post dell'intero supporto di archiviazione. Scegliendo l'opzione *Contents of a Folder*. Di solito quando si cerca di acquisire un singolo file viene mostrato un messaggio di *alert* che avvisa l'utente che non saranno compresi ne i metadati, ne i file eliminati, ecc. In più avvisa l'utente che l'acquisizione verrà effettuata con il formato **AD1** che è un formato proprietario, questo perché **E01** viene utilizzato per le acquisizioni *Physical Drive*, ci sarebbe **L01** per le acquisizioni *Logical Drive* ma resta ancora un formato chiuso.



Quando si decide di acquisire una copia personalizzata, ad esempio scegliendo cartelle/file multipli che si trovano in path differenti, possiamo procedere con **File > Add Evidence Item** e poi selezionare il disco e sfogliando tutte le directory si possono scegliere i file/cartelle di interesse ed aggiungerli ad una **Custom Content Image**.



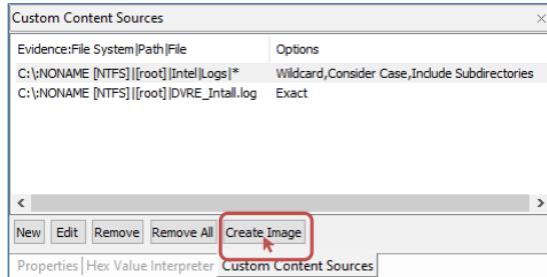
Una volta selezionati i file/cartelle di nostro interesse possiamo procedere con *Create Image*. Almeno che non vogliate soffermarvi sulle voci *New* oppure *Edit...*



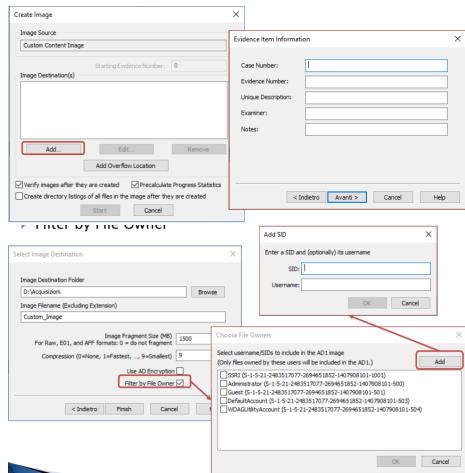
Cliccando su *Edit* possiamo specificare dei filtri di elaborazione della copia, ricercando ad esempio alcune informazioni all'interno del disco per analizzare solo determinati file, ad esempio:

- **Esempio 1** → Documents|*.doc? ⇒ questo filtro sta dicendo che stiamo cercando tutti i file Word (doc, docx) presenti in una qualunque cartella chiamata *Documents*.
- **Esempio 2** → |Microsoft|Outlook|*.pst? ⇒ questo filtro invece indica tutti gli archivi di posta elettronica Microsoft Outlook di tutti gli utenti.

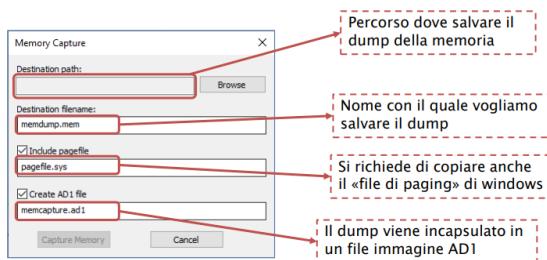
Cliccando invece *New* si crea da zero un nuovo filtro senza partire da un file selezionato precedentemente.



Possiamo quindi procedere al creare una Custom Image.



Prima di creare l'immagine però vengono chieste delle specifiche sull'evidence e poi ci sono delle opzioni di filtraggio, come ad esempio anche quella per utente.



FTK Imager ci permette di effettuare un'altra tipologia di acquisizione che è la *Capture Memory* che permette di effettuare un dump della RAM.

8 Lezione 8

8.1 Crittografia

La **Crittografia** è l'atto di rendere oscuro ciò che scrivi o vuoi comunicare. Essa proviene dalla **Crittologia** che non è altro che la scienza che si occupa della comunicazione in forma sicura e di solito segreta.

Crittografia	VS	Crittoanalisi
studio e applicazione dei principi e delle tecniche per rendere l'informazione inintelligibile a tutti tranne che al destinatario		scienza e arte di risolvere i crittosistemi per recuperare l'informazione nascosta

In antichità c'erano vari modi per comunicare in segreto ed essi venivano adoperati in vari campi, dall'uso militare e diplomatico a quello religioso. Tre **Tecniche di Cifratura a Sostituzione** erano:

- ▶ **Atbash:** alfabeto rovesciato (a->Z; b->Y; c->X; ... ; m->N)
- ▶ **Albam:** alfabeto diviso in due metà (a->N;b->O;c->P; ... ;m->Z)
- ▶ **Atbah:** relazione numerica fra le lettere (a=1; b=2; ...; z=26)
 - Prime 9 lettere (a-i): 10 - lettera => lettera sostituente;
 - Successive 9 lettere (j-s): 28 - lettera => lettera sostituente;
 - Ultime 8 lettere (t-z): 45 - lettera => lettera sostituente

Un cifrario a sostituzione noto e molto diffuso nell'antichità era il **Cifrario di Cesare** il quale per comunicare aggiungeva **+3** ad ogni lettera, quindi per esempio la lettera A cifrata diventava la lettera D, e così via fino a comporre le frasi cifrate desiderate. Ovviamente un cifrario del genere non ci volle molto a scoprirlo, infatti questa tipologia di cifrari semplici furono sostituiti poi dalle **Macchine Cifranti** tra le quali c'era **Enigma** che era un dispositivo elettromeccanico per cifrare e decifrare messaggi, venne usata durante la seconda guerra mondiale.

Possiamo affermare che nella storia della crittografia ci sono tre stadi principali:

- **Primo Stadio:** questo andava dalle prime civiltà fino al secolo scorso e che prevedeva algoritmi di cifratura sviluppati ed implementati a mano.
- **Secondo Stadio:** è incentrato per lo più durante la seconda guerra mondiale, ed è dove apparvero le prime macchine cifranti.

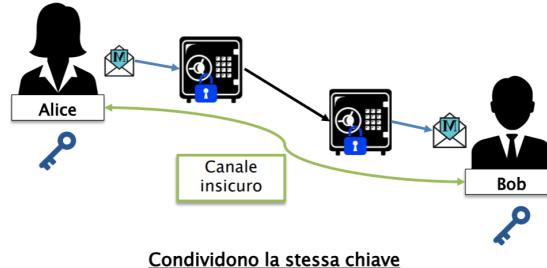
- **Terzo Stadio:** rientrano gli ultimi 50 anni e grazie all'avvento dei computer si sono sviluppati nuovi algoritmi di crittografia e nuove tecniche di crittoanalisi, di conseguenza si sono sviluppati anche nuovi campi d'azione.

8.2 Protocolli

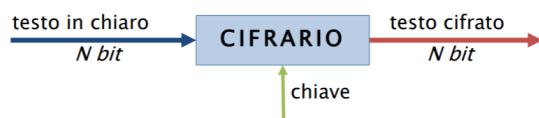
La crittografia è basata sui **protocolli**. Questi non sono altro che la definizione delle interazioni fra le parti comunicanti per ottenere le proprietà di sicurezza desiderate. I protocolli a loro volta si basano su una serie di protocolli più semplici detti **Primitive Crittografiche**, esse risolvono problemi più "facili" ed insieme vengono usate per risolvere problemi di natura complessa. Le primitive sono attese a risolvere problemi come:

- **Cifratura:** cifrari simmetrici o asimmetrici.
- **Autenticazione ed Integrità:** funzioni di hash o mac.
- **Firme Digitali.**
- **Altro:** generazione di numeri pseudo-casuali, ecc.

8.2.1 Cifrario Simmetrico

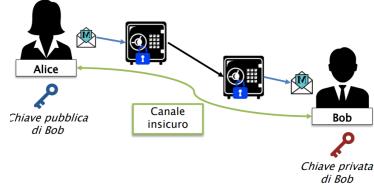


Due utenti condividono la stessa chiave, questa viene usata sia per cifrare che decifrare le informazioni. Questa tipologia di cifratura è ritenuta poco sicura, il canale di comunicazione potrebbe non essere privato e far passare la chiave in chiaro renderebbe vana la cifratura, inoltre gli interlocutori devono scambiarsi a priori la chiave.



Grazie alla chiave per il cifrario, il testo in chiaro, di lunghezza $N \text{ bit}$, viene cifrato e viene restituito un output cifrato ad $N \text{ bit}$, con la stessa lunghezza di input.

8.2.2 Cifrario Asimmetrico



Vengono impiegate due differenti chiavi, ogni *"utente"* ne ha una copia, una **chiave privata** che viene usata per la decifratura ed è conosciuta solo dal proprietario, ed una **chiave pubblica** che usa il mittente quando vuole dialogare con quello specifico utente, ovvero il proprietario della coppia di chiavi, e quest'ultima è conosciuta da chiunque.

8.2.3 Firma Digitale

La **Firma Digitale** non è altro che l'apposizione di una firma su un documento digitale. Essa deve essere facilmente prodotta dal legittimo firmatario, nessuno deve poter riprodurre la firma altrui, ed infine chiunque può facilmente verificare una firma. Tra gli algoritmi usati per le firme digitali abbiamo, **RSA** oppure **DSS**.

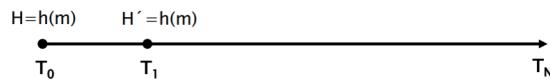
8.2.4 Funzioni di Hash

Grazie alle funzioni di **HASH** risolviamo il problema dell'integrità del dato. La definizione formale dice:

"Il valore hash $h(M)$ è una rappresentazione non ambigua e non falsificabile del messaggio M ."



Come abbiamo visto per le copie forensi, per sfruttare l'hash come validatore di integrità dei dati, ovvero per controllare che un dato sia sempre lo stesso nel tempo e non sia stato alterato, possiamo confrontare l'hash in tempi differenti:



Se H ed H' sono uguali in diversi istanti di tempo, allora il dato risulta essere inalterato.

La funzione di hash è così definita:

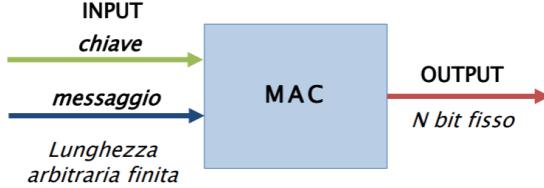
$$h : \Sigma^* \rightarrow \Sigma^n$$

È possibile, per due messaggi diversi m_1 ed m_2 , avere che $h(m_1) = h(m_2)$, ovvero posso trovare una collisione.



Di regola esistono infinite collisioni, in quanto il co-dominio è piccolo rispetto al dominio. Per questo sono stati sviluppati hash a 128, 256, 512 bit, in modo da aumentare i bit di output e di conseguenza aumentare anche la computazione però.

8.2.5 Funzione M.A.C.



Diversamente dalle funzioni di hash, le **funzioni M.A.C.** prendono in pasto anche una chiave di cifratura per la funzione, essa serve a garantire l'invio del messaggio.

8.3 Proprietà di Sicurezza

- **Confidenzialità:** si intende la protezione del dato da un soggetto estraneo. Spesso viene intesa come **privacy** e **secretezza** ma si fa sempre riferimento alla protezione delle informazioni da soggetti non autorizzati. Essa viene garantita dai sistemi di cifratura simmetrica ed asimmetrica.
- **Autenticazione:** è la certezza di identificare l'interlocutore, ovvero l'origine di un messaggio. Questo risulta possibile grazie agli algoritmi di firma digitale.
- **Integrità:** solo chi è autorizzato può modificare l'attività di un sistema o le informazioni trasmesse. La differenza con la **confidenzialità** è che in alcuni casi, e con alcuni sistemi di cifratura, un avversario capace, potrebbe cambiare il contenuto di un messaggio cifrato, senza conoscere chiave o algoritmo di cifratura.
- **Non Ripudiabilità:** è impossibile negare l'occorrenza di una determinata azione. Ad esempio durante la trasmissione di messaggi, il mittente non potrà negare di aver iniziato uno scambio di messaggi, come ad esempio per la **P.E.C.**, ovvero la posta certificata, in generale servizi che attestano l'invio e la ricezione dell'informazione.
- **Anonimia:** proteggere l'identità di chi sta utilizzando un servizio o proteggere l'accesso al servizio stesso.

8.4 Funzioni di Hash - Proprietà

- **One-Way (pre-image resistant):** dato un hash y è computazionalmente difficile trovare M dove $y = h(M)$.
- **Sicurezza Debole (second pre-image resistant):** dato M , è computazionalmente difficile trovare una variazione di M , chiamata M' , dove $h(M) = h(M')$.

- **Sicurezza Forte (collision resistant):** è computazionalmente difficile trovare due diversi messaggi con lo stesso valore di hash.

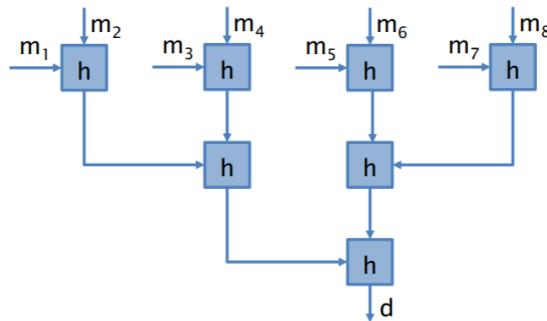
Una **One-Way Hash Function** verifica le prime due proprietà e viene detta **Weak One-Way Hash Function**.

Una **Collision Resistant Hash Function** verifica invece la terza proprietà e viene detta **Strong One-Way Hash Function**.

8.5 Funzioni di Hash - Costruzione

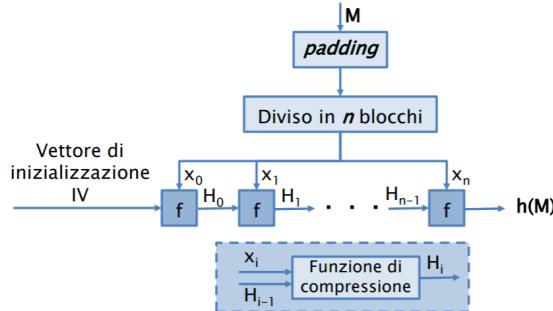
Messaggi di lunghezza arbitraria sono trattati utilizzando hash con input fisso. Il messaggio di input M viene diviso in K blocchi di lunghezza fissa $[m_1, m_2, \dots, m_K]$, i blocchi poi possono essere trattati in modo seriale/iterato oppure in modo parallelo.

8.5.1 Modello Parallelo



In questo esempio il messaggio è diviso in 8 *blocchi* tutti della stessa lunghezza. I blocchi di due in due vengono dati in pasto alla funzione di hash h , ed ogni output viene combinato con l'output dei blocchi successivi fino ad arrivare ad un output finale unico. Questo modello risulta esser più veloce in quanto elabora contemporaneamente più funzioni e risulta performante. Ovviamente solo se la funzione h risulta essere resistente alle collisioni, allora anche il modello lo sarà.

8.5.2 Modello Iterato

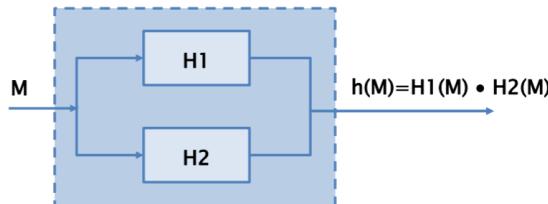


Il messaggio viene sempre suddiviso in n blocchi. Al messaggio iniziale viene aggiunta una fase di **padding**, ovvero se ho un valore di **10** e voglio 3 blocchi, allora avrò:

- **Primo Blocco** → [1, 2, 3, 4]
- **Secondo Blocco** → [5, 6, 7, 8]
- **Terzo Blocco** → [9, 10, ...]

In questo modo abbiamo l'ultimo blocco con dimensione diversa dagli altri, per questo motivo introduciamo il padding, ovvero normalizzare il nostro input in modo da non avere blocchi di dimensioni diverse. A differenza del modello parallelo, qui, il primo blocco viene dato in pasto ad un vettore di inizializzazione che a sua volta viene poi dato in pasto alla funzione di hash f , il quale output viene dato al blocco successivo, così fino ad arrivare al blocco n -esimo. Una collisione per $h(M)$ implica una collisione di h , questo modello risulta essere più lento ma anche più sicuro.

8.5.3 Modello Cascata



Una collisione per $h(M)$ vuol dire trovare una collisione sia per $H1$ che per $H2$. Usando due algoritmi diversi è più difficile trovare una collisione. L'output dell'hash sarà il prodotto delle due funzioni di hash combinate insieme. $h(M) = H1(M) * H2(M)$

9 Lezione 9

9.1 Funzioni di Hash

9.1.1 Little-endian e Big-endian

Rappresentazione di parole da 32 bit



- **Little-endian:** il byte con indirizzo più basso è quello meno significativo.

Valore parola: $W = 2^{24}B4 + 2^{16}B3 + 2^8B2 + 2^0B1$

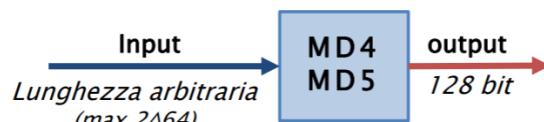
- **Big-endian:** il byte con indirizzo più basso è quello più significativo.

Valore parola: $W = 2^0B4 + 2^8B3 + 2^{16}B2 + 2^{24}B1$

9.2 MD4/MD5 - Message Digest

- **MD4:** fu progettato nel 1990, venne formalizzato con la [RFC 1320 → RFC 6150](#).
- **MD5:** fu progettato nel 1991, venne formalizzato con la [RFC 1321 → RFC 6151](#).

Operazioni efficienti su architetture a 32 bit *little-endian*.



In input possiamo avere al massimo 2^{64} bit.

9.2.1 Obiettivi di MD4/MD5

- **Sicurezza Forte:** computazionalmente è difficile trovare due messaggi con lo stesso hash.
- **Sicurezza Diretta:** sicurezza non basata su problemi teorici difficili computazionalmente.
- **Velocità:** algoritmo adatto per implementazioni software molto veloci.
- **Semplicità e Compattezza:** semplice da descrivere e da implementare, nessun uso di tabelle e di complesse strutture dati.

9.2.2 Padding del messaggio

Sia per **MD4** che per **MD5** il messaggio viene processato in diversi blocchi, tutti da **512 bit**. Non per forza tutti i blocchi riescono ad essere composti da **512 bit**, allora viene fatto uso del padding. Ogni blocco sarà costituito da **16 parole di 32 bit**. M' è l'hash del messaggio originario M , esso sarà costruito utilizzando:

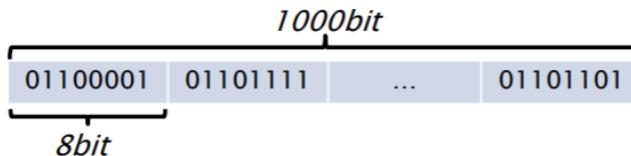
Messaggio originario M , p bit di padding e poi b bit di rappresentazione della lunghezza di M (Max 2^{64} , quindi avremo:

$$M' = M \underbrace{100\dots0}_{p} \underbrace{b}_{64\text{ bit}}$$

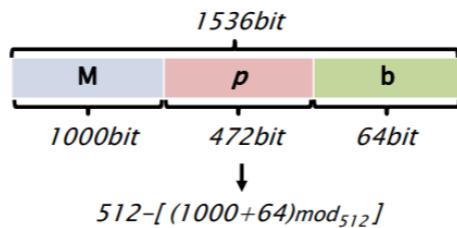
$$p = 512 - [(M + b) \bmod 512]$$

L'ultimo blocco sarà composto, qualora ci fosse il pudding, dai bit di padding e dai bit rappresentativi della lunghezza del messaggio. Esempio pratico:

$$|M| = 1000 \text{ bit}$$



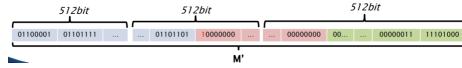
$$M' = Mp b$$



3 blocchi da **512 bit** = 1536 bit, per conoscere il numero di bit di padding devo fare:

$$1536 - (1000 + 64) = 472 \text{ oppure } 512 - [(1000 + 64) \bmod 512] = 512 - 40 = 472$$

Così abbiamo che i primi **1000 bit**, divisi sui primi due blocchi, rappresentano il messaggio, altri **24 bit** restanti nel secondo blocco sono usati per il padding ed i restanti **448 bit** di padding sono presi dal terzo blocco che viene completato dai **64 bit** di rappresentazione per arrivare a **512 bit** totali.



Ogni blocco ha **16 parole** da **32 bit**, quindi per 3 blocchi abbiamo **48 parole**.

9.2.3 Operazioni

MD4 ed MD5 impiegano diverse operazioni sulle parole, ogni operazione prende in input due parole X, Y e restituiscono una nuova parola. Tra le varie operazioni abbiamo:

- $X \wedge Y$: AND bit a bit di X ed Y .
- $X \vee Y$: OR bit a bit di X ed Y .
- $X \oplus Y$: XOR bit a bit di X ed Y .
- $\neg X$: COMPLEMENTO bit a bit di X .
- $X + Y$: SOMMA INTERA modulo 2^{32} .
- $X \ll s$: SHIFT circolare a sinistra di s bit.

9.2.4 Funzioni

Funzioni definite su parole a **32 bit**. MD4 prevede **3 ROUND** mentre MD5 ne prevede **4 ROUND**. Di conseguenza ci sono $3/4$ funzioni che lavorano in questo modo:

- **MD4:**
 - **ROUND 1:** $F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$ [IF X THEN Y ELSE Z].
 - **ROUND 2:** $G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge Z) \vee (X \wedge Y)$ [2 su 3].
 - **ROUND 3:** $H(X, Y, Z) = X \oplus Y \oplus Z$ [Bit di Parità].
- **MD5:**
 - **ROUND 1:** $F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$ [IF X THEN Y ELSE Z].
 - **ROUND 2:** $G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$ [IF Z THEN X ELSE Y]
 - **ROUND 3:** $H(X, Y, Z) = X \oplus Y \oplus Z$ [Bit di Parità].
 - **ROUND 4:** $I(X, Y, Z) = Y \oplus (X \vee (\neg Z))$

X	Y	Z	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Ogni round prende in input un blocco corrente, quindi 16 parole, inoltre prende il valore corrente del buffer ed un insieme di 4 parole, $ABCD$, per 128 bit. Ogni round consiste di 16 operazioni:

- $[ABCD.k.s] \rightarrow \text{MD4}$
- $[ABCD.k.s.i] \rightarrow \text{MD5}$, dove k, s, i sono delle costanti.

L'output dell'ultima fase viene sommato all'input della prima fase, parola per parola. L'output della L -esima fase è il digest a 128 bit.

9.2.5 Funzione di Compressione - MD4

Ogni round abbiamo detto avere 16 operazioni, ognuna delle quali agisce su un buffer di 4 parole ($ABCD$).

$t = (A + W(B, C, D) + X[j] + y[j]) \ll s[j] \Leftrightarrow (A, B, C, D) = (D, T, B, C)$
Dove :

- W : è la funzione relativa al round corrente (F, G, H).
- $X[j]$: è predefinito e reperibile nell'algoritmo.
- $y[j]$: è una costante additiva relativa al round corrente.
- $s[j]$: shift ciclico.

9.2.6 Funzione di Compressione - MD5

Ogni round abbiamo detto avere 16 operazioni, ognuna delle quali agisce su un buffer di 4 parole $ABCD$.

$A \leftarrow B + (A + W(B, C, D) + X[k] + T[i]) \ll s$

Dove:

- W : è la funzione relativa al round corrente (F, G, H, I).
- k : è l'indice della parola.
- i : è l'indice dell'iterazione.
- $X[k]$: è la k -esima parola di 32 bit nell' i -esimo blocco = $M'[16i+k]$.
- $T[i]$: è l' i -esimo elemento della tabella di costanti.
- s : è lo shift ciclico.

1	d76aa478	17	f61e2562	33	fffa3942	49	f4292244
2	e8c7b756	18	c040b340	34	8771f681	50	432aff97
3	242070db	19	265e5a51	35	6d9d6122	51	ab9423a7
4	c1bdceee	20	e9b6c7aa	36	fde5380c	52	fc93a039
5	f57c0faf	21	d62f105d	37	a4beea44	53	655b59c3
6	4787c62a	22	02441453	38	4bdecfa9	54	8f0ccc92
7	a8304613	23	d8a1e681	39	f6bb4b60	55	ffeff47d
8	fd469501	24	e7d3fb8	40	bebfb70	56	85845dd1
9	698098d8	25	21e1cde6	41	289b7ec6	57	6fa87e4f
10	8b44f7af	26	c33707d6	42	eaa127fa	58	fe2ce6e0
11	fffff5bb1	27	f4d50d87	43	d4ef3085	59	a3014314
12	895cd7be	28	455a14ed	44	04881d05	60	4e0811a1
13	6b901122	29	a9e3e905	45	d9d4d039	61	f7537e82
14	fd987193	30	fcefa3f8	46	e6db99e5	62	bd3af235
15	a679438e	31	676f02d9	47	1fa27cf8	63	2ad7d2bb
16	49b40821	32	8d2a4c8a	48	c4ac5665	64	eb86d391

Figura 1: Tabella delle Costanti.

9.2.7 Implementazione dell’algoritmo MD4

Le parole all’interno del buffer vengono inizializzate con dei valori definiti e conosciuti, successivamente vengono eseguiti due cicli, il primo che scorre tutte le parole nel blocco corrente ed il secondo per scorrere le singole parole. Per ogni round vengono poi usate le funzioni descritte per **MD4**, nel primo round la funzione **F**, nel secondo la funzione **G** e nel terzo ed ultimo round la funzione **H**.

```

A ← 01234567; B ← 89abcdef; C ← fdecba98; D ← 76543210;
for i = 0 to N/16-1 do
    for j = 0 to 15 do X[j] ← M'[16i+j]
    AA ← A; BB ← B; CC ← C; DD ← D;
    // [ABCD K S] INDICA A = (A + F(B,C,D) + X[K]) << S
    [ABCD. 0. 3] [DABC. 1. 7] [CDAB. 2.11] [BCDA . 3.19]
    [ABCD. 4 . 3] [DABC. 5. 7] [CDAB. 6.11] [BCDA . 7.19]
    [ABCD. 8 . 3] [DABC. 9. 7] [CDAB.10.11] [BCDA.11.19]
    [ABCD.12. 3] [DABC.13. 7] [CDAB.14.11] [BCDA.15.19]

    Round 1   // [ABCD K S] INDICA A = (A + G(B,C,D) + X[K]) + 5A827999) << S
    [ABCD. 0. 3] [DABC. 4. 5] [CDAB. 8. 9] [BCDA.12.13]
    [ABCD. 1. 3] [DABC. 5. 5] [CDAB. 9. 9] [BCDA.13.13]
    [ABCD. 2. 3] [DABC. 6. 5] [CDAB.10. 9] [BCDA.14.13]
    [ABCD. 3. 3] [DABC. 7. 5] [CDAB.11. 9] [BCDA.15.13]

    Round 2   // [ABCD K S] INDICA A = (A + H(B,C,D) + X[K] + 6ED9EBA1) << S
    [ABCD. 0. 3] [DABC. 8. 9] [CDAB. 4.11] [BCDA.12.15]
    [ABCD. 2. 3] [DABC.10. 9] [CDAB. 6.11] [BCDA.14.15]
    [ABCD. 1. 3] [DABC. 9. 9] [CDAB. 5.11] [BCDA.13.15]
    [ABCD. 3. 3] [DABC.11. 9] [CDAB. 7.11] [BCDA.15.15]

    Round 3
    A←A+AA; B ← B+BB; C ← C+CC; D ← D+DD;
    output: (A, B, C, D)

```

N = numero di «word»(32bit) del messaggio «M»

9.2.8 Implementazione dell'algoritmo MD5

Stessa procedura di MD4, ma con un round, e quindi anche una funzione, in più. In questo caso le costanti saranno scelta da una tabella di 64 costanti additive, non come per MD4 dove ne troviamo solamente 2.

```

A← 01234567; B ← 89abcdef; C ← fddecba98; D ← 76543210;
for i = 0 to N/16-1 do
    for j = 0 to 15 do X[j] ← M'[16i+j]
    AA ← A; BB ← B; CC ← C; DD ← D;
    // [ABCD k s ij] INDICA A = B + ((A + F(B, C, D) + X[k] + T[ij]) << s)
    [ABCD. 0. 7. 1] [DABC. 1.12. 2] [CDAB. 2.17. 3] [BCDA. 3. 22. 4]
    [ABCD. 4. 7. 5] [DABC. 5.12. 6] [CDAB. 6.17. 7] [BCDA. 7. 22. 8]
    [ABCD. 8. 7. 9] [DABC. 9.12.10] [CDAB.10.17.11] [BCDA.11.22.12]
    [ABCD.12. 7.13] [DABC.13.12.14] [CDAB.14.17.15] [BCDA.15.22.16]

    Round 1   // [ABCD k s ij] INDICA A = B + ((A + G(B, C, D) + X[k] + T[ij]) << s)
    [ABCD. 1. 5.17] [DABC. 6. 9.18] [CDAB.11.14.19] [BCDA. 0.20.20]
    [ABCD. 5. 5.22] [DABC.10. 9.22] [CDAB.15.14.23] [BCDA. 4.20.24]
    [ABCD. 9. 5.25] [DABC.14. 9.26] [CDAB. 3.14.27] [BCDA. 8.20.28]
    [ABCD.13. 5.29] [DABC. 2. 9.30] [CDAB. 7.14.21] [BCDA.12.20.32]

    Round 2   // [ABCD k s ij] INDICA A = B + ((A + H(B, C, D) + X[k] + T[ij]) << s)
    [ABCD. 5. 4.33] [DABC. 8.11.34] [CDAB.11.16.35] [BCDA.14.23.36]
    [ABCD. 1. 4.37] [DABC. 4.11.38] [CDAB. 7.16. 39] [BCDA.10.23.40]
    [ABCD.13. 4.41] [DABC. 0.11.42] [CDAB. 3.16.43] [BCDA. 6.23.44]
    [ABCD. 9. 4.45] [DABC.12.11.46] [CDAB.15.16.45] [BCDA. 2.23.48]

    Round 3   // [ABCD k s ij] INDICA A = B + ((A + I(B, C, D) + X[k] + T[ij]) << s)
    [ABCD. 0. 6.49] [DABC. 7.10.50] [CDAB. 5.15.51] [BCDA. 5. 21. 5]
    [ABCD.12. 6.53] [DABC. 3.10.54] [CDAB. 1.15.55] [BCDA. 1.21.56]
    [ABCD. 8. 6.57] [DABC.15.10.58] [CDAB.13.15.59] [BCDA.13.21.60]
    [ABCD. 4. 6.61] [DABC.11.10.62] [CDAB. 9.15.63] [BCDA. 9.21.64]

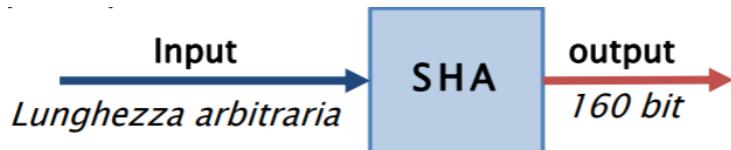
    A←A+AA; B ← B+BB; C ← C+CC; D ← D+DD;
    output: (A, B, C, D)

```

N = numero di «word»(32bit) del messaggio «M»

9.3 Algoritmo SHA

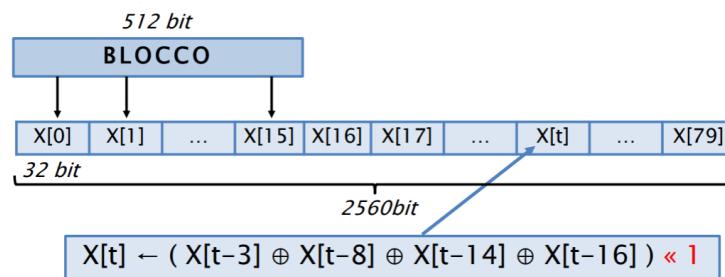
Secure Hash Algoritmo, anche detto **SHA**, è lo standard del governo americano dal 1993. Diciamo che in parte ha sostituito l'**MD5**, ma vengono tutt'ora usati entrambi. Nel 1994 fu modificato e nominato **SHA-1** certificato dall'**RFC 3174** e dalla **RFC 6194**, questa modifica consisteva nell'aggiunta di uno shift nell'espansione dei blocchi. Questo algoritmo opera su architetture a 32 bit Big-endian e risulta più sicuro rispetto ad **MD4/MD5** in quanto ha un output di **160 bit**.



In questo algoritmo il processo di padding risulta lo stesso di MD4/MD5, viene fatto solo un cambiamento, ovvero viene aggiunta l'**espansione del blocco di iterazione**.

9.3.1 Espansione del blocco di iterazione

Il primo blocco di 512 bit viene copiato così com'è nel blocco di espansione e per le successive sotto-sequenze di 32 bit si fa uso della funzione $X[t]$ che va a costruire, attraverso degli **XOR** e delle precedente parola, il blocco finale. Sono in totale **4 round** con 20 operazioni ciascuno e per ogni iterazione una parola $X[i]$ viene calcolata dal blocco di input.



- **ROUND 1:** $t = 0, \dots, 19 : F(t, X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$ [IF X THEN Y ELSE Z].
 - **ROUND 2:** $t = 20, \dots, 39 : F(t, X, Y, Z) = X \oplus Y \oplus Z$ [Bit di Parità].
 - **ROUND 3:** $t = 40, \dots, 59 : F(t, X, Y, Z) = (X \wedge Z) \vee (Y \wedge Z) \vee (X \wedge Y)$ [2 su 3].
 - **ROUND 4:** $t = 60, \dots, 79 : F(t, X, Y, Z) = Y \oplus X \oplus Z$ [Bit di Parità].

Avendo 80 operazioni totali esse vengono divise in ogni round per blocco, in più vengono aggiunte le costanti additive $k[t]$ per ogni round:

- **ROUND 1:** $5a827999$
- **ROUND 2:** $6ed9eba1$
- **ROUND 3:** $8f1bbcdcc$
- **ROUND 4:** $ca62c1d1$

X	Y	Z	F(0,..)	F(20,..)	F(40,..)	F(60,..)
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

9.3.2 Rappresentazione Algoritmo SHA-1

```
A=67452301; B=efcdab89; C=98badcfe; D=10325476; E=c3d2e1f0;  
for i = 0 to N/16-1 do  
    for j = 0 to 15 do  
        X[j] ← M'[16i+j]  
    for t = 16 to 79 do  
        X[t] ← ( X[t-3] ⊕ X[t-8] ⊕ X[t-14] ⊕ X[t-16] ) « 1 ← Aggiunto  
        con SHA-1  
        AA ← A; BB ← B; CC ← C; DD ← D; EE ← E;  
    for t=0 to 79 do  
        TEMP ← (A«5) + F(t,B,C,D) + E + X[t] + K[t]  
        E ← D  
        D ← C  
        C ← (B«30)  
        B ← A  
        A ← TEMP  
        A ← A + AA; B ← B + BB; C ← C + CC; D ← D + DD; E ← E + EE;  
output: (A, B, C, D, E)
```

9.4 Breve differenza tra MD4/MD5 e SHA-1

SHA-1 ha una sicurezza maggiore con un output di 32 bit più lungo rispetto ai 128 bit di MD4/MD5. Sono entrambi algoritmi molto veloci, SHA-1 ha più passi ma risulta ugualmente veloce e semplice da scrivere ed implementare.

10 Lezione 10 ed 11

10.1 Analisi

L'analisi va sempre eseguita su una copia, deve essere ripetibile scientificamente. Essendo ripetibile, utilizzando diversi strumenti e diverse operazioni, avremo sempre lo stesso risultato. L'analisi ci permette di ricostruire eventi passati mediante la lettura di dati digitali, essa ha **3 obiettivi** principali:

1. **Riscontro delle informazioni**: analisi sui reperti a riscontro di ciò che già conosce il P.M.
2. **Nuove informazioni**: portare alla luce nuovi elementi di indagine.
3. **Contro analisi**: smontare l'analisi tecnica della controparte.

10.2 Montare un file immagine - Linux

Dopo aver acquisito il disco di interesse con il comando **dd** possiamo analizzare il file immagine creato a partire dal supporto e poi montarlo, ovvero renderlo visibile tra i dischi del sistema riuscendo quindi ad analizzare il supporto senza collegarlo fisicamente.

```
root@caine:/# fdisk -l /mnt/dest/dd_image/sda.dd
Disk /mnt/dest/dd_image/sda.dd: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72a3c36c

Device      Boot   Start     End Sectors Size Id Type
/mnt/dest/dd_image/sda.ddp1        2048 2099199 2097152   1G b  W95 FAT32
/mnt/dest/dd_image/sda.ddp2    2099200 8388607 6289408   3G b  W95 FAT32
```

Utilizzando il comando **fdisk -l** e specificando il path in cui si trova l'immagine del supporto che vogliamo analizzare, possiamo listare la struttura, ovvero la partizione dell'immagine, ovvero del supporto in analisi. In questo caso abbiamo due partizioni che sono **sda.ddp1** ed **sda.ddp2**.

10.2.1 Come montare l'immagine?

Ciò che andremo a montare è una delle partizioni identificate nella fase precedente, ovvero quella con i dati di interesse.

Device	Boot	Start	End	Sectors	Size	Id	Type
/mnt/dest/dd_image/sda.ddp1		2048	2099199	2097152	1G	b	W95 FAT32
/mnt/dest/dd_image/sda.ddp2		2099200	8388607	6289408	3G	b	W95 FAT32

Usando linux, facciamo uso del comando **mount** con alcuni parametri, in modo da non danneggiare la copia, esso però accetta solo immagini in formato **DD/Raw** non segmentate.

```
root@caine:/# mount -o ro,loop,offset=1074790400 /mnt/dest/dd_image/sda.dd /mnt/sda_dd
```

Dove i parametri:

- **ro**: indica la modalità di sola lettura, "read only".
- **loop**: consente di creare un virtual block device da un file.
- **offset="byte"**: punto di inizio della partizione da montare.
[questo perché ci troviamo ad avere più partizioni]

Qualora avessimo un disk image, sempre in formato **DD/Raw** però spartito su più file, allora dobbiamo procedere diversamente:

```
root@caine:/# ls -l /mnt/dest/dd_image/
total 4194308
-rwxrwxrwx 1 root root 2147483648 apr  8 01:16 sda.000
-rwxrwxrwx 1 root root 2147483648 apr  8 01:23 sda.001
-rwxrwxrwx 1 root root      823 apr  8 01:23 sda.log
```

Prima di effettuare il **mount** bisogna "fondere" i file spartiti in unico file. Per farlo utilizziamo il comando **affuse** al quale viene dato il primo segmento dell'immagine e poi il path di destinazione, quello che farà questo comando è creare un link al merge dei segmenti, quindi è una fusione **virtuale** che non occuperà spazio.

Successivamente procediamo come consueto con **mount**.

```
root@caine:/# affuse /mnt/dest/dd_image/sda.000 /mnt/sda_fuse
root@caine:/# ls -l /mnt/sda_fuse/
total 0
-r--r--r-- 1 root root 4294967296 gen 1 1970 sda.000.raw

root@caine:/# fdisk -l /mnt/sda_fuse/sda.000.raw
Disk /mnt/sda_fuse/sda.000.raw: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72a3c36c
Device            Boot      Start     End  Sectors  Size Id Type
/mnt/sda_fuse/sda.000.raw1          2048 2099199 2097152   1G b W95 FAT32
/mnt/sda_fuse/sda.000.raw2        2099200 8388607 6289408   3G b W95 FAT32
```

Se invece avessimo uno dei formati della famiglia EWF spartiti, allora procediamo come segue:

```
root@caine:/# ls -l /mnt/dest/e01_image/
total 235526
-rw-r--r-- 1 root root 104857600 apr  8 02:26 sda.E01
-rw-r--r-- 1 root root 104857600 apr  8 02:28 sda.E02
-rw-r--r-- 1 root root 31457280 apr  8 02:29 sda.E03
-rw-r--r-- 1 root root    7161 apr  8 02:29 sda.info
```

In questo esempio abbiamo un disk image del formato **E01**, il comando per eseguire il merge dei file della famiglia EWF è **ewfmount**.

```
root@caine:/# ewfmount /mnt/dest/e01_image/sda.E01 /mnt/sda_fuse
root@caine:/# ls -l /mnt/sda_fuse/
total 0
-r--r--r-- 1 root root 4294967296 apr  8 02:31 ewf1

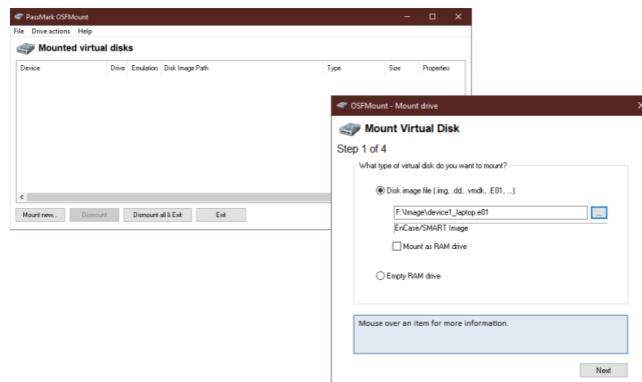
root@caine:/# fdisk -l /mnt/sda_fuse/ewf1
Disk /mnt/sda_fuse/ewf1: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72a3c36c
Device            Boot      Start     End  Sectors  Size Id Type
/mnt/sda_fuse/ewf1p1          2048 2099199 2097152   1G b W95 FAT32
/mnt/sda_fuse/ewf1p2        2099200 8388607 6289408   3G b W95 FAT32
```

Questo prende come primo elemento il primo segmento dell'immagine e come secondo la destinazione per il file merged. Fatto questo possiamo procedere come consuetudine con il comando **mount**. Ovviamente oltre ai tool da riga di comando, linux offre anche un tool con interfaccia grafica, che su caine, si chiama **IMG_MAP**

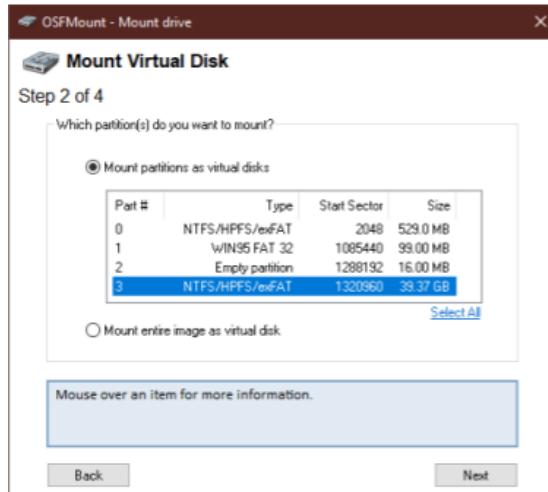
10.3 Montare un file immagine - Windows

10.3.1 OSFMount Tool

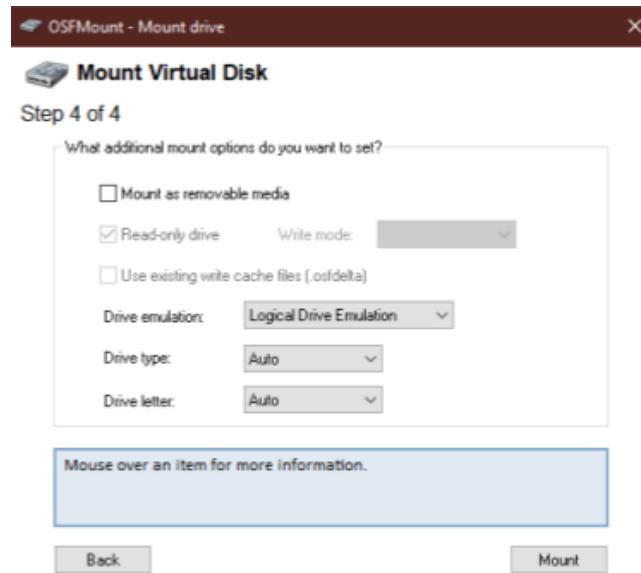
Anche su Windows esistono tool in grado di montare un file immagine. **OSFMount** è un tool con interfaccia grafica intuitivo e semplice.



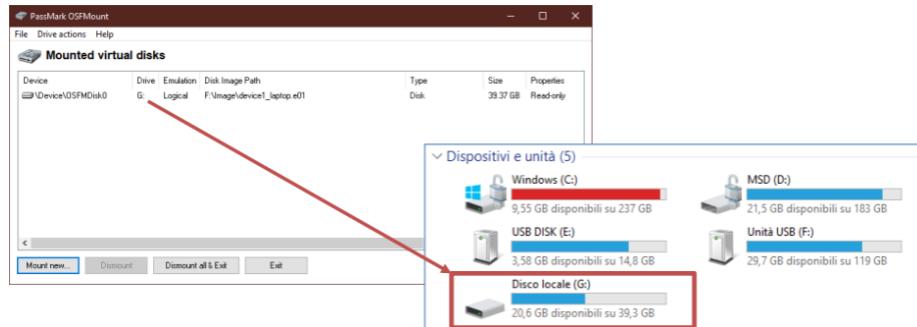
Qualora si volesse montare un nuovo disk image si aprirà una sequenza di 4 step divisi in varie schermate, il terzo di questi step viene eseguito solo quando si seleziona l'opzione "*Mount as RAM drive*". Nel primo step dovremo selezionare il patch del disk image, qualora fosse segmentato basta scegliere il primo segmento.



Il secondo step richiede di decidere tra le partizioni presenti, qualsiasi fossero, e poi proseguire per montare una o più partizioni selezionate.



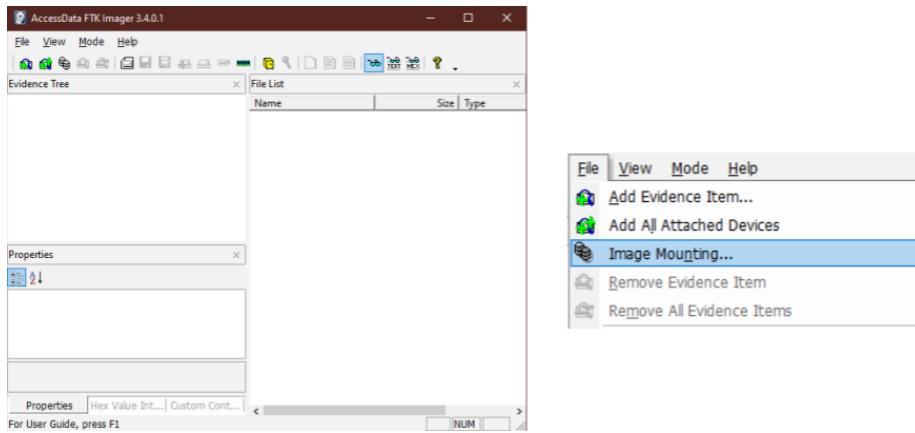
È inoltre possibile specificare dei parametri per il montaggio del disk image. Di default viene montato come *read-only drive*.



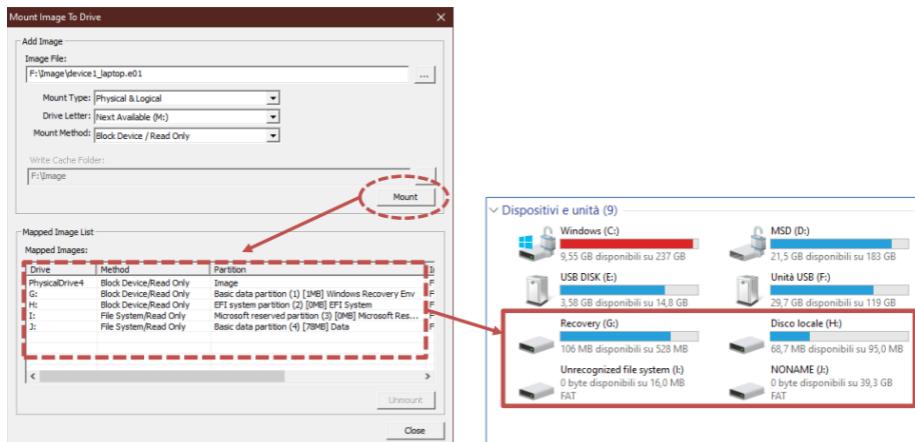
Una volta montato il disco virtuale, Windows riconoscerà l'unità come agganciata anche se è solo un file immagine.

10.3.2 FTK Imager Tool

Un altro tool molto interessante per Windows è **FTK Imager**.



Tra le varie utility di cui dispone questo tool troviamo anche **Image Mounting**, è quindi possibile montare un disk image anche con questo tool.



Selezionando il primo segmento dell'immagine, settando poi i parametri di nostro interesse ed infine procedendo con il *mounting*, effettuiamo il montaggio di tutte le partizioni del disk image.

10.4 Pro vs. Contro di montare un file immagine

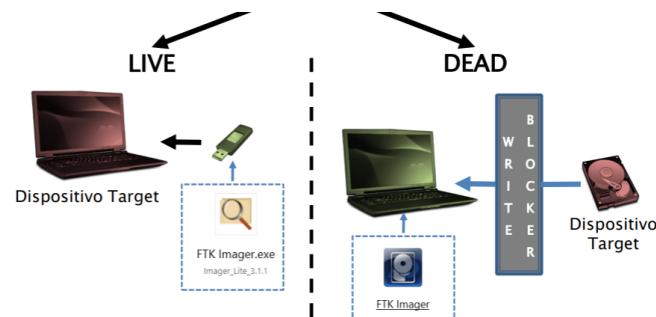
- **PRO:** è un metodo veloce per operazioni semplici, come per estrarre un file ben preciso. Utilizzo di tool non forensics oriented, quindi il fatto di poter montare un disk image al proprio

dispositivo, rende utilizzabili tool che sono al di fuori della forensics, come ad esempio software anti-malware per l'analisi del disco.

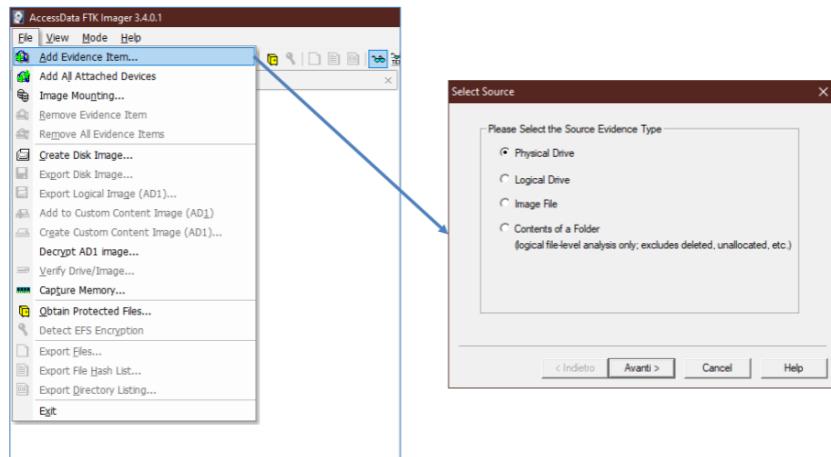
- **CONTRO:** è farraginoso, fare l'analisi completa di un disco in questo modo è tedioso e confusionario. Inoltre è possibile vedere solo i file residenti, ovvero quei file che il file system riconosce, quindi file come quelli nascosti o quelli eliminati non vengono mostrati. Inoltre visto che il riconoscimento del file system è demandato al nostro sistema operativo, se questo non è compatibile allora non potremmo analizzare il disk image.

10.5 FTK Imager come strumento di Analisi

FTK Imager potrebbe essere utile non solo per leggere o montare un disk image, ma potrebbe essere utile anche in fase di perquisizione e sequestro come strumento di analisi in quanto possiamo utilizzarlo sul dispositivo target sia in modalità **LIVE** che in modalità **DEAD**



10.5.1 Come eseguire l'analisi con FTK Imager



Recandoci in alto a sinistra nella home del tool, possiamo notare diverse voci. Per iniziare una nuova analisi possiamo seguire **File** > **Add Evidence Item** e poi in base al tipo di analisi che vogliamo effettuare possiamo scegliere tra:

- **Physical Drive**: per analizzare un device.
- **Logical Drive**: per device con formato **L01**.
- **Image File**: per file immagine.
- **Contents of a Folder**: per un singolo file o una cartella.

10.5.2 Quali sono i limiti di FTK Imager?

È fondamentale conoscere i limiti degli strumenti che si utilizza, potrebbe capitare un tipo di file immagine o di un file system non supportato. Questi sono i vari formati supportati:

Hard Disk Image Formats

The following table lists AccessData Imager-identified and analyzed hard disk image formats:

Identified and Analyzed Hard Disk Image Formats

• Encase, including 6.12	• SnapBack
• Safeback 2.0 and under	• Expert Witness
• Linux DD	• ICS
• Ghost (forensic images only)	• SMART
• AccessData Logical Image (AD1)	• Advanced Forensics Format (AFF)

CD and DVD Image Formats

The following table lists AccessData Imager-identified and analyzed CD and DVD image formats:

Identified and Analyzed CD and DVD File Systems and Formats

• Alcohol (*.mds)	• IsoBuster CUE
• PlexTools (*.pxi)	• CloneCD (*.ccd)
• Nero (*.nrg)	• Roxio (*.cif)
• ISO	• Pinnacle (*.pdi)
• Virtual CD (*.vc4)	• CD-RW,
• VCD	• CD-ROM
• DVD+MRW	• DVCD
• DVD-RW	• DVD-VFR
• DVD+RW Dual Layer	• DVD-VR
• BD-R SRM-POW	• BD-R DL
• BD-R SRM	• CloneCD (*.ccd)
• HD DVD-R	• HD DVD-RW DL
• SVCD	• HD DVD

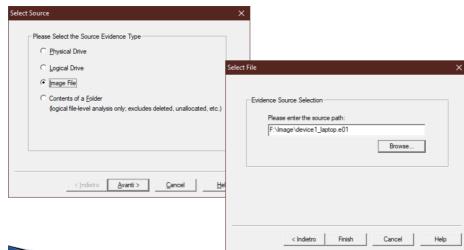
File Systems

The following table lists AccessData Imager-identified and analyzed file systems:

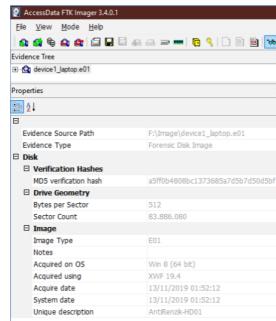
Identified and Analyzed File Systems

• APFS	• HFS
• CDFS	• HFS+
• exFAT	• NTFS
• Ext2FS	• ReiserFS3
• Ext3FS	• VXFS
• Ext4FS	• XFS
• FAT12, FAT16, FAT32	

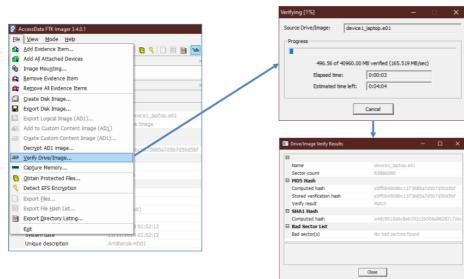
10.5.3 Analisi di un file immagine



Selezionando ***Image File*** tra le varie opzioni, fornendo poi il path del disk image da analizzare, verranno visualizzate alcune informazioni della copia forense, le **header info**:

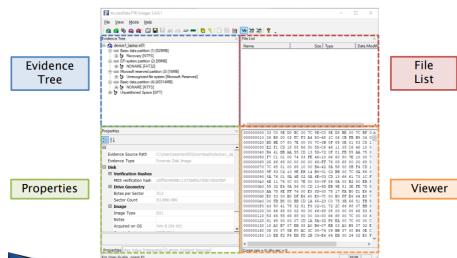


Le header info possono esserci utili quando non siamo noi gli autori della copia forense e vogliamo quindi conoscere come è stata fatta la copia, oppure quando. In primo luogo ci interessa l'hash per capire se la **catena di custodia** è stata rispettata.

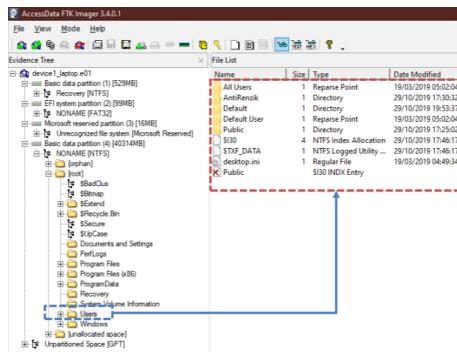


Per controllare che la copia non sia stata alterata possiamo utilizzare l'opzione in ***File > Verify Drive/Image*** che ci permette di effettuare nuovamente l'hash e controllare se matcha con l'hash che troviamo nell'header.

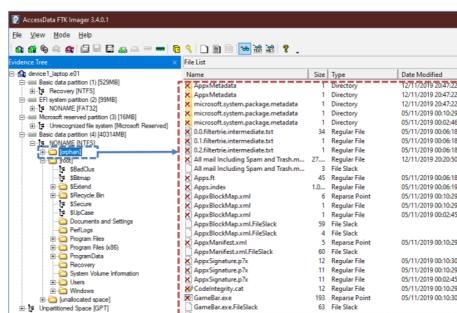
10.5.4 La GUI di FTK Imager



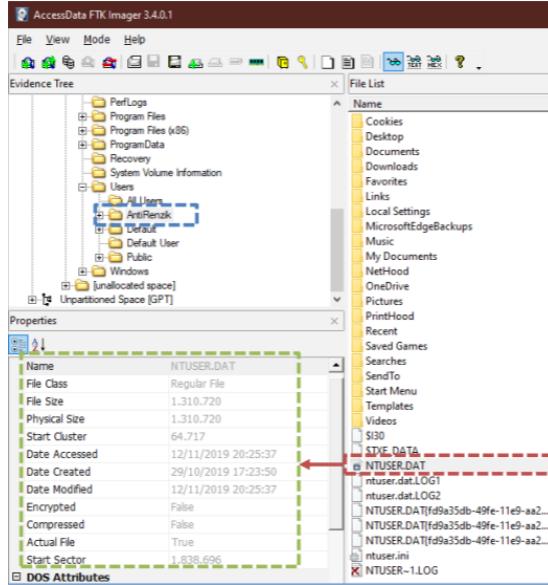
Rispetto ad un analisi grezza con **FTK Imager** abbiamo a disposizione molte più utilità. L'**evidence tree** mostra tutto il contenuto di ogni cartella e di fianco nella **file list** troviamo la lista dei file delle cartelle selezionate.



Una cosa non da poco che ci permette di fare questo software è vedere gli **orphan**, ovvero quei file che sono stati cancellati dall'utente, quindi file senza una cartella padre, ma che continuano ad esistere in memoria parziale.

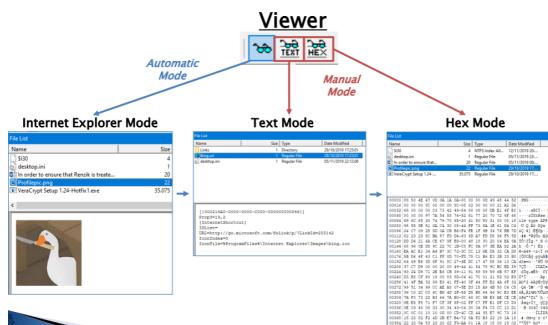


Nella piccola area in basso a sinistra abbiamo le proprietà dei file selezionati, per lo più vengono riportate informazioni temporali a livello di file system ed informazioni di sicurezza.

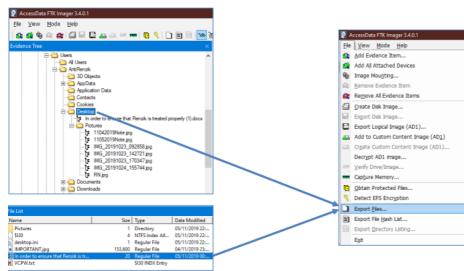


Nella schermata in basso a destra invece abbiamo il **viewer** che sostanzialmente ha tre modalità:

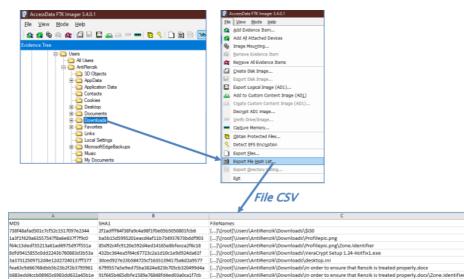
1. **Automatic Mode** che fa uso delle librerie di internet explorer per la visualizzazione di immagini.
2. **Manual Mode Text** che permette di avere una visione testuale dei dati.
3. **Manual Mode Hex** che permette di avere una visione testuale esadecimale dei dati.



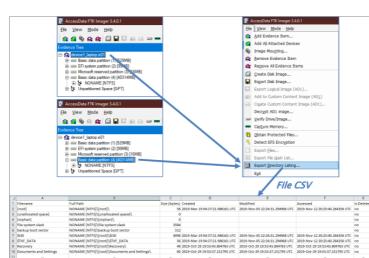
10.5.5 Export dei file di interesse



FTK Imager mette di nuovo a disposizione una funzionalità davvero utile, ovvero l'export dei file o cartelle di interesse. Basta cliccare con il tasto destro del mouse sugli elementi desiderati e poi scegliere **Export Files**. Attenzione ad esportare file durante perquisizioni LIVE, questo export sulla macchina target porterebbe ad un alterazione dello stato della macchina stessa, rendendo così irripetibili futuri accertamenti.



Oltre a poter esportare il file vero e proprio possiamo anche decidere di effettuare un export detto **Hash List** che esporta il calcolo dell'hash dei file selezionati.



Se volessimo invece esportare l'intero disk image per costruire una timeline con le informazioni temporali ritrovate, possiamo scegliere

il metodo **Directory Listing**, questo risulta utile qualora si volesse avere una timeline dei file e del dispositivo.

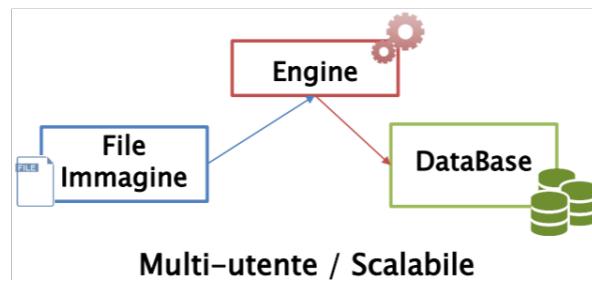
10.6 Strumenti software per l'analisi

Questi strumenti possono essere suddivisi in tre aree:

- **Toolkit**: sono degli strumenti completi che aiutano il forense nell'intera fasi di analisi.
- **Tool Forensics Oriented**: piccoli strumenti che permettono di svolgere specifiche task di analisi.
- **Tool Generici**: tool non progettati per la computer forensics ma che possono risultare utili per le analisi.

10.6.1 Toolkit - FTK vs. Autopsy

FTK (Forensics ToolKit) prodotto dalla stessa casa di **FTK Imager**, è un tool commerciale per Windows che offre diverse specifiche per l'analisi. **Autopsy** invece è un toolkit gratuito ed opensource multi-piattaforma. Entrambi prevedono una separazione a livello progettuale tra il file immagine da analizzare, l'engine che fa l'analisi ed il database dove conservare i risultati delle analisi.



Questo approccio permette ad entrambi i software di essere più scalabili, ad esempio permettendo di avere le tre risorse in luoghi diversi e permettendo un accesso multi-utente.

10.6.2 Toolkit - File Immagine Supportati

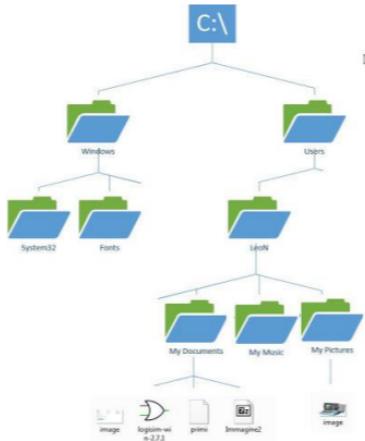
Forensic ToolKit (FTK)	Autopsy
▸ Encase E01	▸ Encase E01
▸ Encase L01 Logical Image	▸ Raw (DD, BIN, IMG)
▸ Expert Witness	▸ Virtual Disk (VMDK, VHD)
▸ SnapBack	
▸ Safeback 2.0 and under	
▸ ICS	
▸ Linux DD	
▸ SMART	
▸ Ghost (forensic images only)	
▸ MSVHD (MS Virtual Hard Disk)	
▸ AccessData Logical Image (AD1)	
▸ Lx0, Lx01	
▸ DMG (Mac)	
▸ VMDK (VmWare Disk)	

10.6.3 Toolkit - File System Supportati

Forensic ToolKit (FTK)	Autopsy
▸ FAT	▸ FAT
▸ exFAT	▸ ExFAT
▸ NTFS	▸ NTFS
▸ Ext2FS	▸ EXT2FS
▸ Ext3FS	▸ EXT3FS
▸ Ext4FS	▸ EXT4FS
▸ APFS	▸ APFS
▸ HFS, HFS+	▸ HFS, HFS+
▸ CDFS	▸ YAFFS2
▸ ReiserFS 3	
▸ VxFS (Veritas File System)	

10.6.4 Le viste nei Toolkit

Queste tornano utili per analizzare i disk image in modo da visualizzare informazioni ben precise ed utili in molteplici modi. L'unica vista utilizzata fino ad ora è quella ad **albero** che propone una rappresentazione gerarchica dei file.



10.6.5 File Type View

Immaginiamo di voler analizzare tutti i file nel disk image di tipo **PDF** con la vista ad albero, ci vorrebbe molto tempo per cercare tutti quei file e poi analizzarli. La catalogazione in base alla tipologia di file aiuta i digital forensers proprio in questo, facilitando la ricerca, l'analisi risulterà più veloce. Questo tipo di catalogazione può avvenire in due modi:

- **Estensione:** suffisso del file, come ad esempio **.pdf**, **.docx**, ... può essere facilmente alterato però.
- **Signature:** oppure detto **Magic Number**, ovvero una sequenza di bit posta nell'offset che serve a definire il formato in cui i dati sono salvati.

Oltre alla catalogazione per file esiste anche la **Classificazione** per file, essa consiste nel classificare i file in base a determinate proprietà, alcune di queste sono ottenute attraverso il processo di analisi che fa il toolkit. Alcune di queste catalogazioni sono:

- **Bad Extension:** esegue un confronto tra l'estensione del file e la signature, verificandone la coerenza.
- **Delete File:** file marcati come cancellati dal file system ma che ancora risiedono in memoria.

Hex signature	89 50 4E 47 0D 0A 1A 0A
ASCII	.PNG....
Offset	0
Ext	PNG

Name	Size
020	4
desktop.ini	1
In order to ensure that...	20
Profilepic.png	22



Name	Size	Type	Date Modified
\$130	4	NTFS Index All...	12/11/2019 20...
desktop.ini	1	Regular File	05/11/2019 22...
In order to ensure that...	20	Regular File	05/11/2019 00...
Profilepic.png	22	Regular File	29/10/2019 17...

```

0000 89 50 4E 47 0D 0A 1A 0A-00 00 00 0D 49 48 44 52 [PNG]...!...IHDR
0010 00 00 00 0C 00 00 00 0C-08 02 00 00 00 21 A2 D6 [!...!...!...!...!...!...!...]
0020 69 00 00 00 03 73 42 49-54 08 08 08 DB E1 4F E0 i...-sBIT-0@0A
0030 00 00 00 97 7A 54 58 74-52 61 77 20 70 72 6F 66 ...zTxtRaw prot
0040 69 EC 65 20 74 79 70 65-20 41 50 50 31 00 00 18 file type APFI...

```


Views

- > File Types
 - > By Extension
 - Images (14612)
 - Videos (34)
 - Audio (192)
 - Archives (423)
 - Databases (39)
 - Documents
 - HTML (4202)
 - Office (20)
 - PDF (2)
 - Plain Text (342)
 - Rich Text (574)
 - Executable
 - .exe (3219)
 - .dll (2148)
 - .bat (9)
 - .cmd (11)
 - .com (22)

By MIME Type

- > application
- > audio
 - mp4 (35)
 - mid (3)
 - mpeg (14)
 - vnd.wave (82)
 - x-ms-wma (18)
- > image
 - vnd.microsoft.icon (145)
 - tiff (1)
 - bmp (178)
 - gif (825)
 - x-portable-graymap (1)
 - png (11174)
 - jpeg (958)
 - svg+xml (138)
 - vnd.abrush.pcx (4)
 - webp (91)
- > text
- > video

Figura 2: Autopsy - File Type View

File Extension (161.695 / 161.695)

- <missing> (14.944 / 14.944)
 - 000 (3 / 3)
 - 001 (3 / 3)
 - 002 (3 / 3)
 - 003 (1 / 1)
 - 30319 (1 / 1)
 - 30319 64 (1 / 1)
 - 3mf (10 / 10)
 - 5b (1 / 1)
 - 67 (1 / 1)
 - 6c (1 / 1)
 - 7db (2 / 2)
 - 7e (1 / 1)
 - 7f (2 / 2)
 - 80 (1 / 1)
 - 87 (1 / 1)
 - a0 (1 / 1)
 - ad (10 / 10)
 - acm (24 / 24)
 - adnin (3 / 3)

File Category (349.144 / 349.144)

- + Archives (2.927 / 2.927)
- + Databases (46 / 46)
- + Documents (67.937 / 67.937)
- + Email (257 / 257)
- + Executable (9.196 / 9.196)
- + Folders (36.929 / 36.929)
- + Graphics (17.544 / 17.544)
- + Internet/Chat Files (4.128 / 4.128)
- + Mobile Phone (0 / 0)
- + Multimedia (305 / 305)
- + OS/File System Files (18.015 / 18.015)
- + Other Encryption Files (12.289 / 12.289)
- + Other Known Types (1.346 / 1.346)
- + Presentations (4 / 4)
- + Slack/Free Space (16.510 / 16.510)
- + Spreadsheets (4 / 4)
- + Unknown Types (161.707 / 161.707)
- + User Types (0 / 0)

Figura 3: FTK - File Type View

10.6.6 Known File View

Un'altra tipologia di analisi che permettono i toolkit è tramite l'hash, ovvero possiamo decidere di far calcolare l'hash di tutti i file nel disk image. Questo processo ha lo scopo di confrontare l'hash dei file in analisi con un **database/elenco** di hash di file già noti e pubblici, andando a trovare i così detti **Known File** i quali sono suddivisi in due categorie:

- **Ignorable File**: file conosciuti e non di interesse. Sottrazione di migliaia di file dall'analisi. Libreria più usata/conosciuta è **National Software Reference Library**.
- **Notable File**: file conosciuti di notevole interesse. Ricerca mirata di determinati file. Alcuni progetti come **Project VIC** nacquero per combattere la pedopornografia, costruendo librerie di hash di file conosciuti e di interesse.

10.6.7 Artefatti View

Oltre questo tipo di analisi abbiamo che i toolkit producono degli **Artefatti**, ovvero ci permettono di creare nuove informazioni dall'unione e dall'estrazione dei file presenti nel disk image.

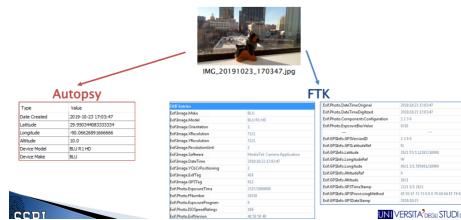


Ad esempio:

- **Metadati**: sono informazioni strutturate aggiuntive del file. Queste informazioni non sono aggiunte dall'utente ma dal software stesso. Per un documento word queste informazioni potrebbero essere "il numero di volte in cui è stato editato" ad esempio.

General	General & Microsoft Word Metadata
Name: Corso di CF_lezione_137.docx	Name: Corso di CF_lezione_137.docx
Item Number: 353081	File Type: Microsoft Word 2010 (M4)
Location: Corso di CF_lezione_137.docx	Path: Corso di CF_lezione_137.docx
Author: Marco ARR	Author: Marco ARR
Template: Normal.dotm	Template: Normal.dotm
Last modified: Marco ARR	Last modified: Marco ARR
Revision number: 7	Revision number: 7
Total editing time: 26 minutes, 57 seconds	Total editing time: 26 minutes, 57 seconds
Create date: 08/04/2010 10:18:00 (2010-04-08 10:18:00 UTC)	Create date: 08/04/2010 10:18:00 (2010-04-08 10:18:00 UTC)
Last saved date: 08/04/2010 10:18:00 (2010-04-08 10:18:00 UTC)	Last saved date: 08/04/2010 10:18:00 (2010-04-08 10:18:00 UTC)
Number of pages: 1	Number of pages: 1
Number of characters: 11,574	Number of characters: 11,574
Creating application: Microsoft Office Word	Creating application: Microsoft Office Word
Source: Microsoft Word	Source: Microsoft Word
Line Count: 162	Line Count: 162
Paragraphs: 41	Paragraphs: 41
Drop cap style: None	Drop cap style: None
Document Sections Count: Tables1	Document Sections Count: Tables1
Company: Microsoft	Company: Microsoft

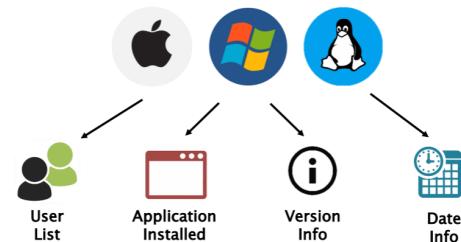
Uno dei metadati più famosi è sicuramente l'**Exif** ovvero le informazioni aggiuntive di una foto, come per esempio il dispositivo con cui è stata scattata, oppure il luogo dello scatto, ecc.



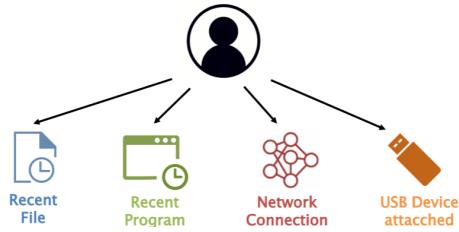
- **Email Archive:** un ulteriore tipologia di analisi comprende gli archivi di posta. Essi non sono altro che file molto complessi, anche visti come database, che devono essere analizzati, infatti le informazioni in questo archivio vanno elaborate e poi prodotte. Informazioni come allegati, mittenti e destinatari, orari e giorni, ecc.



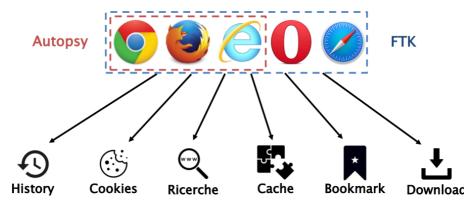
- **System Information:** informazioni riguardanti l'ambiente di lavoro, il sistema operativo, informazioni sugli utente, i programmi installati, la versione del sistema operativo e le informazioni temporali correlate.



- **User Activity:** tutte le informazioni relative all'utente all'interno del sistema operativo. Ad esempio per conoscere l'utente che ha avuto accesso ad un determinato file e quando, oppure quali programmi ha eseguito fino a prima del sequestro, oppure i dispositivi USB agganciati.

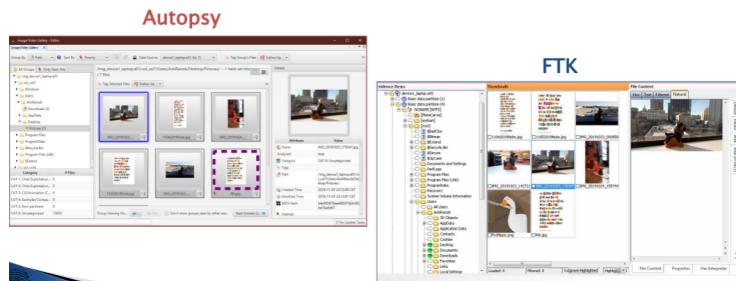


- **Navigazione Web:** analisi di tutte le informazioni sui file dei browser, di tutte le navigazioni internet. Ad esempio, la cronologia, i cookies, le precedenti ricerche, le cache, i downloads, ecc.



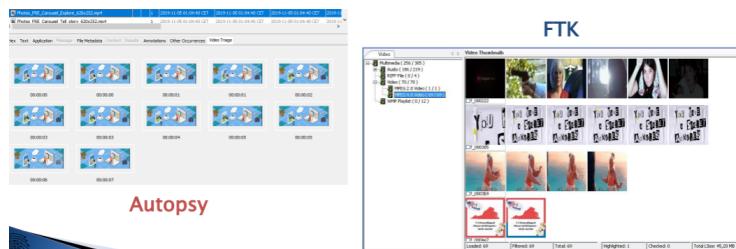
10.6.8 Image Gallery View

Visualizzazione in maniera veloce di file grafici in modo da poter identificare più velocemente ciò che è di nostro interesse.



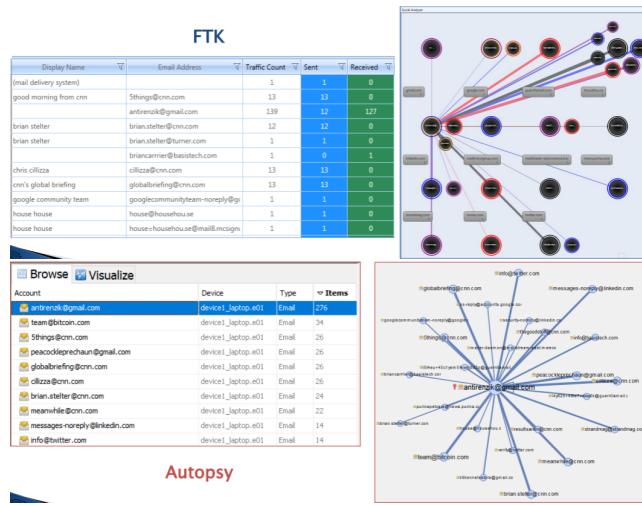
10.6.9 Video Gallery View

Stessa funzione della **Image Gallery** ma trattando anteprime di video, mostrando quindi una percentuale di video in anteprima oppure un tot di frame alla volta.



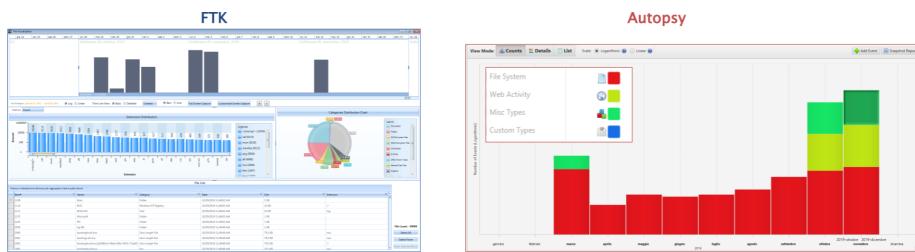
10.6.10 Social Analyzer View

Visualizzare ed evidenziare tutti i contatti che il soggetto ha avuto con differenti soggetti. Questo è utile per l'analisi di personal computer.



10.6.11 Timeline View

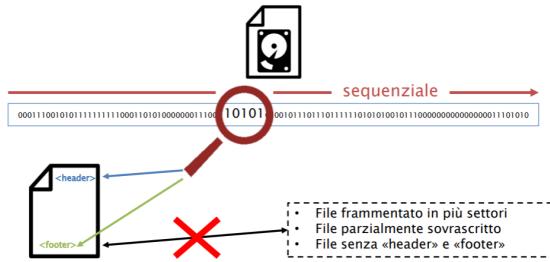
Un'altra visualizzazione di un'analisi può essere di tipo temporale, ovvero vengono elencati tutti i file in ordine cronologico, non dipendenti solo dalle informazioni temporali del file system, ma bensì di tutte le analisi che si sono compiute e poi vengono unite in una **Super Timeline**.



10.7 Toolkit - Altri Strumenti

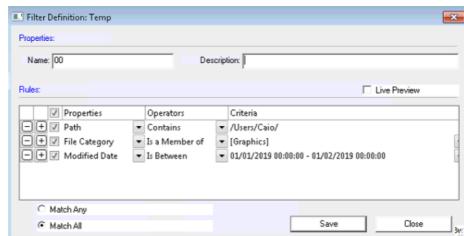
10.7.1 File Carving

Un altro strumento messo a disposizione dai toolkit è il recupero dei file non più residenti nel file system. Questa operazione avviene leggendo in maniera sequenziale il disk image, bit per bit, alla ricerca di un header, ovvero la signature di quella tipologia di file che si sta cercando, quei bit indicheranno l'inizio di quel possibile file, infine si cerca il footer per capire dove termina. Non sempre però esiste un header o un footer, il file potrebbe essere sovrascritto e non completamente leggibile.

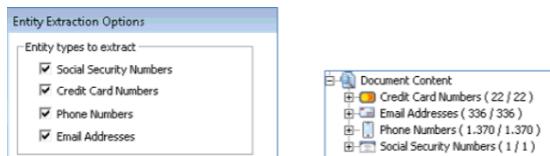


10.7.2 Ricerche semi-manuali

Ricerche tramite attributi, filtrando tra i vari risultati.

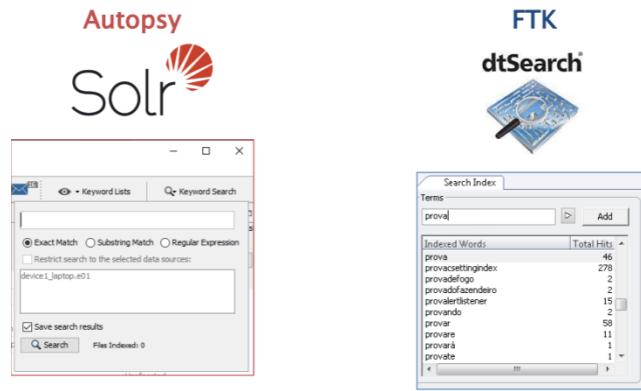


Oppure ricerche tramite **espressioni regolari**.



10.7.3 Indicizzazione

Un processo di analisi piuttosto complesso è l'**indicizzazione** in quanto ci permette di cercare il file attraverso il suo contenuto, tramite parole chiave all'interno del file ricercato.



10.8 Ancora altri Strumenti

- Strumenti di **Decrypt** qualora sui file o sui dischi ci fossero delle chiavi di cifratura il forense deve adoperare tool di decriptazione.
- Strumenti di **Malware Analysis**, senza l'utilizzo di anti-virus si ricercano file noti per essere potenzialmente dannosi.
- Strumenti di **Processign Image** che permettono di riconoscere oggetti all'interno di file grafici.
- Strumenti di **Traduzione** qualora ci fosse la necessità di tradurre i dati da una lingua straniera.

10.9 Export/Report Toolkit

Per esportare i file di interesse, essi vengono raggruppati secondo dei criteri che variano a seconda del software, poi possono essere esportati. Entrambi gli strumenti visti permettono di fare della reportistica riportando tutte le informazioni dei file in *pdf*, *html*, *xml*, in modo da poterle stampare e mostrare alle autorità giudiziarie.

The image displays two screenshots of forensic analysis tools, FTK and Autopsy, illustrating their export/report features.

FTK: The top-left screenshot shows the "File List" interface. It includes a tree view of files under "Name" and a detailed view of selected files. A context menu is open over a file named "OneDrive.rpt", listing options like "Properties", "View File in Directory", and "Export selected rows to CSV". Below this, another context menu is shown over a "Tags" node, including "File Tags (11)", "Evidence Items (2 / 2)", and "Unchecked items (349.139 / 349.139)".

Autopsy: The top-right screenshot shows the "Autopsy" interface. It features a "Report Navigation" sidebar on the left and a main "Autopsy Forensic Report" pane. The report pane displays case information (Case: Case1, Number of Images: 3, Examiner: Marco) and "Image Information" (Screenshot). A context menu is open over a "File Tags (11)" node in the "Tags" section, offering options such as "Add File Tag", "Remove File Tag", and "Add/Edit Central Repository Comment".

Bottom: The bottom screenshot shows a side-by-side comparison of the two interfaces. On the left is the "Autopsy" navigation bar, and on the right is the "FTK Case Report" interface. Both panes show various sections of the forensic report, such as "Case Information", "Case Summary", "File Properties", and "Selected Registry Types".

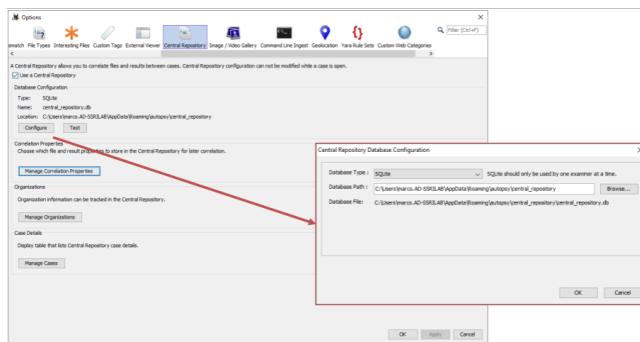
11 Lezione 12

11.1 Autopsy - Configurazione

Una volta scaricato autopsy dal sito ufficiale, la configurazione può essere fatta in **Single User** ovvero la modalità in cui tutti i servizi vengono configurati ed installati su una sola macchina, oppure in modalità **Multi User** che permette di aprire i casi di analisi da più utenti nello stesso momento e permette l'elaborazione automatica delle copie forensi. Questa ultima opzione dovrebbe permettere un'analisi più veloce in quanto abbiamo un'infrastruttura più grande.

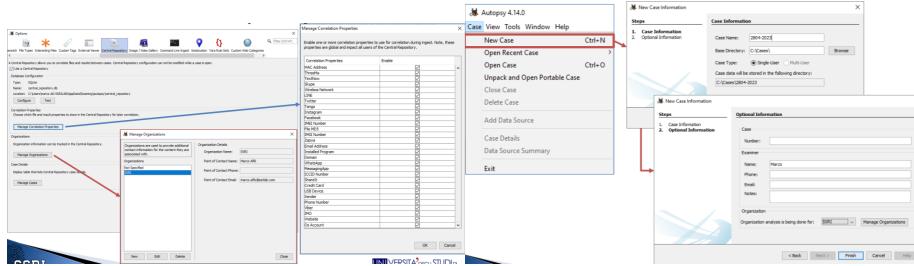
11.1.1 Central Repository

Autopsy ha la caratteristica di avere un **central repository** che non è altro che un database in cui vengono memorizzate le informazioni di casi precedentemente analizzati, ad esempio è in grado di riconoscere se un file è già stato rinvenuto oppure evidenzia i **notable file**. Questo approccio rende il database del caso corrente più leggero.

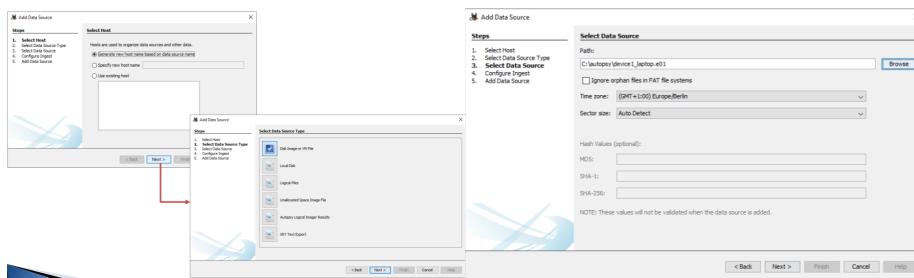


Nella schermata del **Central Repository** abbiamo un'opzione avanzata **Manage Correlations Properties** che contiene tutte le informazioni che il database conserva dei precedenti casi. Inoltre con l'opzione **Manage Organization** possiamo creare gruppi di lavoro multi-utente.

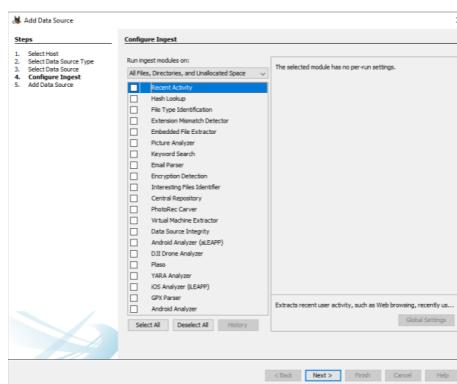
11.1.2 Creazione del Caso



Recandoci in alto a sinistra nella schermata principale del software, **Case >> New Case** possiamo creare un nuovo caso e specificare la **Base Directory** che è dove verrà salvato il database del caso. Possiamo poi completare con informazioni opzionali e cliccare su **Finish**.



Successivamente verrà chiesto di selezionare il disk image che si vuole analizzare. Prima il formato e poi il file vero e proprio.

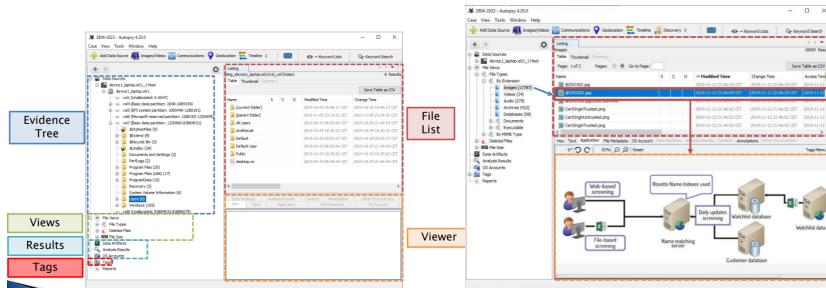


Infine attraverso gli **Ingest Module** possiamo specificare varie opzioni di analisi.

11.1.3 Ricordiamo i file supportati da Autopsy

Disk Image:	Volume:	File System:
‣ Encase E01	‣ DOS	‣ FAT
‣ Raw (DD, BIN, IMG)	‣ GPT	‣ ExFAT
‣ Virtual Disk (VMDK, VHD)	‣ MAC	‣ NTFS
	‣ BSD	‣ EXT2FS
	‣ Solaris	‣ EXT3FS
		‣ EXT4FS
		‣ APFS
		‣ HFS, HFS+
		‣ YAFFS2

11.2 Autopsy - Interfaccia



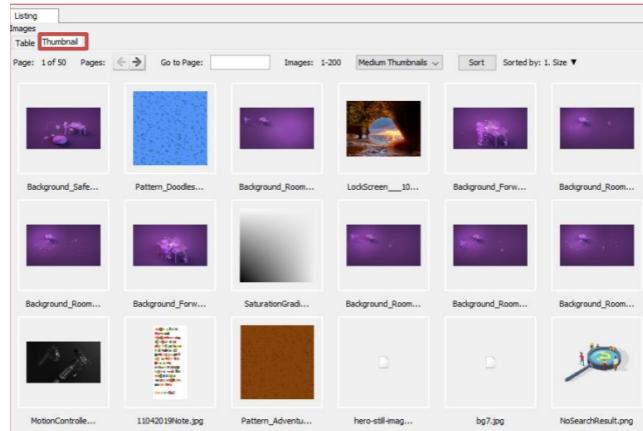
In parte simile alla schermata di FTK, notiamo però che senza applicare delle view, Autopsy ci raggruppa per categoria i file. Le opzioni di listing possono essere di due categorie:

- **Table:**

Listing					
Images					
Table					
Name	S	C	O	Modified Time	Change Time
IMG_20191023_170347.jpg				2019-11-01 23:13:51 CET	2019-11-01 23:33:34 CET
IMG_20191023_092858.jpg				2019-11-01 23:13:52 CET	2019-11-01 23:33:34 CET
IMG_20191024_155744.jpg				2019-11-01 23:13:49 CET	2019-11-01 23:33:34 CET
IMG_20191023_142721.jpg				2019-11-01 23:13:53 CET	2019-11-01 23:33:34 CET
Spotify_FirstRun_Header.png				2019-11-05 01:07:22 CET	2019-11-05 01:07:22 CET
guest.bmp				2019-03-19 05:49:34 CET	2019-10-29 21:46:44 CET
user.bmp				2019-03-19 05:49:34 CET	2019-10-29 21:46:44 CET

- **S:** sta per **SCORE** e indica un file importante o di interesse.
- **C:** sta per **COMMENTS** ed indica se il file è stato commentato.
- **O:** sta per **OCCURENCES** ed indica il numero di volte in cui è stato rinvenuto il file.

- **Thumbnail:**



11.3 Moduli di Elaborazione - Ingest Module

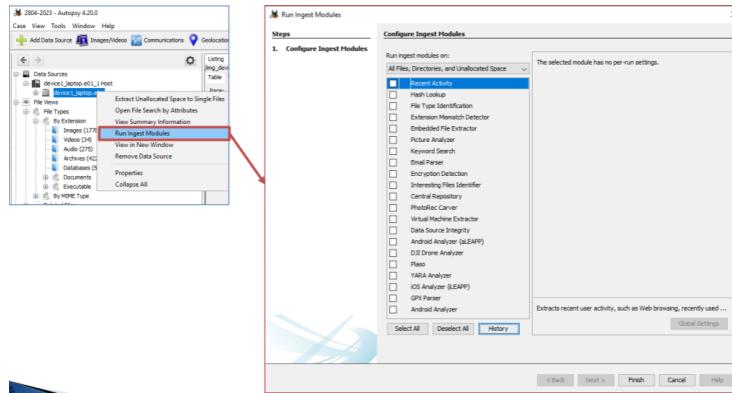
I **moduli di elaborazione** non sono altro che dei **plug-in** che analizzano i dati all'interno del disk image. Alcuni di questi plug-in si occupano di:

- **Hashing**
- **Identificazione del File Type**
- **User Activity**
- **Indexing**
- **File Carving**

L'ingest module è gestito dall'**Ingest Manager** che è colui che gestisce i vari processi di analisi lanciati in background. L'**ingest manager** inoltre ha il compito di processare i file in base a delle priorità stilando una lista di file potenzialmente di maggior interesse, queste sono:

- **Cartelle Utenti.**
- **Programmi e cartelle nella root.**
- **Cartella del Sistema Operativo.**
- **Spazio non allocato.**

Gli ingest module su più file immagine vengono eseguiti parallelamente.

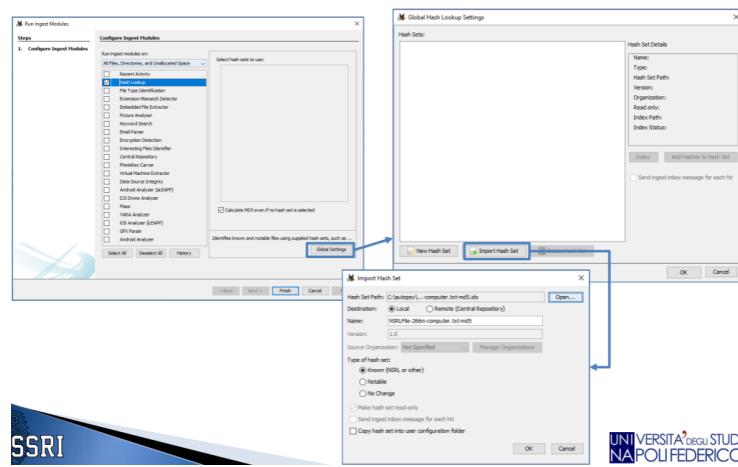


11.3.1 Ingest Module - Hash Lookup

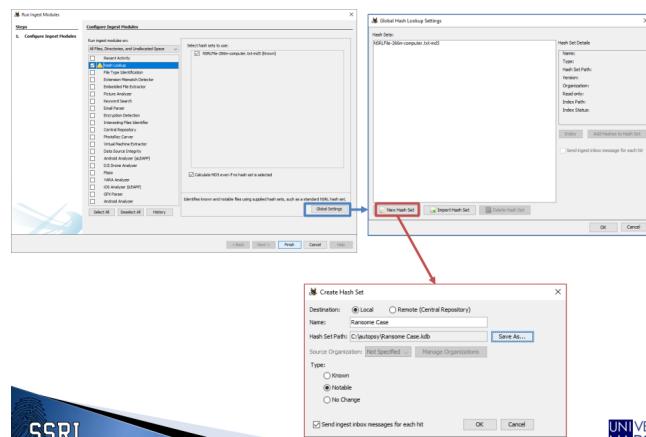
L'**hash lookup** si occupa di calcolare l'hash **MD5** per ogni file del disk image e poi memorizzarne le informazioni risultanti nel **Case Database**. Esso si occupa inoltre di ricercare hash già calcolati

all'interno di una **Known Hash**, ovvero una hash list comprendente **Ignorable File** oppure **Notable File**. Ogni file, al termine dell'ingest module, verrà identificato con tre valori di **Known Status**, che sono:

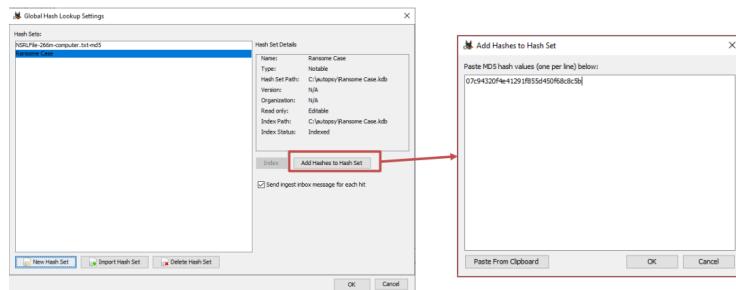
- **Unknown** (default): Hash calcolato ma nessun match trovato.
- **Known** (ignorable): File non di interesse, hash noto.
- **Notable** (known bad): file di interesse, hash noto.



Un vantaggio che abbiamo applicando questo tipo di confronto tra hash è quello di ignorare i file classificati come **Known** anche dagli altri moduli, possono essere nascosti alle views ed anche dalla vista ad albero, questo velocizza notevolmente l'analisi. Ovviamente oltre a poter cercare per file **ignorable**, Autopsy ci permette di fare il contrario, ovvero confrontare per cercare file **notable**.



Si segue la stessa procedura, spuntiamo però in questo caso *Notable*. In questo esempio viene usato un hashset creato da noi, personalizzato, qualora si volessero cercare degli specifici hash trovati già in precedenza.



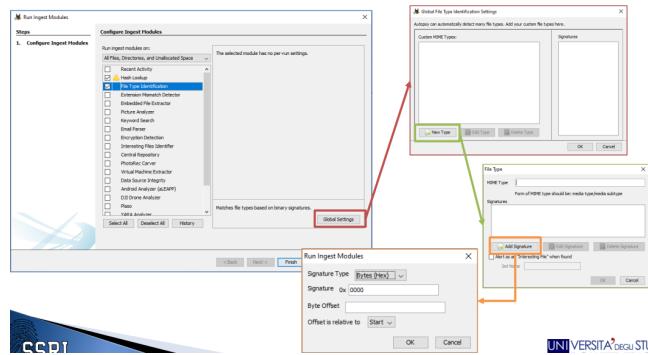
Qualora volessimo aggiungere uno specifico hash all'hashset personalizzato possiamo farlo tranquillamente, se ad esempio volessimo cercare uno specifico file.

The image shows the Autopsy 4.2.0 interface. On the left, the tree view shows a folder named 'Hash Set' containing 'Ransom Case'. In the main pane, a table titled 'Ransom Case' lists a single row: 'Source Name' f_000239, 'S' 0, 'C' 0, 'MD5 Hash' 0794320f4e4291fb55d450f68dc5b, 'Comment' [redacted], and 'File Path' [redacted]. Below the table is a preview area showing a small image of a document. At the bottom, there are tabs for Hex, Text, Application, File Metadata, OS Account, Data Artifacts, Analysis Results, Context, Annotations, and Other Occurrences.

Il risultato della ricerca tramite Hash Lookup sono due file che matchano l'hash inserito manualmente, questi sono due **Notable File**.

11.3.2 Ingest Module - File Type Identification

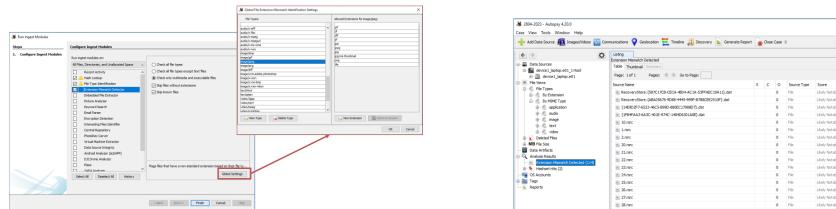
Questa tipologia di modulo non fa altro che identificare i file attraverso la loro **signature** e non solo attraverso l'estensione del file. Ad esempio, la firma **0xFFD8** rappresenta i file in formato **JPG**, in questo modo vengono raccolti tutti i file con la stessa firma. I risultati di questo modulo vengono conservati nel **Case Database**, esse sono informazioni utili anche ad altri moduli. Basata sulla libreria **Tika**, usa una catalogazione **Mime Type** [*Application/Zip, Audio/Mpeg*]



Possiamo decidere se ampliare questa libreria aggiungendo ulteriori tipologie di file non presenti.

11.3.3 Ingest Module - Extension Mismatch Detector

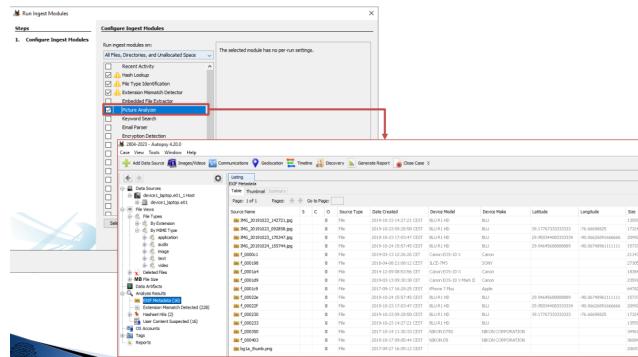
Dipendente dal modulo precedente, questo modulo confronta la coerenza tra l'estensione del file e la sua signature, ovvero la categoria del file. Qualora ci fosse un mismatch potremmo ipotizzare che l'utente abbia provato a nascondere qualcosa.



Ricordarsi però di prendere con le pinze ciò che troviamo nel risultato di questo modulo, questo perché non sempre risulta essere veritiero.

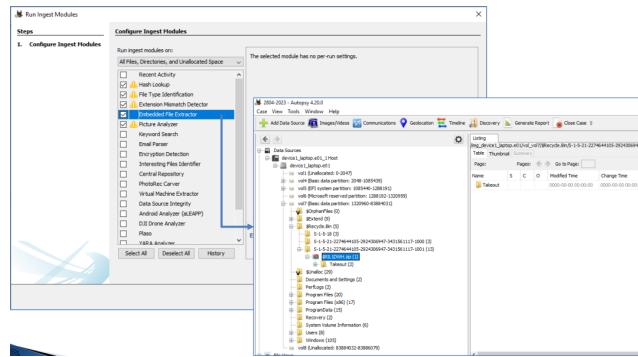
11.3.4 Ingest Module - Picture Analyzer

Questo modulo si occupa di analizzare tutti i file grafici, principalmente **JPG** e ne estrae i metadati **EXIF**. Questo modulo converte formati **HEIC/HEIF** in **JPG** automaticamente.



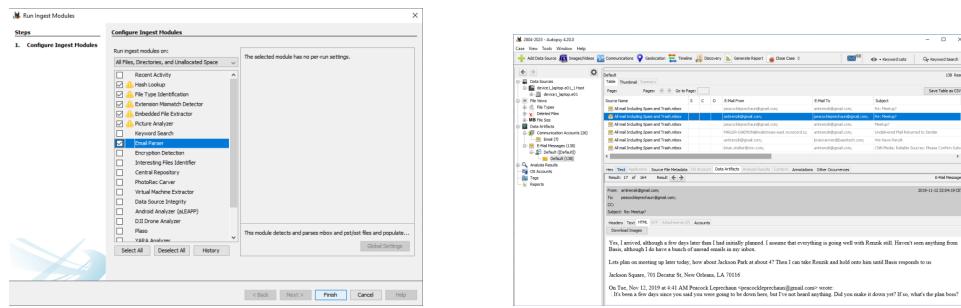
11.3.5 Ingest Modulo - Embedded File Extractor

È un modulo che serve ad estrarre i file incapsulati in altri file, ovvero dagli archi (come **ZIP**, **RAR**, **7ZIP**) compressi. Questo modulo estraie anche file grafici da documenti **office** o **pdf**. I file estratti vengono salvati nel **Case Folder** e saranno visibili nella vista ad albero. Alcune volte gli archivi sono protetti da password, allora **Autopsy** segnala subito quel file rendendo noto il fatto che non ha potuto decomprimere quell'archivio perché protetto.



11.3.6 Ingest Module - Email Parser

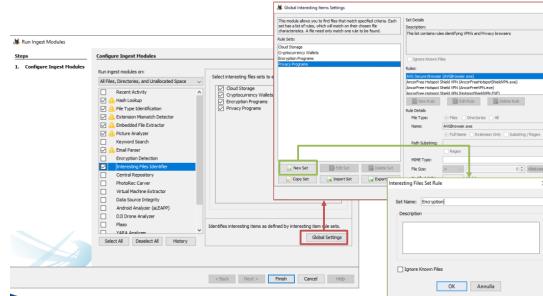
Questo modulo ricerca ed analizza tutti gli archivi di posta, scom-pattandoli, insieme anche ad eventuali allegati, e mostrandoli correttamente nella sezione **Data Artifacts** nella categoria *e-mail messages*. È inoltre possibile analizzare intere conversazioni tra due utenti che si sono scambiati un allegato grazie al raggruppamento in **threads** degli allegati.



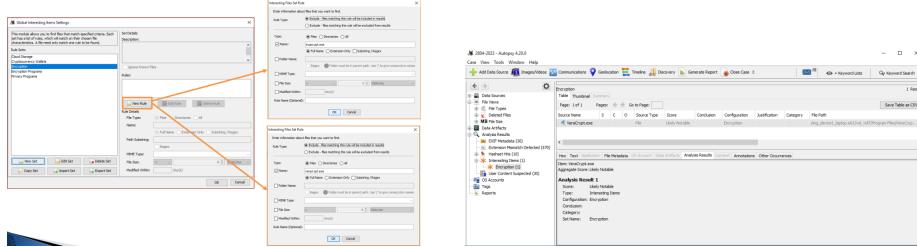
11.3.7 Ingest Module - Interesting Files Identifier

Questo modulo ci permette di etichettare file di interesse, non attraverso l'hash questa volta, ma bensì attraverso il settaggio di alcuni parametri. Il ritrovamento di questi file viene notificato, tra i file di possibile interesse abbiamo:

- **iPhone Backup.**
- **VMWare Image.**
- **Bitcoin Wallet.**
- **Cloud Storage Client.**



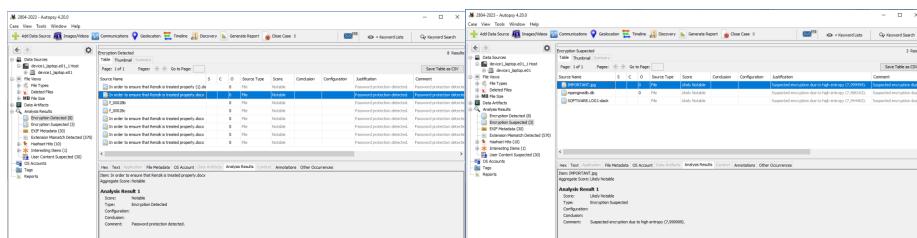
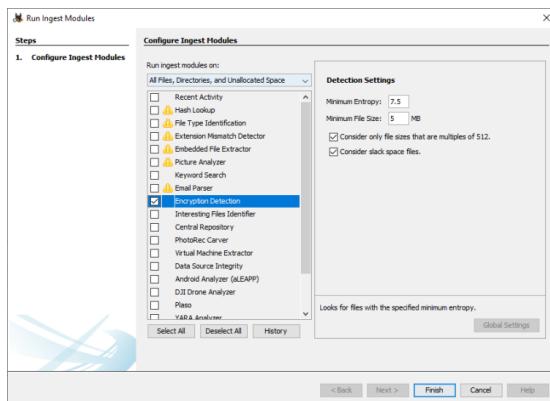
Per iniziare possiamo creare un nuovo set di regole, innanzitutto gli diamo un nome.



Subito dopo si specificano i parametri per le regole del file che vogliamo cercare.

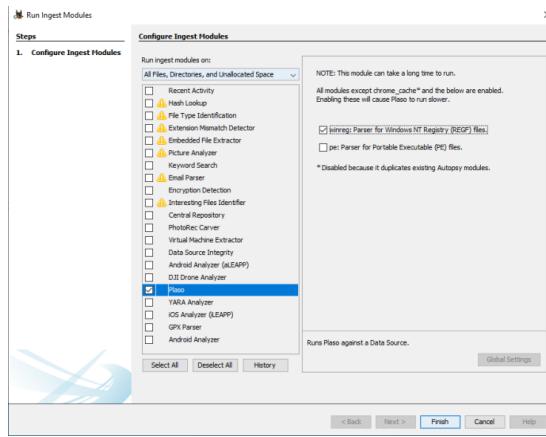
11.3.8 Ingest Module - Encryption Detection

Questo modulo permette di etichettare file e volumi che sono o potrebbero essere cifrati. Il modulo non permette però di decifrare, quindi deve essere accompagnato, se voluto, da un software di decifratura. File o volumi con **High Entropy**, oppure file con dimensioni un multiplo di 512, queste sono tipologie di file che questa analisi potrebbe portare alla luce.



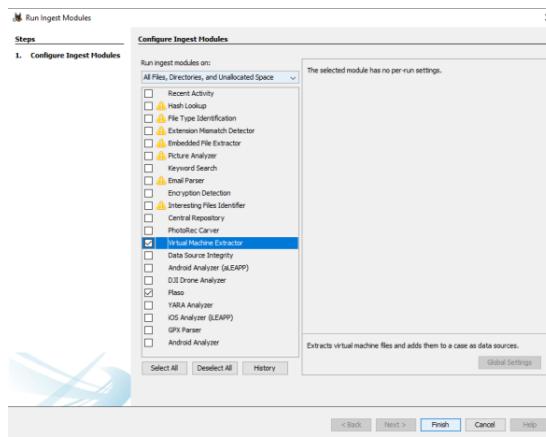
11.3.9 Ingest Module - Plaso

Plaso è un tool opensource che viene integrato in un modulo di **Autopsy**, esso permette di effettuare il parsing di file di log, ed altri tipi di file, ed estrarne il **timestamp**, estrae più timestamp possibile e forma poi una timeline.



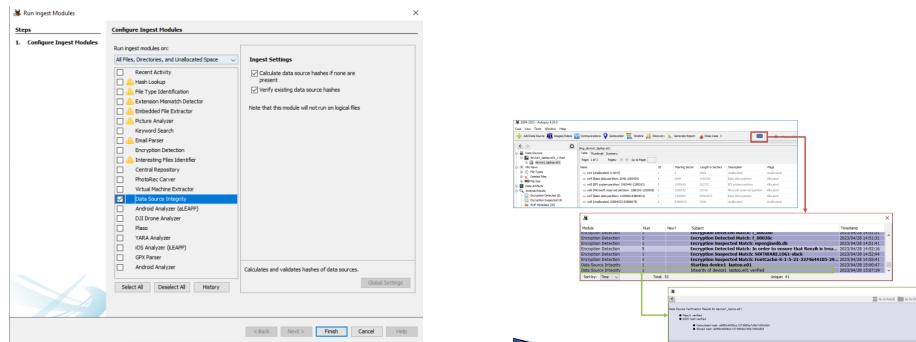
11.3.10 Ingest Module - Virtual Machine Extractor

Modulo che permette la ricerca di file rappresentanti macchine virtuali, ne viene poi fatta una copia e viene inserita nel **DataSource** del caso, questo perché in fin dei conti si tratta sempre di un disk image il quale può essere analizzato nuovamente.



11.3.11 Ingest Module - Data Source Integrity

Modulo che permette di calcolare e poi validare l'hash del reperto, ovvero ci assicura l'integrità dell'evidence. Recupera l'hash qualora fosse stato prodotto in fase di creazione del disk image, poi calcolare l'hash del disk image e ne fa il confronto, in caso di fallimento notifica con un alert l'utente.



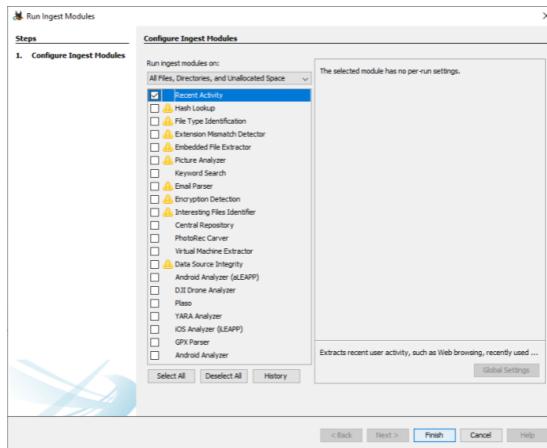
12 Lezione 13

12.1 Ingest Module - Recent Activity

Questo modulo serve a raccogliere principalmente informazioni sull'attività dell'utente col sistema. Estraie dati partendo dall'analisi di:

- **Web Browser** (Cronologia, Cookie, Download, ...).
- **Registri di Sistema**: Dispositivi USB, lista degli utenti, programmi installati ed eseguiti.
- **Cestino**.

I risultati di queste varie analisi sono messi in *Extracted Content*:



12.1.1 Artefatti Web

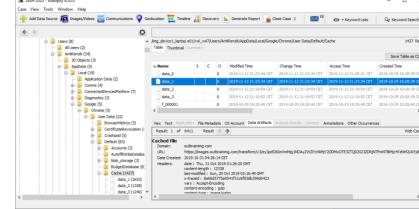
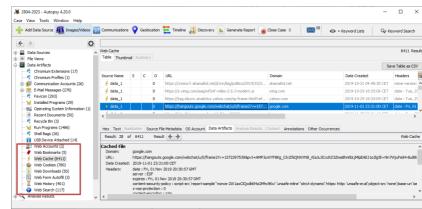
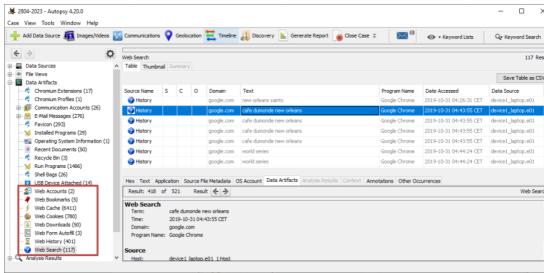
	History	Cookie	Bookmark	Download	Cache	Auto Fill
Chrome	X	X	X	X	X	X
Firefox	X	X	X	X	-	X
IE/Edge	X	X	X	-	-	-
Safari	X	X	X	X	-	-

I risultati dei vari browser vengono uniti, il resoconto è questo:

The image contains four separate windows of the Autopsy 4.2.0 forensic analysis tool, each displaying search results for different types of browser artifacts. The windows are arranged in a 2x2 grid.

- Top Left Window:** Shows search results for 'OS Account' artifacts. It lists two entries: 'OS Account' and 'OS Account - Google Chrome'. Both entries have a 'Date Created' of 2018-11-22 22:45:41 and a 'Source Name' of 'Google Chrome'. The 'Type' column shows 'Recent Activity' for both.
- Top Right Window:** Shows search results for 'Program' artifacts. It lists three entries: 'Program' (Date Created 2018-11-22 22:45:41), 'Unlocked' (Date Created 2018-11-22 22:45:41), and 'Drop off' (Date Created 2018-11-22 22:45:41). The 'Source Name' column shows 'Google Chrome' for all three.
- Bottom Left Window:** Shows search results for 'File' artifacts. It lists two entries: 'File' (Date Accessed 2018-11-22 22:45:41) and 'History' (Date Accessed 2018-11-22 22:45:41). The 'Source Name' column shows 'Google Chrome' for both.
- Bottom Right Window:** Shows search results for 'Recent History' artifacts. It lists two entries: 'Recent History' (Date Accessed 2018-11-22 22:45:41) and 'History' (Date Accessed 2018-11-22 22:45:41). The 'Source Name' column shows 'Google Chrome' for both.

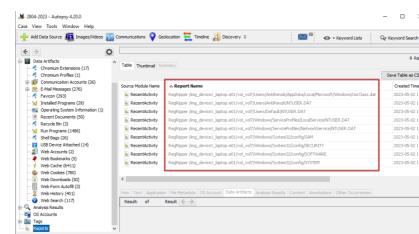
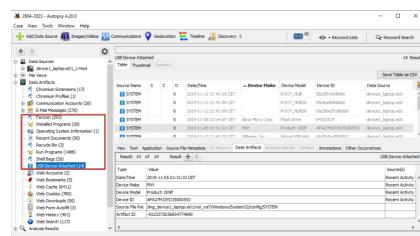
Web Search - Risultati per ricerca



12.1.2 Analisi dei Registri

Questo modulo oltre a fare l'analisi degli artefatti web si occupa dell'analisi dei registri di sistema. Esegue analisi delle chiavi di registro mediante l'uso di **Regripper**, un tool opensource inglobato in **Autopsy** che analizza il contenuto dei registri e visualizza i risultati, non è un tool interattivo. I registri di sistema Windows sono: **System, Software, Security, SAM, NTUser**. Lo scopo principale di questa analisi è avere informazioni riguardo:

- Dispositivi USB connessi.
 - Programmi installati ed eseguiti.
 - Informazioni di sistema dell'utente



12.1.3 Recycle Bin

Questo modulo permette anche l'analisi dei file cancellati ma ancora presenti nel cestino. Quando un file viene cancellato, su Windows, il sistema operativo va ad eliminare/perdere il **filename** e va poi a conservare questa informazione in un altro piccolo file, detto **file manifest**, associato ai file cancellati conservando il **filename** originale ed il suo path.

Source Name	S	C	O	Path	Time Deleted	Username	Data Source
4PCSY5.txt				C:\Users\AnFrenck\Desktop\4PCSY5.txt	2019-11-05 23:25:40 CET		
43LZK35Z.zip				C:\Users\AnFrenck\Downloads\43LZK35Z.zip	2019-11-12 21:01:50 CET		
unnamed.jpg				C:\Users\AnFrenck\Downloads\unnamed.jpg	2019-11-12 22:01:29 CET		device_1.jpg

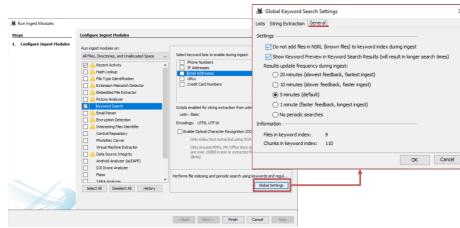
Name	S	C	O	Modified Time	Change Time	Access
In order to ensure that Renkal is treated properly.docx				2019-11-05 01:23:02 CET	2019-11-05 01:23:02 CET	2019-11-05 01:23:02 CET
Renkal - In order to ensure that Renkal is treated properly.docx				2019-11-05 01:23:02 CET	2019-11-05 01:23:02 CET	2019-11-05 01:23:02 CET
unnamed.jpg				2019-11-12 22:06:29 CET	2019-11-12 22:06:29 CET	2019-11-12 22:06:29 CET

Autopsy inserisce anche nella vista ad albero questi file contrassegnandoli come eliminati. Questo per semplificare il lavoro del C.F.

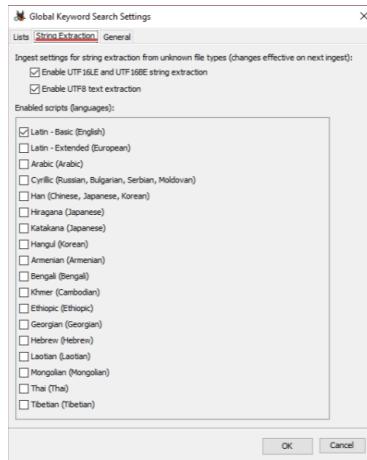
12.2 Ingest Module - Keyword Search

Questo modulo permette di abilitare la ricerca testuale, estraendo ogni parola da ogni singolo file e fa poi un associazione tra le parole rinvenute e l'**ID** del file. Fa uso di **Apache Solr** che è il motore di indicizzazione di **Autopsy** che oltre a memorizzare le keyword che sono nei diversi file, memorizza il testo estratto di file e dagli artefatti (*con artefatti si intende tipo artefatti web*).

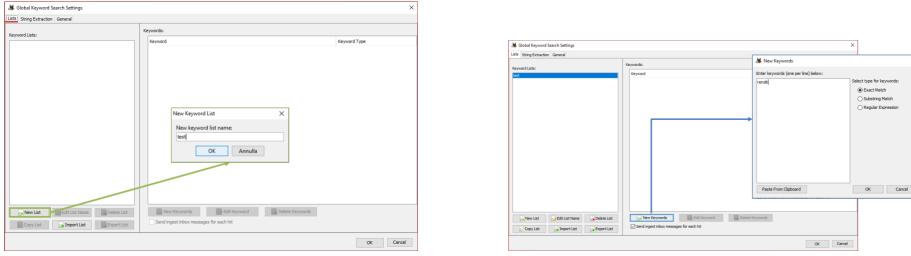
Per l'estrazione del contenuto dei file e dei metadati si fa uso di **Apache Tika** che permette ad **Autopsy** di comprendere correttamente il contenuto di un file per poi confrontarne le keyword. I file non riconosciuti o corrotti sono forzatamente letti con uno **String Extractor**. Si fa inoltre uso di un **HTML Text Extractor** qualora ci fossero dei file HTML da interpretare, si occupa di estrarre anche i commenti e java script. Infine viene eseguita una **normalizzazione**, ovvero vengono fatte ricerche *case sensitive* e viene usato unicode per eliminare problematiche con gli accenti o caratteri particolari.



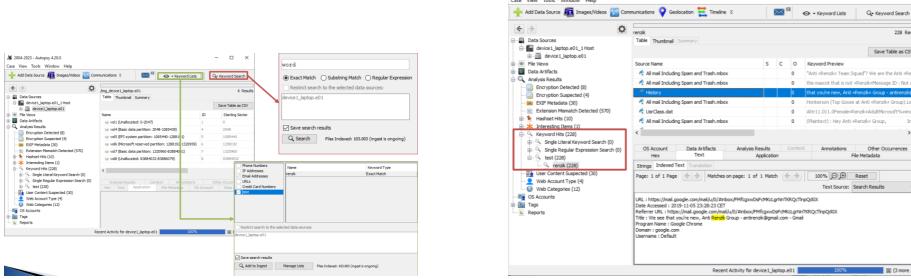
È possibile aggiungere diverse configurazioni, possiamo decidere di escludere i file *Ignorable*, come detto in precedenza.



Possiamo settare anche le impostazioni per lo **String Extractor** come poter decidere l'encoding ed il linguaggio.



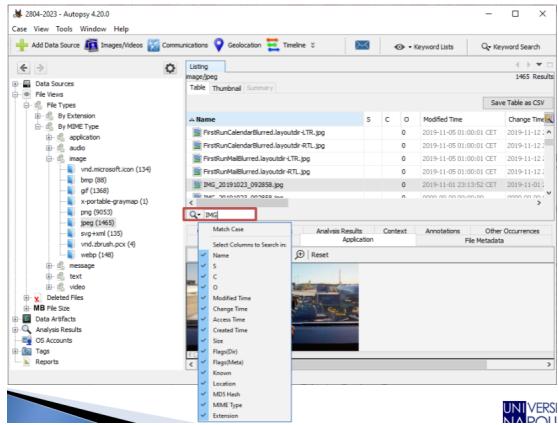
Abbiamo inoltre la possibilità di creare delle liste con delle keywords



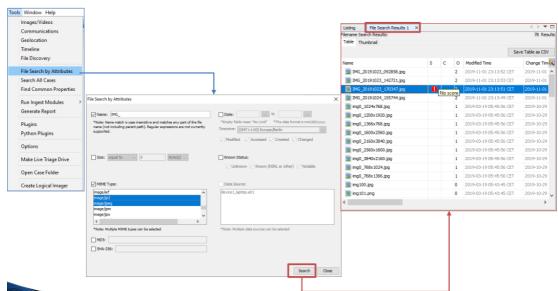
Dopo aver avviato il modulo possiamo attivare la keyword search.

12.3 File Search by Attributes

Senza l'ausilio di alcun modulo, **Autopsy** mette a disposizione la possibilità di ricercare attraverso le informazioni che il software ha già precedentemente messo in formato tabellare.

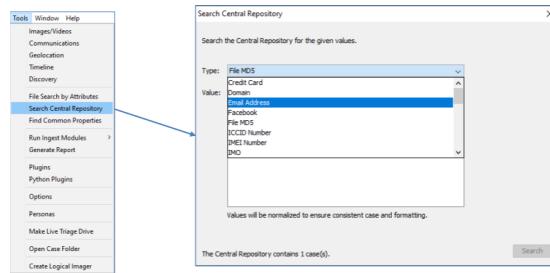


È possibile effettuare questa operazione anche tramite un apposito tool che fornisce qualche informazione in più.



12.4 Search Central Repository

Abbiamo detto che il **Central Repository** è il database principale dove **Autopsy** immagazzina diverse informazioni di tutte le analisi effettuate nel tempo. Qualora volessi cercare info riguardo un precedente caso posso usare il tool di ricerca per il Central Repository.



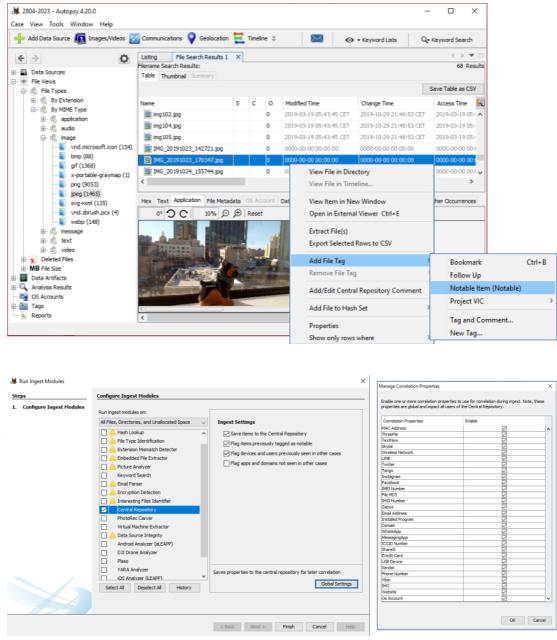
12.5 Ingest Module - Central Repository

Questo modulo prende il nome dal repository centrale di cui fa uso **Autopsy**, il suo scopo però è di correlare il caso corrente con i casi precedentemente analizzati, ovvero ricerca informazioni già note in altra casi e che potremmo rinvenire anche nel caso corrente. Oltre che a cercare informazioni già note, questo modulo permette di tenere aggiornato il central repository, è buona norma infatti, alla fine di una analisi, rilanciare il modulo in modo da aggiungere le nuove informazioni analizzate. Il central repository conserva:

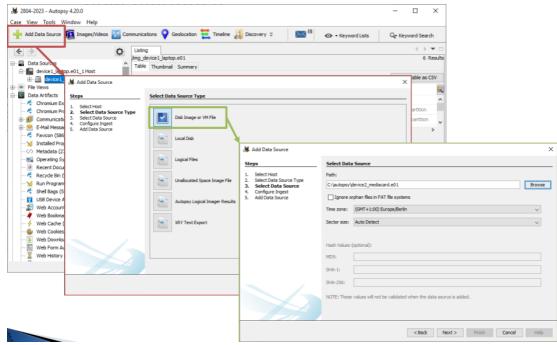
- **Valore.** (Hash, Numero Telefono, E-mail, ecc)
- **Caso.**
- **Data Source:** informazioni sul disk image di provenienza.
- **File Path.**
- **Commento del C.E.**
- **Notable Status:** notable, ignorable.

Aggiunta di un file di interesse.

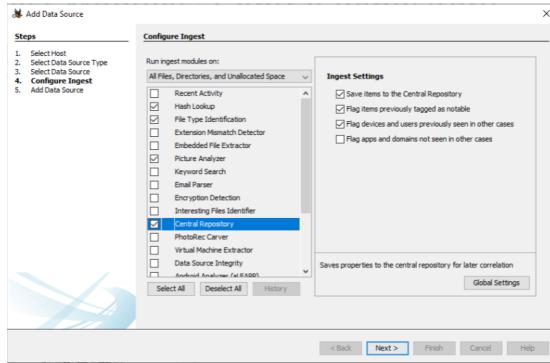
Rilanciando il modulo possiamo aggiungere al nostro central repository le informazioni denotate come di interesse (notable), così da riempire il repository di informazioni aggiuntive del caso.



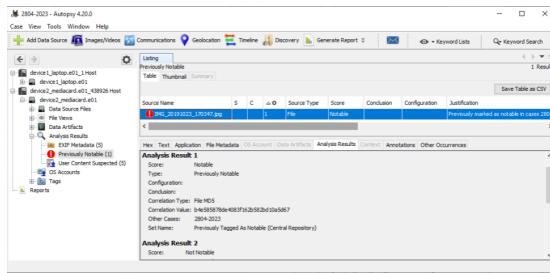
12.6 New Evidence



Con **Autopsy** è possibile aggiungere al caso corrente su cui stiamo lavorando più di un disk image, è anche possibile aggiungerne uno in seguito, in questo modo.



Seguendo i vari step per l'aggiunta si arriva alla sezione dei moduli utili da eseguire per l'analisi. Avendo premura di selezionare moduli utili come **Hash Lookup** e **Central Repository**.



Al termine dell'analisi della nuova evidence possiamo riscontrare il file immagine che avevamo precedentemente contrassegnato come *notable* spuntare nuovamente come match dell'analisi della nuova *evidence*.

12.7 Ingest Module - PhotoRec Carver

Questo modulo prevede il recupero dei file cancellati tramite l'utilizzo di **PhotoRec**, un tool opensource di *Data Carving* che lavora sullo spazio non allocato, analizzando bit a bit tutto il disco senza importarsi delle partizioni e dei file system. Il risultato lo troviamo in **\$CarvedFile**. L'unico contro che troviamo analizzando solo lo spazio non allocato è che a volte alcuni file si perdono e non vengono recuperati dallo spazio allocato.

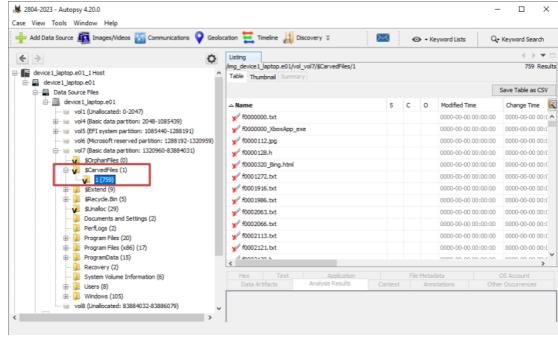
Valid File Types							
The following is the list of valid file types for the version of PhotoRec currently used by Autopsy.							
1cd	caf	dag	gp2	max	p0b	r0d	vfb
3dm	cam	dxf	gp5	mb	p0f	r0x2	vib
7z	catdrawing	odt	gpg	mcd	p0s	sav	vcnk
a	cdr	odp	grv	mdb	p0w	scr	vcng
ab	cha	ods	gzt	mdf	pfx	ses	wallet
ace	class	odt	gtf	mta	plib	sgtca	wdp
accdb	comicdoc	ess	htc	mtf	plt	shb	wim
ace	cov	etx	hdr	mtf	png	st1	win
ado	cp	evtx	inf	mtf	tdm	st2	wks
adfdesign	cpl	evtx	hsp	mg	tdm	st3	
ai	crt	ext	hw	mk3	tdm	st4	
ai2f	csh	ext	http	mk5	tdm	st5	
all	crt	fat	html	mk7	tdm	st6	
ais	crt	fat1	ico	mk8	tdm	st7	
and	d2k	fbs	icos	mov/dvd	tdm	st8	
aer	dad	fcp	ico	mp3	tdm	st9	
ape	dat	fcs	ics	mpg	tdm	st10	
apple	DB	fd5	ifo	mpl	tdm	st11	
apr2	db	fd8	im3	mrw	tdm	st12	
arj	dfb	fh5	info	mp3	tdm	st13	
assF	dhc	fit	iso	mp4	tdm	st14	
atl	des	fits	itc	mpg	tdm	st15	
asm	ddf	flac	itu	myo	tdm	st16	
atd	des	flp	ktu	qdb	tdm	st17	
au	diskimage	flv	lks	nd2	tdm	st18	
axp	djr	fm	json1d4	nes	tdm	st19	
axx	dep	fb3	lrb	n3x	tdm	st20	
bac	doc	fbm	kohs	nk2	tdm	st21	
bde	dpc	fp5	key	nsf	tdm	st22	
bin	dpr	frw	l1t	ogg	tdm	st23	
binvox	diz2	fwesay	l1t	raw	tdm	st24	
DS_Store	ds	frw	lck	rcd	tdm	st25	
dt	ds	fs	lrc	rdt	tdm	st26	
blend	dss	fu4	lso	ref	tdm	st27	
bmp	dst	g4	lso	ref	tdm	st28	
bog	dst	gcr	luk3	rap	tdm	st29	
bvr	dump	gho	lzh	rap	tdm	st30	
bz2	dv	gl	lzo	pcb	tdm	st31	
cdd	dzl	glf	lzo	pet	tdm	st32	
cab	drv	gn*	mat	pxc	tdm	st33	

Configurazione del modulo.

Name	S	C	O	Modified Time	Change Time	Access Time
unnamed_39073_11444442_13488844				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_209710941_011538968				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_380314402_193966752				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_1239610-83884012				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_1935278572_3002374040				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_211002224_2117954048				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_2117954048_2124898972				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_212489872_2421437996				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00
unnamed_39073_212489872_2536179520				2009-09-01 00:00:00	2009-09-01 00:00:00	2009-09-01 00:00:00

Porzioni del disco non allocato sulle quali lavora **PhotoRec**.

Il risultato dell'analisi lo troviamo in questa cartella la quale conterrà tutti i file *carved*.



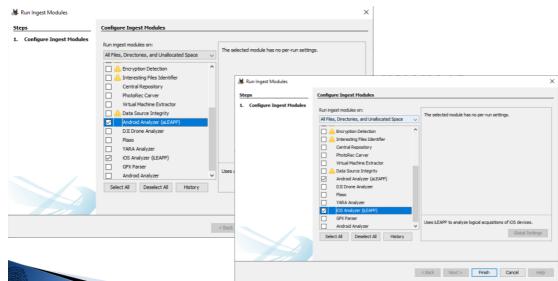
12.8 Ingest Module - Android Analyzer

Questo modulo permette di effettuare l'analisi di Disk Image rappresentanti dispositivi android. Analisi di database android ed app di terze parti. Questo modulo estrae:

- Registro chiamate.
- Contatti.
- Messaggistica.
- Browser.
- Geolocation.

12.9 Ingest Module - iLEAPP, aLEAPP

Questi due moduli per l'analisi dei log vengono adoperati per dispositivi **iOS** ed **Android**.



12.10 Viste Specializzate

12.10.1 Timeline Graphic Interface

I toolkit oltre che raccogliere informazioni ci permettono di visionare le stesse informazioni con viste differenti. Abbiamo accennato delle timeline quando parlammo di **Plaso** che permetteva di avere una visione grafico cronologica dei dati presenti nel disk image. Questa interfaccia permette di visualizzare le attività del sistema organizzate temporalmente, in questo ordine:

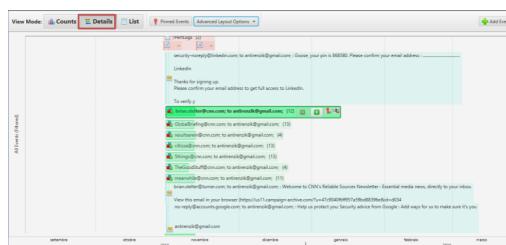
- File Time estratti dal file system.
- Web Activity estratte dalle recent activity.
- EXIF.
- Plaso.
- Altro.

Lo scopo è quello di capire:

- Quando è stato usato il sistema?
- Cosa è accaduto in un certo tempo?
- Cosa è accaduto prima e dopo determinati eventi?



Nella sezione **Counts** abbiamo il numero di eventi in quel periodo.



Nella sezione **Details** invece troviamo ogni singolo evento nello specifico.

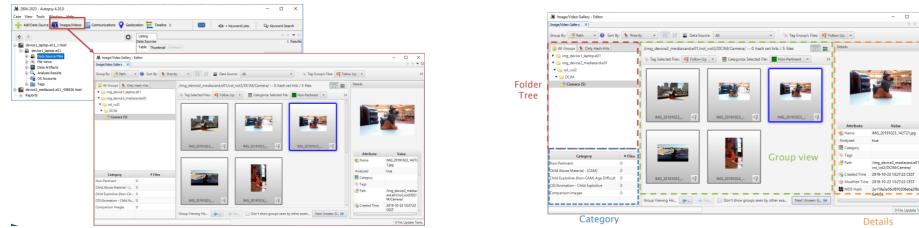
View Mode	Counts	Details	List	Add Event	Snapshot Report	Refresh View
Date/Time	Event Type	Description		Tagged	Hash Hit	395,651 events
2019-09-18 17:49:30	[M]	/Applications/AndroidAppData/Local/Google/Chrome/app_1/data/1/1_a/101744_data/1_a/101744.0				
2019-09-18 19:20:30	[M]	/Applications/AndroidAppData/Local/Google/Chrome/app_1/chromiumCache/3/1/10305381/system.css				
2019-09-20 06:14:04	[M]	/Windows/Win32/_microsoft-windows-services_..._0.0.1832.411,_033537147d72182/pingos.exe				
2019-09-20 06:14:04	[M]	/Windows/WindowsPowerShell/v1.0/powershell.exe				
2019-09-20 06:29:50	[M]	/Windows/Win32/_microsoft-windows-games_gata_1_103821_411,_49c9fdd4ba382/purpletear.dll				
2019-09-20 06:29:57	[M]	/Windows/Win32/_microsoft-windows_ck_m_1_103821_411,_49c9fdd4ba382/t07eCommerce.dll				
2019-09-20 06:29:57	[M]	/Windows/Win32/_microsoft-windows-services_..._103821_411,_033537144d72162/pingos.exe				
2019-09-20 06:29:58	[M]	/Windows/Win32/_microsoft-windows_..._103821_411,_033537144d72162/pingos.exe				
2019-09-20 06:29:58	[M]	/Windows/Win32/_microsoft-windows_..._103821_411,_033537144d72162/pingos.exe				
2019-09-20 06:29:59	[M]	/Windows/Win32/_microsoft-windows_games_gata_1_103821_411,_49c9fdd4ba382/securerecotab.dll				
2019-09-20 06:29:59	[M]	/Windows/Win32/_microsoft-windows_games_gata_1_103821_411,_49c9fdd4ba382/securerecotab.dll				
2019-09-20 06:30:01	[M]	/Windows/Win32/_microsoft-windows_gaming_..._103821_411,_033537144d72162/ChicCore.dll				
2019-09-20 06:30:01	[M]	/Windows/Win32/_microsoft-windows_gaming_..._103821_411,_033537144d72162/ChicCore.dll				
2019-09-20 06:30:04	[M]	/Windows/Win32/_microsoft-windows-services_..._103821_411,_033537144d72162/pingos.dll				
2019-09-20 06:30:04	[M]	/Windows/Win32/_microsoft-windows-services_..._103821_411,_033537144d72162/pingos.dll				
2019-09-20 06:30:14	[M]	/Windows/Win32/_microsoft-windows-services_..._103821_411,_033537144d72162/pingos.dll				
2019-09-20 06:36:27	[M]	/Windows/System32/powershell.exe				
2019-09-20 06:36:27	[M]	/Windows/Win32/_microsoft-windows-services_..._0.0.1832.411,_49c9fdd4ba382/runner.exe				

La sezine *List* invece comprende tutte le informazioni usate per creare l'interfaccia grafica della timeline.

12.10.2 Image Gallery Interface

La possibilità di visualizzare tutti i file grafici in una vista separata, ovvero visualizzarli mediante le anteprime in maniera veloce. Questo tipo di vista mostra il contenuto di una cartella alla volta andando per priorità:

- Numero di Risultati positivi all'Hash.
 - Numero di immagini/video



Bordi

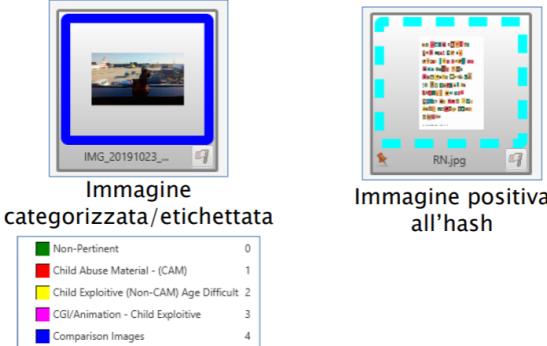
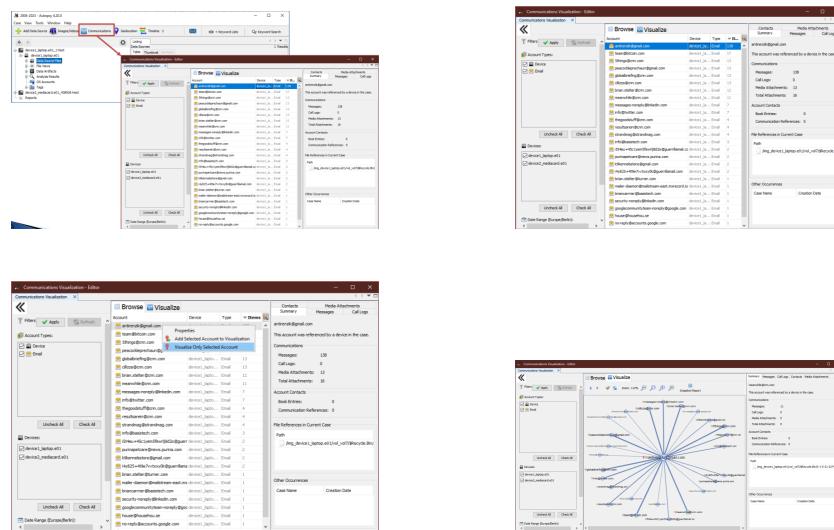


Image Gallery mette a disposizione una caratterizzazione dei bordi per riconoscere più velocemente il tipo di immagine con il quale si ha a che fare.

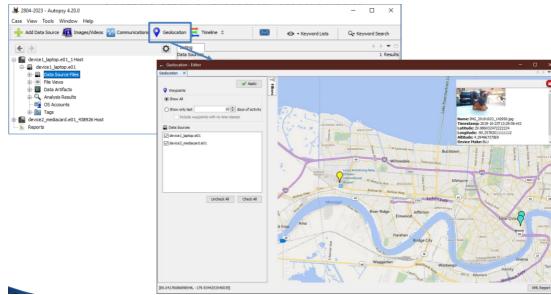
12.10.3 Communication Interface

Questa vista permette di visualizzare i dati delle comunicazioni in maniera differente. Per lo più su dispositivi desktop viene riempita con email tramite l'**email parser**, per dispositivi android la maggior parte è "una comunicazione" essendo il device attivo a comunicare. Questa è una vista orientata agli account, ovvero vengono visualizzate tutte le attività associate ad un account ed in più anche le relazioni con altri account.



12.10.4 Geolocation Interface

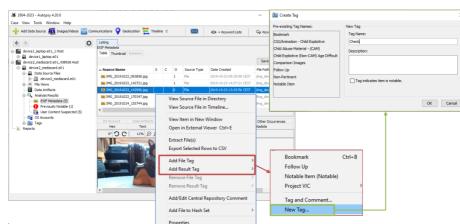
Una vista che riepiloga tutti gli artefatti dai quali sono state estratte informazioni sulla posizione. Per quanto riguarda personal computer per lo più queste info vengono estratte dall'**exif parser** dalle fotografie scattate.



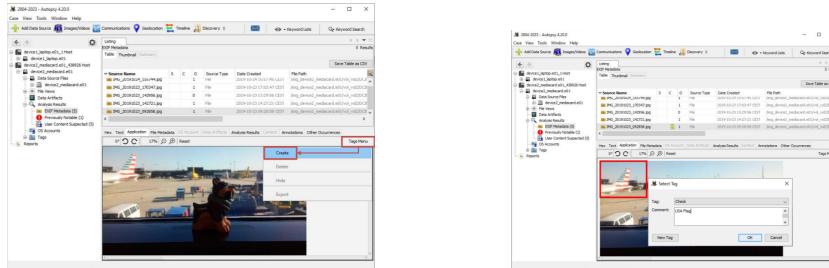
12.11 Tag e Report

12.11.1 Tagging

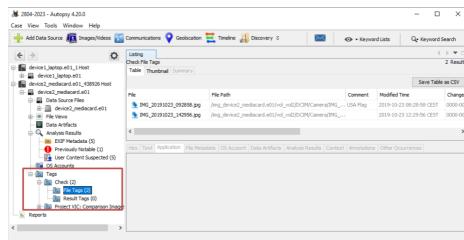
Una volta terminata l'analisi c'è bisogno di esportare i file ritenuti di interesse, per poterlo fare, **Autopsy** mette a disposizione quello che chiamiamo **Tagging**, ovvero creare un riferimento ad un file/item di interesse e permettendo anche di commentarlo. Ogni tag è associato ad un esaminatore in modo da conoscere chi li ha etichettati. L'obiettivo principale del tagging è quello di ritrovare facilmente i file di interesse, evidenziarli ed esportarli in un **Report** finale.



- **Add File Tag:** permette di tagger l'intero file.
- **Add Result Tag:** permette di tagger un'informazione del file. (es: navigazione su un sito)



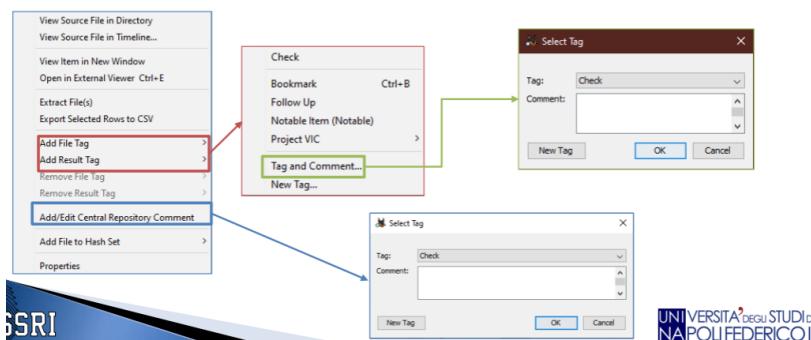
Per quanto riguarda i file grafici è possibile creare dei tag su delle specifiche porzioni di immagine.



Tutti i file taggati si trovano nella sezione *Tags*, sono suddivisi per tipo di tag.

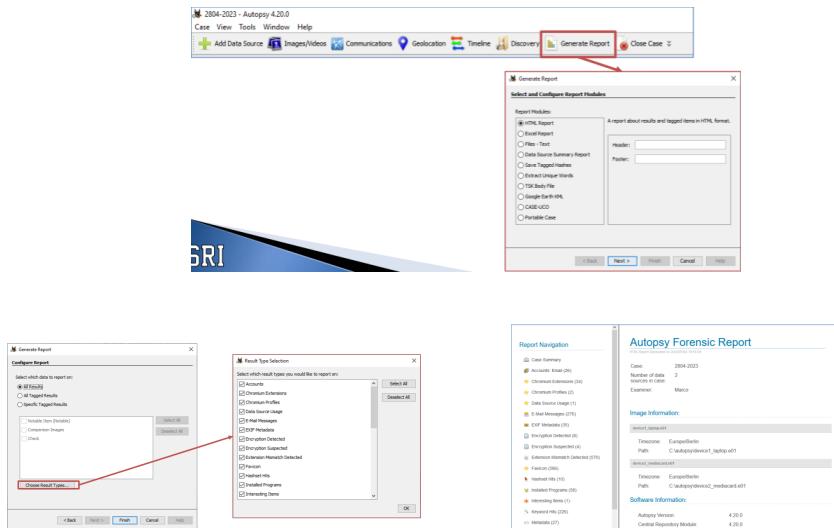
12.11.2 Comments

- ▶ Consente di annotare il motivo dell'interesse di un file\item:
 - Verrà visualizzato nel Report
 - Può essere salvato nel «Central Repository»



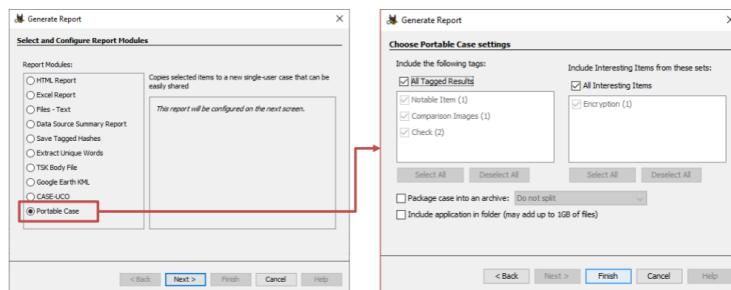
12.11.3 Reporting

Obiettivo principale dopo l'analisi è fare il **Reporting**, ovvero l'export di tutte le informazioni che abbiamo taggato. Lo scopo del report è quello di condividere informazioni. Ci basterà cliccare su **Generate Report** dopo aver taggato tutti i file di nostro interesse e potremmo scegliere diversi formati in cui salvare il nostro file di report.



12.11.4 Reporting - Portable Case

Versione più leggera del caso originale comprendente solo i file etichettati e solo i file nella categoria **Interesting Item**. Questo report ha un database **SQLite** e verrà esportato nel *case folder*.

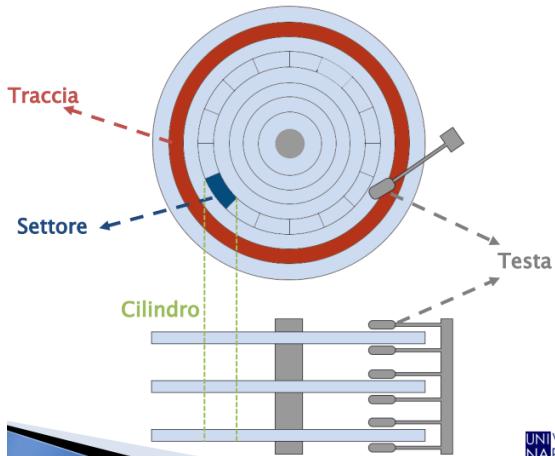


13 Lezione 14

13.1 L'analisi - Il disco



- **Testa:** esegue le operazioni di lettura e scrittura.
- **Braccio:** si muove su ogni piattello.
- **Piattelli:** dischi magnetici dove vengono salvati i dati.



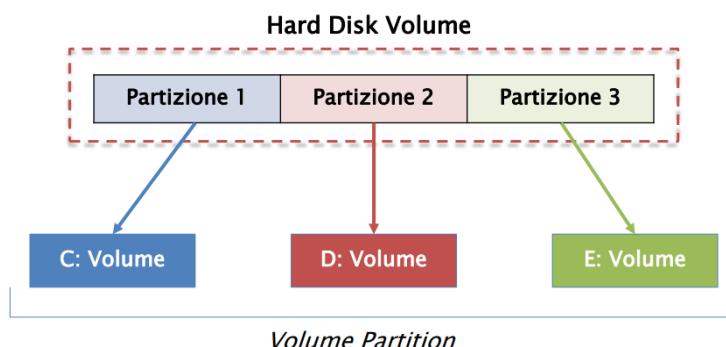
A livello più logico, il disco, è diviso in:

- **Tracce:** è un anello intorno al piattello che ha indirizzo di partenza "zero", più esternamente, andando ad incrementare verso l'interno ed arrivando poi al centro.
- **Settori:** sono la parte più piccola indirizzabile (dimensione minima: 512 byte). Ogni settore ha un indirizzo che è identico per ciascuna traccia. Per identificare un settore devo prima

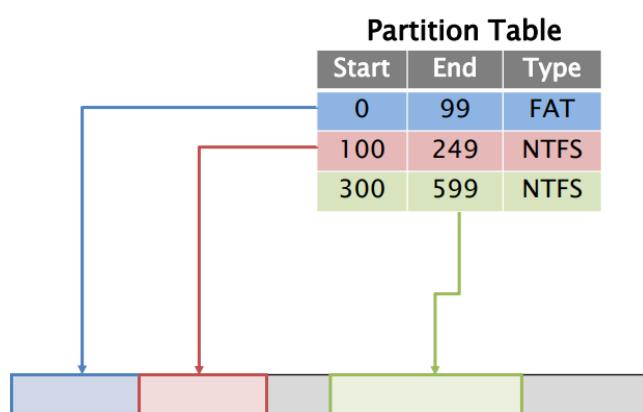
decidere l'indirizzo del cilindro, poi il numero della testa ed infine l'indirizzo del settore. Questo tipo di indirizzamento è chiamato **Logical Block Address (LBA)**.

13.2 L'analisi - I volumi

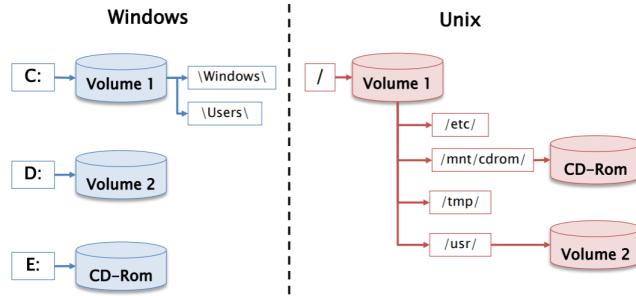
Con analisi intendiamo cercare la struttura dati che è all'interno del volume e che coinvolge le operazioni di partizionamento o di unione, che è possibile eseguire sui dischi, il **Volume System** gestisce l'esecuzione di questi due obiettivi. Il **Volume** è un insieme di settori per memorizzare dati. Una **Partizione** invece è un insieme di settori consecutivi in un volume.



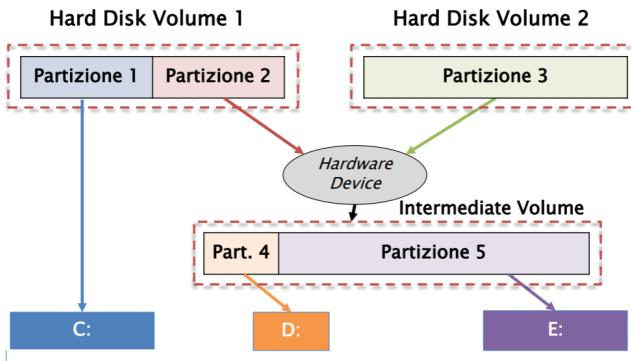
I comuni sistemi di partizionamento fanno uso di una o più tabelle di partizionamento dove ogni **entry** indica l'inizio e la fine della partizione, specificando, ad esempio, il settore di inizio e di fine della partizione e poi anche il tipo di file system.



Lo scopo di un sistema di partizionamento è quello di organizzare la disposizione del volume.



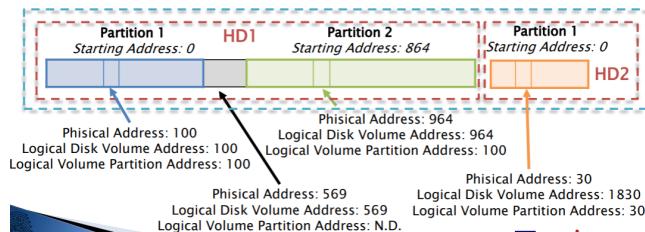
Ogni sistema operativo fa uso di un sistema di partizionamento.
Ad esempio Unix fa un uso diverso dei volumi rispetto a Windows.



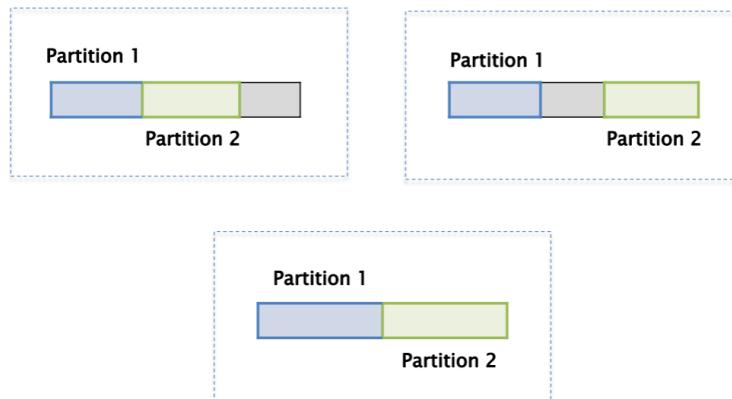
Sistemi più grandi e più complessi fanno uso, oltre al partizionamento, anche dell'unione dei volumi, ovvero presentano più dischi come se fossero un unico disco. Principalmente serve ad aggiungere ridondanza per sopperire in caso di problemi di un disco, ovvero i RAID, e poi per aumentare lo spazio di archiviazione.

13.2.1 Indirizzamento dei settori

- **Physical Address (L.B.A):** l'indirizzo del settore è calcolato in base al primo settore del disco.
- **Logical Disk Volume Address:** l'indirizzo del settore è calcolato in base al primo settore del volume.
- **Logical Volume Partition Address;** l'indirizzo del settore è calcolato in base al primo settore della partizione.



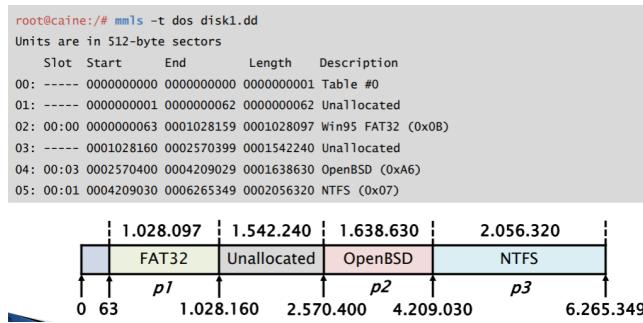
Uno dei principali motivo per analizzare un **volume system** è quello di controllare la partizione basandosi sulle altre già presenti. Ad esempio, verificando che non ci siano altre partizioni nascoste all'interno del volume, controllando tra la fine dell'ultima partizione e la fine del volume, e poi controllando tra la fine della prima partizione e l'inizio dell'ultima.



13.3 Esempi più pratici

13.3.1 La lista delle partizioni in un file immagine

Tramite l'ausilio di un tool, disponibile nella maggior parte delle distro forensics oriented, che permette di listare tutte le partizioni presenti nel disk image, a differenza di altri comandi come **fdisk**, questo ci permette di visualizzare anche i settori che non sono stati assegnati ad alcuna partizione. Questo tool è **mmls**, che viene adoperato in questo modo:



13.3.2 Estrazione delle partizioni in un file immagine

Facciamo uso del comando **dd** ed utilizzando le opzione *count* e *skip*. Qualora il sistema di partizionamento fosse corrotto o danneggiato ci sono dei tool che permettono di ricostruire il sistema di partizionamento attraverso la signature del file system che si trova su quella partizione.

```
root@caine:/# mmls -t dos disk1.dd
Units are in 512-byte sectors
      Slot Start      End      Length     Description
00: ----- 0000000000 0000000000 0000000001 Table #0
01: ----- 0000000001 0000000062 0000000062 Unallocated
02: 00:00 0000000063 0001028159 0001028097 Win95 FAT32 (0x0B)
03: ----- 0001028160 0002570399 0001542240 Unallocated
04: 00:03 0002570400 0004209029 0001638630 OpenBSD (0xA6)
05: 00:01 0004209030 0006265349 0002056320 NTFS (0x07)

root@caine:/# dd if=disk1.dd of=disk1_p1.dd bs=512 skip=63 count=1028097 [p1]
root@caine:/# dd if=disk1.dd of=disk1_p2.dd bs=512 skip=2570400 count=1638630 [p2]
root@caine:/# dd if=disk1.dd of=disk1_p3.dd bs=512 skip=4209030 count=2056320 [p3]
```

13.3.3 Recupero delle partizioni danneggiate in un file immagine

Uno dei tool che permette il recupero di partizioni che hanno perso il sistema di partizionamento è **gpart**. Esso riesce ad identificare i diversi file system tramite dei test su alcuni settori specifici.

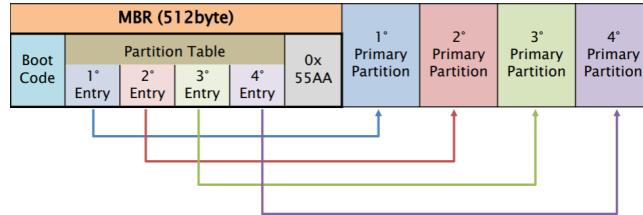
```
root@caine:/# gpart -v disk2.dd
* Warning: strange partition table magic 0x0000.
[. . .]

Begin scan...
Possible partition(DOS FAT), size(800mb), offset(0mb)
  type: 006(0x06)(Primary 'big' DOS (> 32MB))
  size: 800mb #s(1638566) s(63-1638628)
  chs: (0/1/1)-(101/254/62)d (0/1/1)-(101/254/62)r
  hex: 00 01 01 00 06 FE 3E 65 3F 00 00 00 A6 00 19 00
Possible partition(DOS FAT), size(917mb), offset(800mb)
  type: 006(0x06)(Primary 'big' DOS (> 32MB))
  size: 917mb #s(1879604) s(1638630-3518233)
  chs: (102/0/1)-(218/254/62)d (102/0/1)-(218/254/62)r
  hex: 00 00 01 66 06 FE 3E DA E6 00 19 00 34 AE 1C 00
```

13.4 I Volumi - DOS Partition

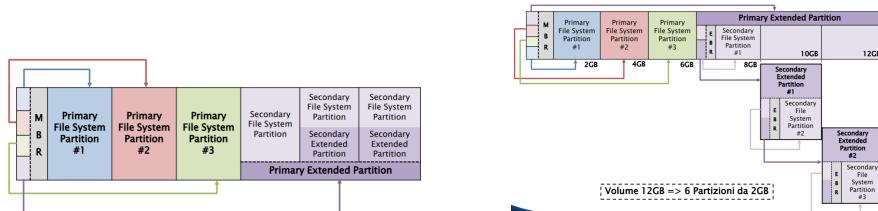
Uno dei primi sistemi di partizionamento è stato il **DOS Partition**, o anche detto **MBR**, ovvero **Master Boot Record**, che è il nome del primo settore dove vengono memorizzate le strutture dati di questo tipo di sistemi di partizionamento. Il **DOS Partition** viene tutt'ora utilizzato, per lo più su dischi al di sotto di 2 TB, e nel primo settore, ovvero nell'**MBR** possiamo trovare:

- **Boot Code**: esso contiene istruzioni per il BIOS sulla tabella di partizionamento e su come localizzare ed avviare il Sistema Operativo.
- **Partition Table**: tabella con al massimo 4 entry, quindi al massimo 4 partizioni. La tabella è composta da:
 - Starting CHS Address.
 - Ending CHS Address.
 - Starting LBA Address.
 - Number of sectors in partition.
 - Type of partition.
 - Flags.
- **Signature**: 0x55AA, è ciò che troviamo alla fine della tabella di partizione.



Questa tipologia di struttura elementare diventa più complessa quando si cerca di superare i limiti tecnologici del sistema di partizionamento, superando le 4 partizioni.

- **Primary File System Partition:** è quella partizione che è presente e viene descritta nell'**MBR**, la prima entry della tabella di partizione è collegata alla partizione primaria che contiene un file system.
- **Primary Extended Partition:** resta comunque una partizione primaria descritta nella prima tabella di partizione, ma al suo interno viene suddivisa in ulteriori partizioni. Infatti al suo interno troveremo:
 - **Una propria tabella di partizione:** essa farà riferimento alle diverse partizioni all'interno della extended partition.
 - **Secondary File System Partition:** partizione puntata dalla prima entry della tabella di partizione e contiene anch'essa un file system. (*partizione logica*)
 - **Secondary Extended Partition:** partizione puntata dalla seconda entry, anche essa contiene altre partizioni e la propria tabella di partizione. Questa a sua volta conterrà le stesse cose della *Primary Extended Partition*.



In questo modo viene abbattuto il vincolo delle 4 partizioni massime.

13.4.1 Boot Code

Il **Boot Code** è situato nei primi **446 byte** del primo settore, ovvero all'interno dell'**MBR**. Lo scopo del boot code è di analizzare le varie partizioni e verificare quale sia avviabile, successivamente viene contrassegnata con un flag.

Molti virus erano progettati per stanzarsi nei **446 byte** del boot code, in modo da essere eseguiti ogni volta che si riavvia il disco.

Il settore **MBR** viene allocato all'inizi del disk volume e ad ogni extended partition. In queste ultime la parte riservata al boot code resta inutilizzata e vengono utilizzate solo le prime due entry dell'**EBR**.

13.4.2 Partition Table

Byte Range	Description	Essential
0-445	Boot Code	No
446-461	Partition Table Entry #1	Yes
462-477	Partition Table Entry #2	Yes
478-493	Partition Table Entry #3	Yes
494-509	Partition Table Entry #4	Yes
510-511	Signature value (0xAA55)	No

Byte Range	Description	Essential
0-0	Bootable Flag	No
1-3	Starting CHS Address	Yes
4-4	Partition Type	No
5-7	Ending CHS Address	Yes
8-11	Starting LBA Address	Yes
12-15	Size in Sectors	Yes

Logica di una partition table di tipo **DOS Partition**.

Type	Description	Type	Description	Type	Description
0x00	Empty	0x11	Hidden FAT12, CHS	0xa0/1	Hibernation
0x01	FAT12, CHS	0x14	Hidden FAT16, 16-32 MB, CHS	0xa5	FreeBSD
0x04	FAT16, 16-32 MB, CHS	0x16	Hidden FAT16, 32 MB-2GB, CHS	0xa6	OpenBSD
0x05	Microsoft Extended, CHS	0x1b	Hidden FAT32, CHS	0xa8	Mac OSX
0x06	FAT16, 32 MB-2GB, CHS	0x1c	Hidden FAT32, LBA	0xa9	NetBSD
0x07	NTFS	0x1e	Hidden FAT16, 32 MB-2GB, LBA	0xab	Mac OSX Boot
0x0b	FAT32, CHS	0x42	Microsoft MBR, Dynamic Disk	0xb7	BSDI
0x0c	FAT32, LBA	0x82	Solaris x86 Linux Swap	0xb8	BSDI swap
0x0e	FAT16, 32 MB-2GB, LBA	0x83	Linux	0xee	EFI GPT Disk
0x0f	Microsoft Extended, LBA	0x84	Hibernation	0xef	EFI System Partition
		0x85	Linux Extended	0xfb	Vmware File System
		0x86/7	NTFS Volume Set	0xfc	Vmware swap

Tipologie di partizione riportate in esadecimale.

13.5 DOS Partition - Analisi

Vedremo come eseguire l'analisi di una **DOS Partition** partendo da un file immagine ed usufruendo del comando **dd** e di un editor di testo esadecimale chiamato **xxd**.

```
root@caine:/# dd if=disk3.dd bs=512 skip=0 count=1 | xxd
0000000: eb48 9010 8ed0 bc00 b0b8 0000 8ed8 8ec0 .H.....[. . .]
0000034: 0048 6172 6420 4469 736b 0052 6561 6400 .Hard Disk.Read.
0000040: 2045 7272 6f72 00bb 0100 b40e cd10 ac3c Error.....<
00000416: 0075 f4c3 0000 0000 0000 0000 0000 0000 .u......
00000432: 0000 0000 0000 0000 0000 0000 0000 0001 .....A`.....
00000448: 0100 07fe 3f7f 3f00 0000 4160 1f00 8000 ....?..?..A`.....
00000464: 0180 83fe 3f8c 8060 1f00 cd2f 0300 0000 .....?..?..../...
00000480: 018d 83fe 3fcc 4d90 2200 40b0 0f00 0000 .....?..M."@.....
00000496: 01cd 05fe ffff 8d40 3200 79eb 9604 55aa .....@2.y...U.
```

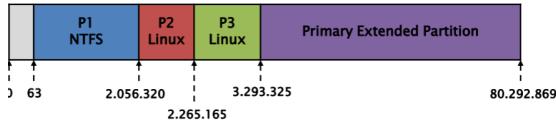
Dal byte numero 446, fino al byte 509 troviamo la tabella di partizione. Prima di questa invece si trova il boot code.

P1: 0001 0100 07fe 3f7f 3f00 0000 4160 1f00
P2: 8000 0180 83fe 3f8c 8060 1f00 cd2f 0300
P3: 0000 018d 83fe 3fcc 4d90 2200 40b0 0f00
P4: 0000 01cd 05fe ffff 8d40 3200 79eb 9604

Part. 0-15	BootFlag 0-0	Start CHS 1-3	Type 4-4	End CHS 5-7	LBA 8-11	Size 12-15
P1	00	01 01 00	07	fe 3f 7f	3f 00 00 00	41 60 1f 00
	00	00 01 01	07	7f 3f fe	00 00 00 3f	00 1f 60 41
	00	-	NTFS	-	63	2.056.257
P2	80	00 01 80	83	fe 3f 8c	80 60 1f 00	cd 2f 03 00
	80	80 01 00	83	8c 3f fe	00 1f 60 80	00 03 2f cd
	80	-	Linux	-	2.056.320	208.845

Quanto recuperato nell'**MBR**, viene poi analizzato. Ogni 16 byte sono una entry/partizione, quindi troviamo, per le prime due partizioni, quanto mostrato sopra.

Part. 0-15	BootFlag 0-0	Start CHS 1-3	Type 4-4	End CHS 5-7	LBA 8-11	Size 12-15
P3	00	00 01 8d	83	fe 3f cc	4d 90 22 00	40 b0 00 00
	00	8d 01 00	83	cc 3f fe	00 22 90 4d	00 0f b0 40
	00	-	Linux	-	2.265.165	1.028.160
P4	00	00 01 cd	05	fe ff ff	8d 40 32 00	79 eb 96 04
	00	cd 01 00	05	ff ff fe	00 32 40 8d	04 96 eb 79
	00	-	DOS Ext	-	3.293.325	79.999.545



Per le restanti due partizioni, notiamo che la quarta di esse ha un formato extended quindi ci aspettiamo ulteriori partizioni al suo

interno. Di regola dovremmo effettuare l'analisi anche di quella partizione.

```
root@caine:/# dd if=disk3.dd bs=512 skip=3293325 count=1 | xxd
[. . .]
0000432: 0000 0000 0000 0000 0000 0000 0000 0001 .....
0000448: 01cd 83fe 7fc0 0000 0082 3e00 0000 .....?....>...
0000464: 41cc 05fe bf0b 3f82 3e00 40b0 0f00 0000 A.....?>.@....
0000480: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa .....U.

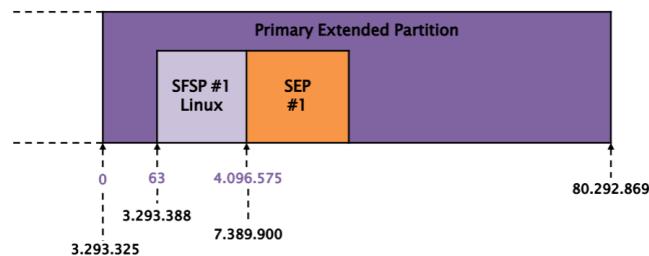
SFSP 1
SEP1
```

Procediamo quindi con l'estrare l'**EBR** della primary extended partition che sappiamo esattamente dove inizia. Troviamo altre due partizioni infatti.

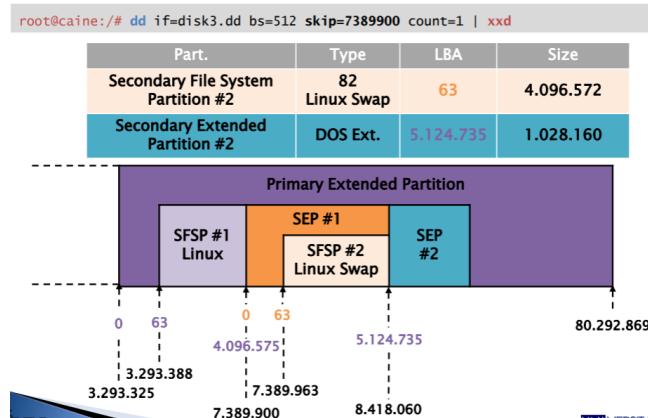
Part. 0-15	BootFlag 0-0	Start CHS 1-3	Type 4-4	End CHS 5-7	LBA 8-11	Size 12-15
SFSP #1	00	01 01 cd	83	fe 7f cb	3f 00 00 00	00 82 3e 00
	00	cd 01 01	83	cb 7f fe	00 00 00 3f	00 3e 82 00
	00	-	Linux	-	63	4.096.572
SEP #1	00	00 41 cc	05	fe bf 0b	3f 82 3e 00	40 b0 0f 00
	00	cc 41 00	05	0b bf fe	00 3e 82 3f	00 0f b0 40
	00	-	DOS E	-	4.096.575	1.028.160

La seconday file system partition è un sistema operativo Linux e la seconda partizione è un ulteriore extended partition (**Secondary Extended Partition**) da analizzare.

Part.	Type	LBA	Size
Secondary File System Partition #1	Linux	63	4.096.572
Secondary Extended Partition #1	DOS Ext.	4.096.575	1.028.160



Non sappiamo cosa può esserci dopo **SEP#1** visto che abbiamo ulteriore spazio.



Estraiamo nuovamente l'**EBR** dalla partizione extended e troviamo una partizione di swap per linux ed un'ulteriore extended partition.

```
root@caine:/# fdisk -lu disk3.dd
Disk disk3.dd: 255 heads, 63 sectors, 0 cylinders
Units = sectors of 1 * 512 bytes
Device      Boot  Start    End   Blocks Id System
disk3.dd1            63  2056319  1028128+  7 HPFS/NTFS
disk3.dd2  *   2056320  2265164   104422+ 83 Linux
disk3.dd3    2265165  3293324    514080 83 Linux
disk3.dd4    3293325  80292869  38499772+  5 Extended
disk3.dd5    3293388  7389899    2048256 83 Linux
disk3.dd6    7389963  8418059   514048+ 82 Linux swap
disk3.dd7    8418123  9446219   514048+ 83 Linux
disk3.dd8    9446283 17639369  4096543+  7 HPFS/NTFS
disk3.dd9   17639433 48371714  15366141 83 Linux
```

Un modo più veloce per identificare le partizioni disponibili di un disk image è tramite il comando **fdisk**.

```

root@caine:/# mmls -t dos disk3.dd
Disk disk3.dd: 255 heads, 63 sectors, 0 cylinders
Units are in 512-byte sectors
      Slot Start      End      Length     Description
00: ----- 0000000000 0000000000 0000000001 Table #0
01: ----- 0000000001 0000000062 0000000062 Unallocated
02: 00:00 0000000063 0002056319 0002056257 NTFS (0x07)
03: 00:01 0002056320 0002265164 0000208845 Linux (0x83)
04: 00:02 0002265165 0003293324 0001028160 Linux (0x83)
05: 00:03 0003293325 0080292869 0076999545 DOS Extended (0x05)
06: ----- 0003293325 0003293325 0000000001 Table #1
07: ----- 0003293326 0003293387 0000000062 Unallocated
08: 01:00 0003293388 0007389899 0004096512 Linux (0x83)
09: 01:01 0007389900 0008418059 0001028160 DOS Extended (0x05)

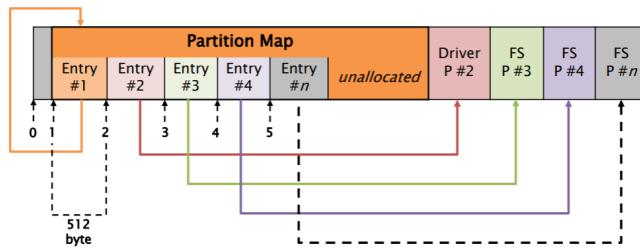
10: ----- 0007389900 0007389900 0000000001 Table #2
11: ----- 0007389901 0007389962 0000000062 Unallocated
12: 02:00 0007389963 0008418059 0001028097 Linux Swap (0x82)
13: 02:01 0008418060 0009446219 0001028160 DOS Extended (0x05)
14: ----- 0008418060 0008418060 0000000001 Table #3
15: ----- 0008418061 0008418122 0000000062 Unallocated
16: 03:00 0008418123 0009446219 0001028097 Linux (0x83)
17: 03:01 0009446220 0017639369 0008193150 DOS Extended (0x05)
18: ----- 0009446220 0009446220 0000000001 Table #4
19: ----- 0009446221 0009446282 0000000062 Unallocated
20: 04:00 0009446283 0017639369 0008193087 NTFS (0x07)
21: 04:01 0017639370 0048371714 0030732345 DOS Extended (0x05)
22: ----- 0017639370 0017639370 0000000001 Table #5
23: ----- 0017639371 0017639432 0000000062 Unallocated
24: 05:00 0017639433 0048371714 0030732282 Linux (0x83)

```

Un'analisi più dettagliata e veloce è possibile tramite **mmls** specificando l'opzione *dos* nel comando andiamo ad informare il tool della presenza di partizioni estese.

13.6 I Volumi - Apple Partition Map

Il sistema di partizionamento dei primi sistemi apple era **APM** pren-
deva la sua natura un po **MBR** ed un po da un altro sistema di
partizionamento che era **BSD**. Rispetto ad **MBR** e **BSD**, **APM** non
impone un numero limite di partizioni che si possono creare, l'unico
limite è la dimensione del volume. Il sistema di partizionamento
è definito dalla **Partition Map** che è situata nel secondo settore del
disco. Non è presente un boot code, in quanto se ne occupava il
firmware del processore, sistema più sicuro, non alterabile come
boot code. Ogni entry della Partition Map descrive una partizione
e la prima entry descrive la Partition Map stessa.



Ogni entry della Partition Map memorizza necessariamente:

- **Settore iniziale della partizione.**
- **Dimensione.**
- **Tipo.**
- **Nome del volume.**

Byte Range	Description	Essential	Byte Range	Description	Essential
0-1	Signature value (0x504D)	No	92-95	Starting sector of boot code	No
2-3	Reserved	No	96-99	Size of boot code in sectors	No
4-7	Total Number of partitions	Yes	100-103	Address of boot loader code	No
8-11	Starting sector of partition	Yes	104-107	Reserved	No
12-15	Size of partition in sectors	Yes	108-111	Boot code entry point	No
16-47	Name of partition in ASCII	No	112-115	Reserved	No
48-79	Type of partition in ASCII	No	116-119	Boot code checksum	No
80-83	Starting sector of data area in partition	No	120-135	Processor type	No
84-87	Size of data area in sectors	No	136-511	Reserved	No
88-91	Status of partition	No			

Qui ogni entry ha ben **512 byte**, quindi ha molto più spazio per
memorizzare informazioni sulla partizione rispetto ad **MBR**, inoltre
per scorrere ogni partizione basta spostarsi di un settore alla volta.

13.7 Apple Partition Map - Analisi

Estrazione ed analisi tramite il comando `dd` leggendo il disk image andiamo ad analizzare la prima entry.

```
root@caine:/# dd if=mac-disk.dd bs=512 skip=1 count=1 | xxd
0000000: 504d 0000 000a 0000 0001 0000 003f PM....?
0000016: 4170 706c 6500 0000 0000 0000 0000 Apple....
0000032: 0000 0000 0000 0000 0000 0000 0000 .....
0000048: 4170 706c 655f 7061 7274 6974 696f 6e5f Apple_partition_
0000064: 6d61 7000 0000 0000 0000 0000 0000 map....
0000080: 0000 0000 0000 003f 0000 0000 0000 0000 .....?
0000096: 0000 0000 0000 0000 0000 0000 0000 0000 .....
[. . .]
```

Byte Range	Description	Value
0-1	Signature value	504d
4-7	Total Number of partitions	0000000a (10)
8-11	Starting sector of partition	00000001 (1)
12-15	Size of partition in sectors	0000003f
16-47	Name of partition in ASCII	Apple
48-79	Type of partition in ASCII	Apple_partition_map

I primi due byte sono per la **signature** ed indicano una partition map.

```
root@caine:/# mm1s -t mac mac-disk.dd

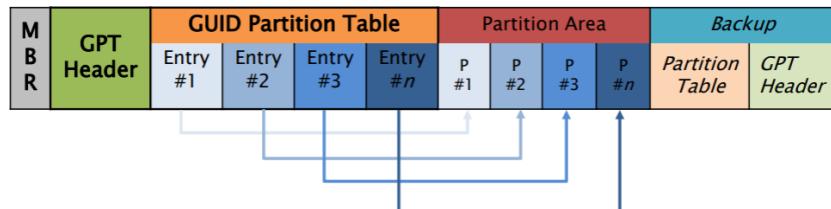
MAC Partition Map
Units are in 512-byte sectors
      Slot Start      End      Length      Description
00: ----- 0000000000 0000000000 0000000001 Unallocated
01: 00 0000000001 0000000063 0000000063 Apple_partition_map
02: ----- 0000000001 0000000010 0000000010 Table
03: ----- 0000000011 0000000063 0000000053 Unallocated
04: 01 0000000064 0000000117 0000000054 Apple_Driver43
05: 02 0000000118 0000000191 0000000074 Apple_Driver43
06: 03 0000000192 0000000245 0000000054 Apple_Driver_ATA
07: 04 0000000246 0000000319 0000000074 Apple_Driver_ATA
08: 05 0000000320 0000000519 0000000200 Apple_FWDriver
09: 06 0000000520 0000001031 0000000512 Apple_Driver_IOKit
10: 07 0000001032 0000001543 0000000512 Apple_Patches
11: 08 0000001544 0039070059 0039068516 Apple_HFS
12: 09 0039070060 0039070079 0000000020 Apple_Free
```

Con il comando `mm1s` velocizziamo il processo di listing di tutte le partizioni. Possiamo notare come siano incluse partizioni con firmware e driver dedicate all'uso della macchina.

13.8 I Volumi - Guid Partition Table

È il sistema di partizionamento più usato da quando è stato introdotto **EFI**. Permette un massimo di **128 partizioni**, è quindi limitato e sono poi usati 64 bit per definire l'**LBA** e questo permette di indirizzare anche volumi più grandi di 2 TB, cosa che con **MBR** non era possibile. **GPT** ha un layout composto da 5 sezioni:

- **Protective MBR**: posizionato nel primo settore del disco e contiene il sistema di partizione **DOS** con una sola entry e il resto della partizione è il restante disco. Questa possibilità è stata introdotta per creare retro-compatibilità con i sistemi che non supportavano **EFI**.
- **GPT Header**: definisce la posizione e la dimensione della tabella di partizione, al termine dell'header si trova un checksum che serve ad evitare che qualcuno modifichi queste informazioni.
- **Partition Table**: ogni entry fa riferimento ad una partizione.
- **Partition Area**: locazione riservata alle partizioni.
- **Backup Area**: qui è dove viene fatta una copia del **GPT Header** e della **Partition Table** all'interno di un altro settore, questo perché, qualora si fosse voluto eliminare una di queste due informazioni sarebbe stato rimosso il primo settore e non altre copie.



```

root@caine:/# mmls -t dos gpt-disk.dd

DOS Partition Table
Units are in 512-byte sectors
  Slot Start      End      Length      Description
 00: ----- 0000000000 0000000000 0000000001 Primary Table (#0)
 01: 00:00 0000000001 0120103199 0120103199 GPT Safety Partition (0xEE)

```

Utilizzando **mmls** con il parametro ***dos*** come se stessimo analizzando una **DOS Partition**, notiamo come venga riscontrato anche l'**MBR** e la sua unica entry che definisce l'intera partizione con **GPT**.

Byte Range	Description	Essential
0-7	Signature value ("EFI PART")	No
8-11	Version	Yes
12-15	Size of GPT header in bytes	Yes
16-19	CRC32 checksum of GPT header	No
20-23	Reserved	No
24-31	LBA of current GPT header structure	No
32-39	LBA of the other GPT header structure	No
40-47	LBA of start of partition area	Yes
48-55	LBA of end of partition area	No
56-71	Disk GUID	No
72-79	LBA of the start of the partition table	Yes
80-83	Number of entries in partition table	Yes
84-87	Size of each entry in partition table	Yes
88-91	CRC32 checksum of partition table	No
92-End Sector	Reserved	No

Queste sono le informazioni che memorizza il **GPT Header**, ovvero le info che memorizza questo sistema di partizionamento. Analizzando l'header iniziamo quindi ad intuire una sorta di layout di partizionamento, ad esempio possiamo subito riscontrare dove si trovano le altre aree, quando è grande la tabella di partizione, ed altre info simili.

13.9 Guid Partition Table - Analisi

Analizzando il secondo settore del disco, ovvero il **GPT Header**, di un disk image, possiam leggere le informazioni.

root@caine:/# dd if=gpt-disk.dd bs=512 skip=1 count=1 xxd		
0000000:	4546 4920 5041 5254 0000 0100 5c00 0000	EFI PART....\...
0000016:	8061 a3b0 0000 0000 0100 0000 0000 0000a.....
0000032:	1fal 2807 0000 0000 2200 0000 0000 0000	..(.....".....
0000048:	fea0 2807 0000 0000 7e5e 4da1 1102 5049	..(.....~^M...PI
0000064:	ab2a 79a6 3ea6 3859 0200 0000 0000 0000	.*y.>8Y.....
0000080:	8000 0000 8000 0000 69a5 7180 0000 0000i.q.....
0000096:	0000 0000 0000 0000 0000 0000 0000 0000
[. . .]		
Byte Range	Description	Value
0-7	Signature value	EFI PART
12-15	Size of GPT header in bytes	5c00 (96)
32-39	LBA of the other GPT header structure	0728a1af (120.103.199)
40-47	LBA of start of partition area	0022(34)
48-55	LBA of end of partition area	0728a0fe (120.103.166)
72-79	LBA of the start of the partition table	0002 (2)
80-83	Number of entries in partition table	0080 (128)
84-87	Size of each entry in partition table	0080 (128)

Informazioni di ogni entry che è composta da 128 byte.

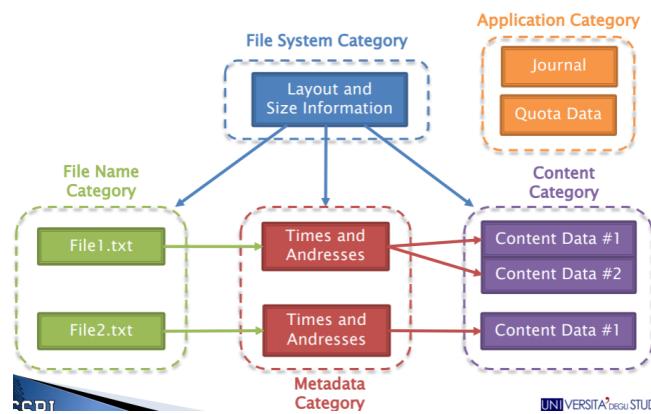
Byte Range	Description	Essential
0-15	Partition type GUID	No
16-31	Unique partition GUID	No
32-39	Starting LBA of partition	Yes
40-47	Ending LBA of partition	Yes
48-55	Partition attributes	No
56-127	Partition name in Unicode	No

14 Lezione 15

14.1 File System

14.1.1 Overview

Il file system nasce per rispondere ad un requisito generale di memorizzazione a lungo termine in modo gerarchico affinché il reperimento delle informazioni memorizzate sia il più semplice possibile. Il file system fa uso di file e directory ed organizza al meglio i dati in modo che il sistema operativo sappia come recuperarli.



Modello per il confronto dei diversi file system e comprendere come essi memorizzano determinati dati e quali strutture dati impiegano. Questo modello fa uso di cinque categorie, tutti i dati che troviamo all'interno di un file system ricadono in una di queste categorie.

- **File System Category**: contiene informazioni generali sul file system. Queste info potrebbero indicarci, ad esempio, dove trovare una determinata struttura dati. I dati che sono presenti o ricadono in questa categoria sono importanti per darci una "mappa" di uno specifico file system.
- **Content Category**: contiene i dati che sono contenuti all'interno di un file, ovvero la ragione per cui si fa uso del file system. La grande parte dei dati in un file system ricade, per lo più, in questa categoria. Tipicamente i dati in un file system sono organizzati in una collezione di settori che per ora chiameremo **Data Unit**.
- **Metadata Category**: contiene i dati che descrivono il file, ovvero tutte quelle informazioni come per esempio "*dove è memorizzato il contenuto del file*", oppure "*quanto è grande*" oppure informazioni temporali come la creazione, ecc.

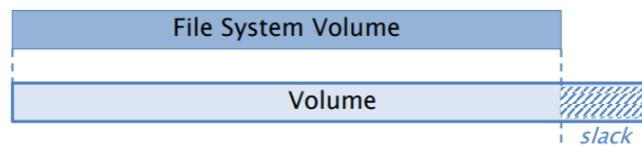
- **File Name Category**: contiene il dato che assegna un nome ad un file. Non sono altro che liste di file name con l'indirizzo al file della categoria metadati.
- **Application Category**: contiene i dati che provvedono a specifiche caratteristiche, questi dati non sono necessari al funzionamento del file system, ma averli rende alcune operazioni più efficienti.

Esistono dati all'interno del file system, che se persi o alterati, compromettono il funzionamento del file system stesso. Ci sono poi dati accessori che non influiscono sul funzionamento di quest'ultimo.

- **Dati Essenziali**: ad esempio informazioni che permettono di recuperare il contenuto del file, oppure il nome del file. Queste informazioni vengono chiamate **Trusted Data**
- **Dati NON essenziali**: ad esempio dati temporali o i permessi utente. Queste informazioni vengono chiamate **Untrusted Data**.

14.2 File System Category

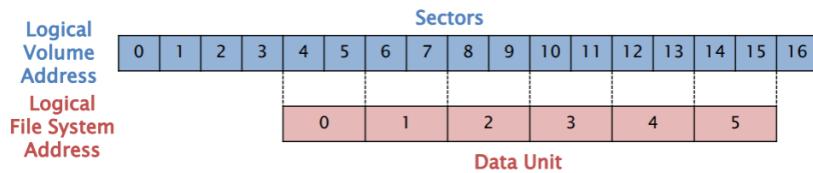
Contiene dati generali che identificano il modo in cui il file system che stiamo analizzando è **unico** e dove sono posizionati gli altri dati importanti. In molti casi questi dati si trovano nel primo settore in una struttura dati piuttosto standard. L'analisi di questa categoria è necessaria per svolgere appunto l'analisi di tutti i file system perché è in questa fase che identifichiamo le altre strutture dati delle varie categorie. Se alcuni di questi dati si corrompono o si perdono, l'analisi diventa più complessa, l'unica soluzione sarebbe avere un backup. Tutti i dati che non riguardano strettamente il layout dei dati sono considerati non essenziali, quindi potrebbero non essere accurati. Un'analisi banale da poter eseguire su questa categoria di dati ci permette di reperire informazioni sul layout dei dati, ma anche info su quale computer è stato creato il file system oppure un **controllo di consistenza**.



Il controllo di consistenza viene effettuato qualora si trovasse dello spazio non allocato sul disco ed al di fuori del file system, questo solitamente viene chiamato **Volume Slack**.

14.3 Content Category

Qui sono presenti le locazioni di memoria usate per salvare file e directory. I dati in questa categoria sono tipicamente organizzati in gruppi di uguali dimensioni detti, per ora, **Data Unit** le quali hanno uno stato che le identifica come "*Allocato*" oppure "*Non Allocato*" e alcune strutture dati del file system ne tengono traccia. Quindi ogni volta che un file aumenta di dimensioni il sistema operativo dovrà metterlo in una data unit **non allocata** più grande rispetto alla precedente. Se un file viene invece cancellato, la data unit associata ad esso viene **disallocata**. L'analisi che viene per lo più svolta in questa categoria è per recuperare i file cancellati. Essendo molti i dati in questa categoria non è possibile eseguire l'analisi manualmente. Anche per il file system c'è un indirizzamento, esso si appoggia al **Logical Volume Address** e viene chiamato **Logical File System Address** che ha lo scopo di raggruppare tutti i settori consecutivi in un'unica **Data Unit**.



Osserviamo come il settore numero 16 rappresenti un **Volume Slack**.

14.3.1 Content Category - Analisi

Per questa categoria di file le informazioni su cui è possibile eseguire un'analisi possono essere:

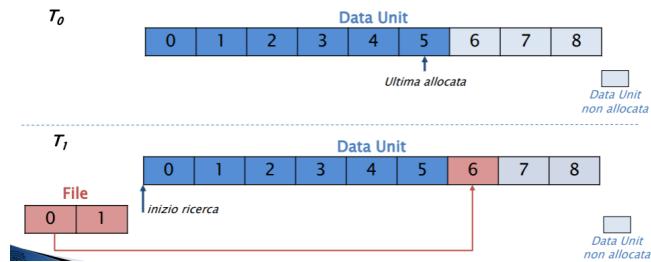
1. **Data Unit View**: ricerca di settori noti del file system.
2. **Logical File System Searching**: ricerca della presenza di uno specifico dato all'interno delle data unit. Questa è un'analisi lenta, una data unit alla volta.
3. **Data Unit Allocation Status**: ricerca concentrata solo sulle data unit che hanno un determinato stato di allocazione, es: le data unit non allocate.
4. **Consistency Check**: ricerca di data unit non referenziate in **Metadata Category**, dette **Orphan Data Unit**.

14.4 Strategie di Allocazione

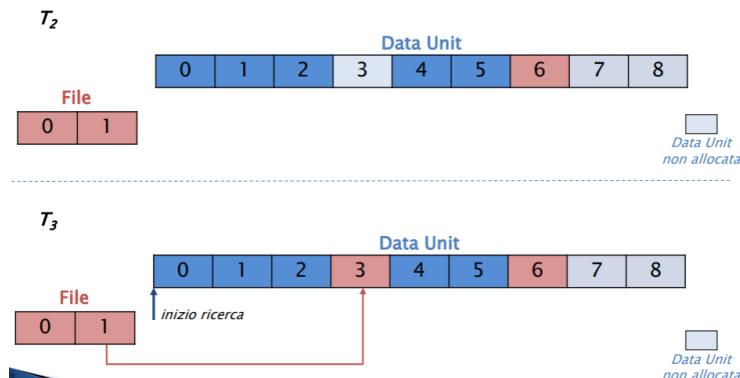
Per quanto riguarda la memorizzazione di file, ogni sistema operativo, fa uso di diverse strategie di allocazione delle data unit. Tipicamente il S.O. quando deve memorizzare un file cerca di allocare le data unit in maniera consecutiva, però questo non è sempre possibile, se ad esempio il file è frammentato su più data unit.

14.4.1 Strategia del Primo Disponibile

Questa strategia consiste nel ricercare, dall'inizio del file system, la prima data unit libera per allocare un file, ed ogni volta si crei un nuovo file si cerca una nuova data unit libera. Questo tipo di strategia crea molti file frammentati, in questo modo:



Ipotizziamo di voler allocare un file che necessita di due data unit. A tempo T_1 viene allocata la prima parte del file nella prima data unit libera e non allocata, in questo caso la numero 6.

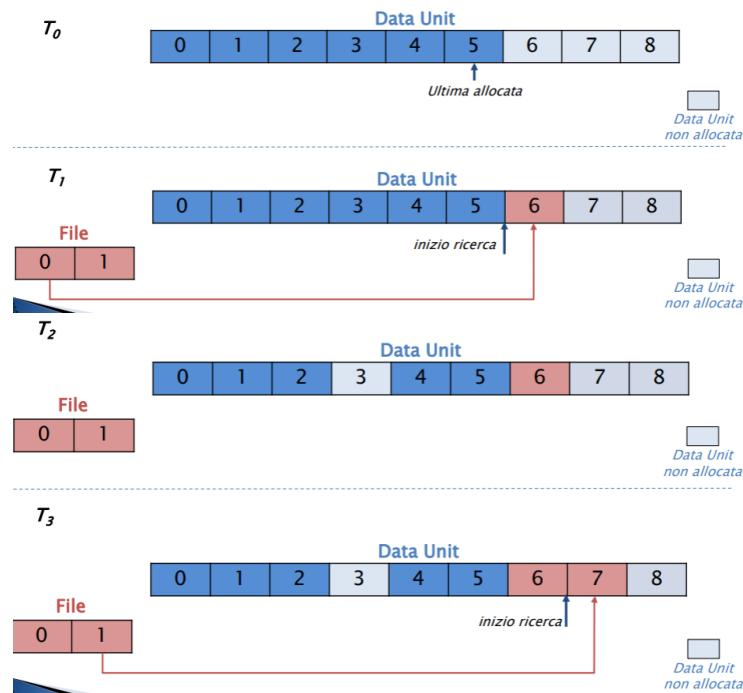


Mentre eseguo questa operazione, al tempo T_2 il file system libera la data unit 3, quindi la prossima data unit non allocata pronta all'uso sarà appunto questa, quindi non appena la restante parte del file dovrà essere allocata verrà scelta la data unit 3 e non la 7 consecutivamente. Con questa strategia, il recupero dei dati cancellati è

più efficiente eseguirlo partendo dalla fine, e non dall'inizio del file system, trovano per lo più data unit disallocate e non sovrascritte.

14.4.2 Strategia del Prossimo Disponibile

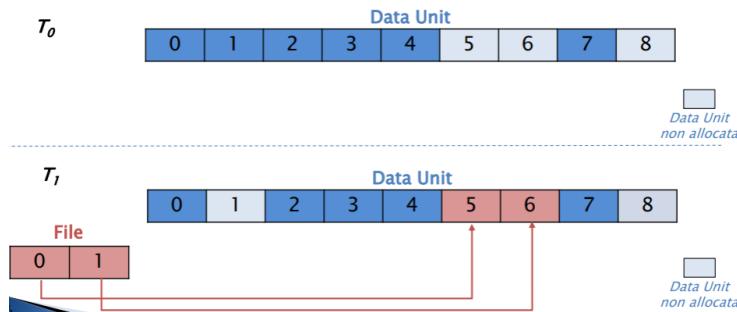
Molto simile alla precedente strategia, solo che invece di cercare ogni volta partendo dall'inizio del file system la nuova data unit libera, si parte dall'ultima data unit utilizzata/allocata. In questo modo:



In questo caso un file che necessita di due data unit per essere allocato parte dalla numero 6 e se nel mentre se ne dealloca una prima non viene considerata, perché con questa strategia andrà a cercare la prima libera partendo dall'ultima allocata, ovvero in questo caso la numero 6. La data unit successiva alla 6 non allocata è proprio la numero 7, quindi avremo il file allocato in due data unit separate ma consecutive. In questo caso risulta più facile ritrovare file cancellati che si trovano all'inizio del file system.

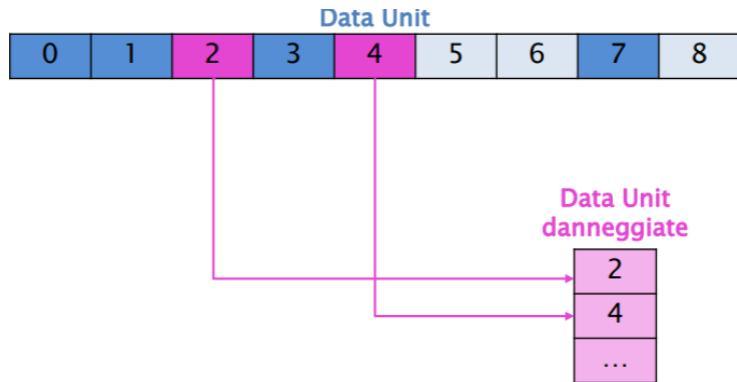
14.4.3 Strategia del più Adatto

Questa strategia ricerca le data unit consecutive che sono in grado di contenere tutte le data unit necessarie ad allocare il file. Questa strategia risulta ottimale nella parte iniziale del file system, ovvero quando devo allocare un nuovo file e si conosce la dimensione totale. Mentre non risulta funziona questa strategia quando il file cresce di dimensioni e obbliga il sistema operativo a cercare un'altra data unit per la porzione di file in più.



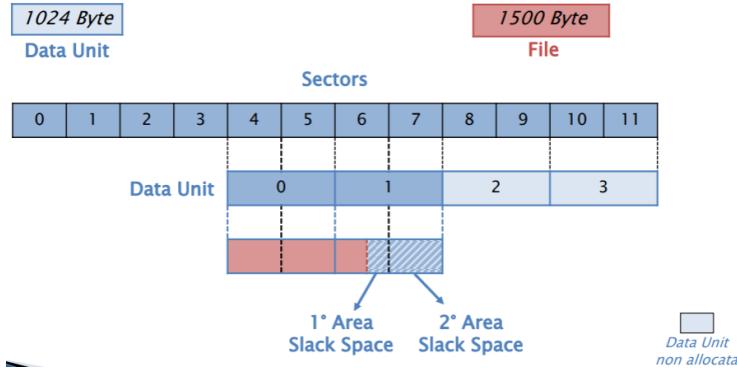
14.4.4 Data Unit Danneggiate

Questo tipo di informazione, soprattutto nei dischi meno moderni, vengono raccolte all'interno del file system per una questione di velocità. Il sistema operativo è quindi a conoscenza delle data unit marcate come **danneggiate**.

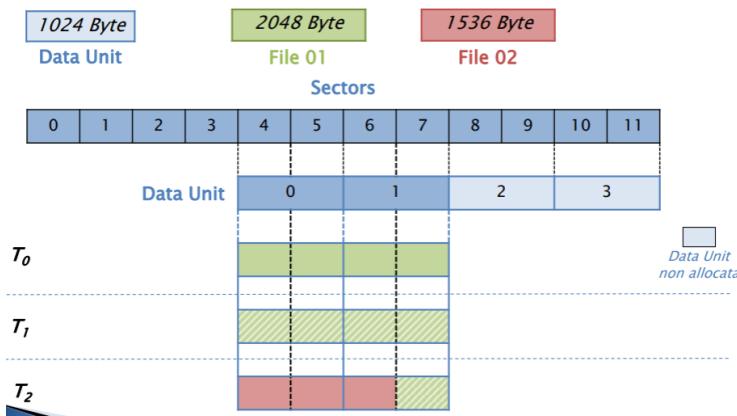


14.5 File System - Slack Space

Quando un file viene creato alloca tutta una data unit, anche se gliene serve meno. Lo **Slack Space** si viene a creare quando resta dello spazio non usato di una data unit allocata.



Lo **slack space** non è sempre "pulito" dal sistema operativo, bensì avvolte possono restare, all'interno di quello spazio, informazioni su file precedentemente allocati in quella zona. Nell'immagine notiamo come si vengano a creare due porzioni di slack space, la prima di queste viene anche detta **Padding** e viene riempita con tutti zero.



14.6 Metadata Category

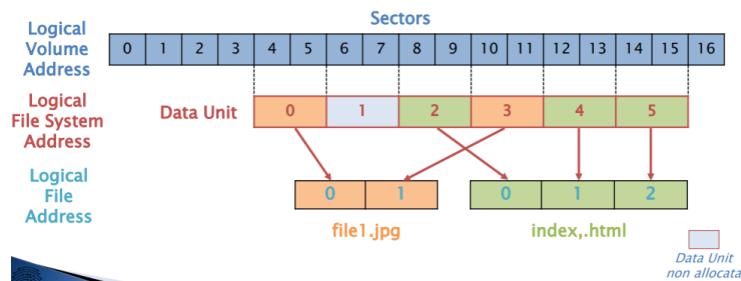
In questa categoria sono presenti i dati descrittivi, ovvero quei dati che vanno a descrivere i dati presenti nella content category. Ad esempio la data di accesso o il numero di data unit allocate per un determinato file. I metadati solitamente vengono memorizzati in tabelle che possono essere di dimensione fissa o dinamica ed ogni entry ha un proprio indirizzo. Quando un file viene cancellato i metadati correlati ad esso passeranno da uno stato di allocato a non allocato.

14.6.1 Metadata Category - Analisi

L'analisi dei dati presenti in questa categoria solitamente è effettuata per avere maggiori dettagli su un determinato file oppure per eseguire una ricerca di alcuni file con determinate caratteristiche descritte in questa categoria.

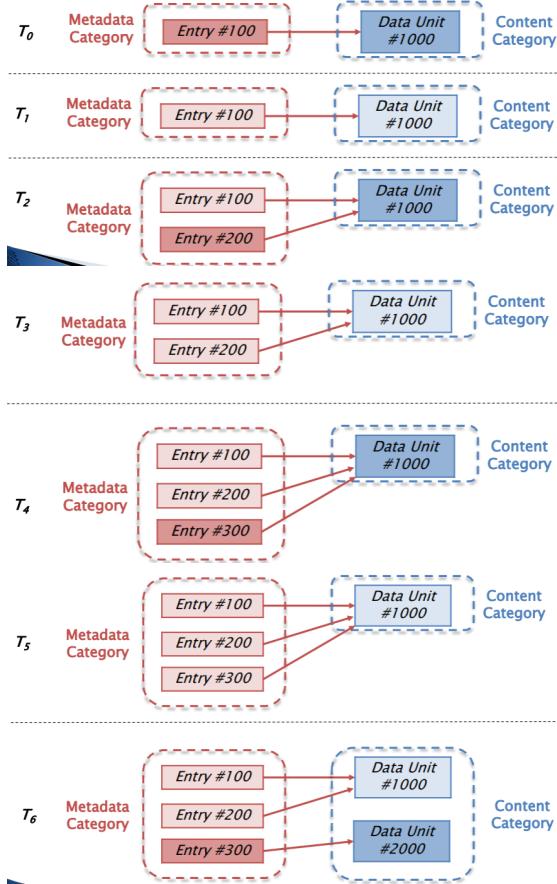
14.6.2 Metadata Category - Logical File Address

Come abbiamo visto, le data unit sono associate ad un logical file system address, quest'ultimo descrive i settori con i quali sono composti. Ad ogni data unit allocata è inoltre associato un **Logical File Address** ed esso è calcolato in base all'inizio del file allocato nella data unit.



14.6.3 Metadata Category - File Recovery

Un recupero dei dati attraverso il metadato è possibile solo se quest'ultimo esiste ancora, ovvero esiste ancora la entry relativa a quel file e non è stata riutilizzata. Per il recupero dei dati basta leggere le allocazioni indicate, questa procedura non è però affidabile in quanto non si ha la certezza che le data unit impiegate corrispondano al file indicato nella metadata.



- A tempo T_1 : il file viene eliminato, la data unit deallocata ed anche le entry.
- A tempo T_2 : viene creato un nuovo file utilizzando la entry 200 e con la stessa data unit di prima.
- A tempo T_3 : il file viene nuovamente eliminato. Analizzando il file system troviamo due enty che puntano alla stessa data unit.
- A tempo T_4 : viene creato un nuovo file con una nuova entry ma stessa data unit.
- A tempo T_5 : il file viene di nuovo eliminato e deallocated.
- A tempo T_6 : nuovo file riutilizzando la entry 300 ed usando una nuova data unit.

Bisogna sempre prendere con le pinze le informazioni recuperate dal file system.

Inoltre ricorda che alcuni file system permettono di memorizzare i dati in un formato compresso così da occupare meno data unit. Ci sono tre livelli di compressione:

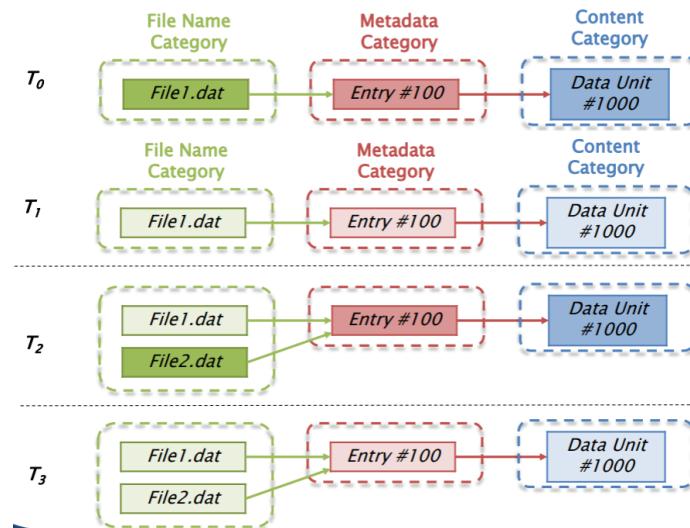
- Compressione dei soli dati all'interno del file (JPG, MP3, ...).
- Compressione di tutto il file, di conseguenza creazione di un nuovo file (ZIP, RAR, ...).
- Compressione eseguita lato file system e di conseguenza invisible al lato applicativo ed utente.

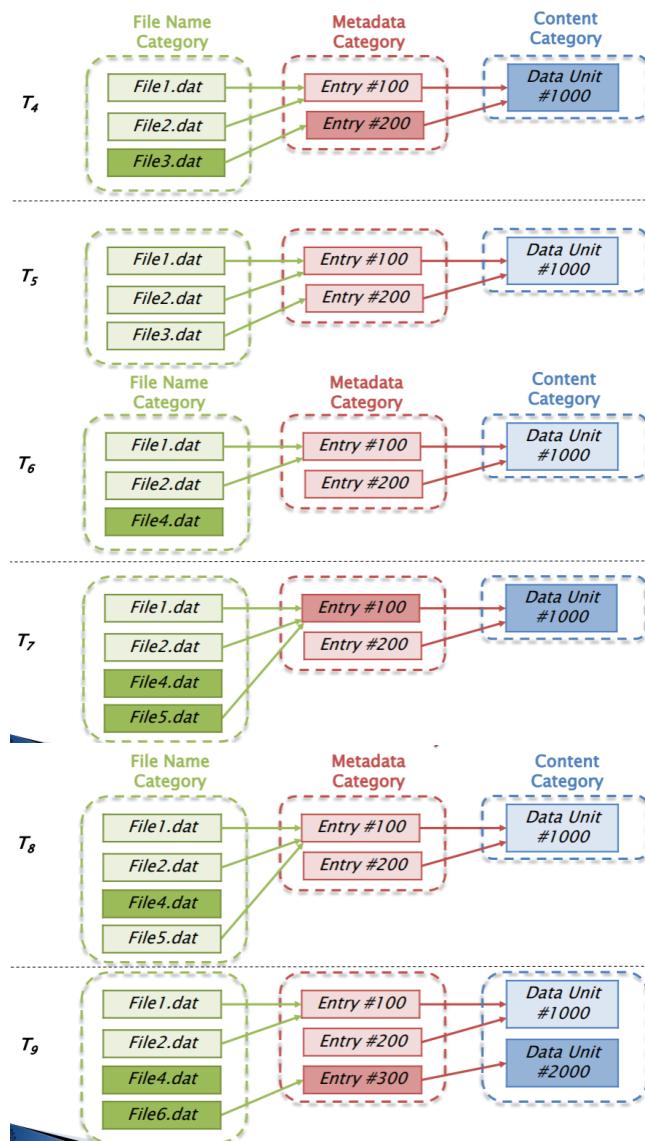
14.7 File Name Category

Questa categoria contiene i nomi dei file e concede all'utente di riferirsi ai file usando un nome invece che un indirizzo, quindi le informazioni di rilievo in questa categoria sono il nome del file e l'indirizzo della entry nella struttura metadata ad esso associata.

14.7.1 File Name Category - File Recovery

È possibile recuperare un file cancellato partendo dall'analisi del file name, recuperando la posizione nella struttura metadati che a sua volta dovrebbe contenere le informazioni sulla data unit dove era allocato il file. Anche qui si ripropongono le stesse problematiche per il recupero dati.





14.8 Application Category

Alcuni file system contengono dati relativi anche ad applicazioni, non sono essenziali per il file system, ma vengono comunque memorizzati in speciali strutture dati del file system affinché risultino più efficienti. Un esempio potrebbe essere il **Journaling** ovvero il principio che viene anche applicato nella gestione dei database, in pratica conserva le modifiche da effettuare e quella effettuate sui metadati in modo da evitare l'inconsistenza. In caso di crash del sistema, al suo riavvio, la cosa più logica è quella di controllare il **journal** per vedere se c'erano o ci sono operazioni importanti da eseguire ed in caso positivo il sistema può decidere tra:

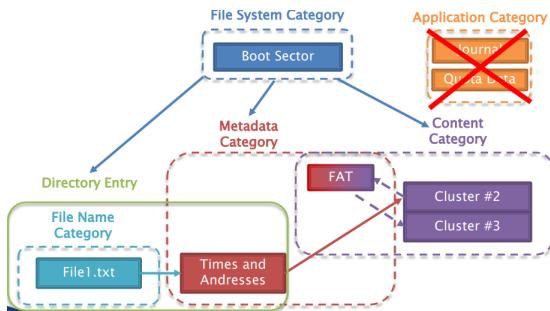
- **Completare le operazioni di modifica.**
- **Oppure ripristinare i dati a prima delle modifiche (rollback)**

14.8.1 Application Category - Analisi

Tramite l'analisi di questa categoria siamo in grado di ricostruire eventi di un incidente recente.

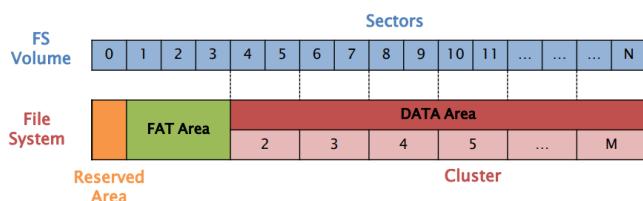
15 Lezione 16

15.1 FAT File System



Il concetto di base è che ogni file e directory vengono allocate in una struttura dati che si chiama **Directory Entry** che contiene il nome del file, la dimensione ed anche l'indirizzo iniziale del file all'interno della **Content Category** ed alcune informazioni che rientrano nella **Metadata Category**. File e directory sono memorizzate nelle data unit che qui però vengono chiamate **Cluster**, questi possono essere trovati nella struttura dati **FAT (File Allocation Table)** che è appunto utilizzata per identificare i successivi cluster di una determinato file ed a livello metadata contiene anche informazioni sulla allocazione del cluster. Esistono differenti versioni di FAT, tra cui, **FAT12**, **FAT16**, **FAT32**, la loro principale differenza sta nella dimensione delle entry nella struttura FAT.

15.1.1 Physical Layout



Composto da tre aree ben specifiche:

- **Reserved Area**: include quasi tutti i dati che rientrano nella file system category.
- **FAT Area**: vengono conservate le diverse strutture FAT principali, più ulteriori backup. La sua dimensione è data dal numero di FAT moltiplicato per la loro dimensione.

- **Data Area:** contiene i cluster utilizzati per memorizzare i file e le directory.

15.1.2 File System Category

I dati che rientrano in questa categoria servono a descrivere il file system. Vediamo dove e come il **FAT File System** memorizza questo tipo di dati e come analizzarli.

FSINFO (FAT32): struttura dati che contiene questo tipo di dati che è situata all'interno del **Boot Sector**, che è nella **Reserved Area**, ed è posizionato come primo settore del file system. Non viene specificata la tipologia di FAT all'interno della struttura dati e per determinarlo bisognerebbe effettuare un calcolo specifico su alcuni dati contenuti nel **Boot Sector**.

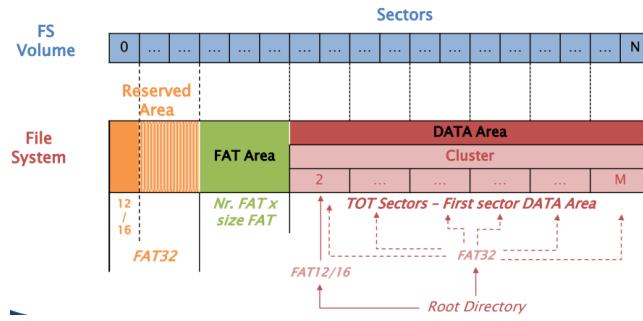
Nota Bene: nel Boot Sector di un FAT32 ci sono diversi dati aggiuntivi come un backup del boot sector e un'altra piccola struttura dati chiamata **File System Info** che contiene le informazioni aggiuntive sul file system, info non essenziali, ma che potrebbero tornare utili al sistema operativo.

Una delle prime cose da fare quando si analizza un FAT File System è quella di localizzare le tre aree fisiche del layout, la gran parte di queste informazioni sono contenute nel boot sector e sono per lo più dati essenziali.

La **Reserved Area** inizia la settore "ZERO" e la sua dimensione è descritta dal boot sector. [FAT12/16 la dimensione è di 1 settore, mentre per FAT32 la dimensione è variabile].

La **FAT Area** che contiene una o più strutture FAT ed inizia subito dopo la **Reserved Area**, la sua dimensione è data dal numero di strutture FAT che contiene per la loro dimensione, [**nr. FAT x Size FAT**] questi valori sono descritti nel boot sector.

La **Data Area:** contiene i cluster, essi memorizzano il contenuto di file e directory. La sua dimensione è data dalla sottrazione del totale numero di settori del file system dall'indirizzo del settore iniziale che lo compone, il numero totale di settori si trova nel boot sector. Un ulteriore valore interessante potrebbe essere il numero di settori che compongono un cluster. Con FAT12/16 all'interno della **Data Area** la **Root Directory** si trova all'inizio dell'area, mentre per FAT32 questa può essere posizionata in qualunque parte della **Data Area** ed infatti l'indirizzo della **Root Directory** lo troviamo riportato nel boot sector. Questa dinamicità della posizione della root è data dal fatto che così in caso di "*Bad Sector*" è più facile spostarla.



Nel boot sector non troviamo solo informazioni essenziali sul layout ma anche informazioni aggiuntive non essenziali. di conseguenza potrebbero essere inaffidabili. Uno di questi valori è rappresentato da una stringa di otto caratteri che si chiama **OEM Name** che indica info sullo strumento utilizzato per creare il file system. Inoltre il file system dedica 4 byte al **Volume Serial Number** che decodifica quella che è la data di creazione del file system. Infine ci sono ulteriori otto caratteri usati per la **File System Label** che dovrebbero in qualche modo informare sul tipo di file system presente.

15.1.3 Boot Sector

Byte	Description	Es.
0-2	Istruzioni assembly per saltare al bootcode	NO
3-10	OEM Name (ASCII)	NO
11-12	Dimensione settore (Byte)	SI
13	Dimensione Cluster (Settori) [x^2 max 32kb]	SI
14-15	Dimensione Reserved Area (Settori)	SI
16	Nr. di FAT [solitamente 2]	SI
17-18	Max nr. File in root directory [FAT12/16] 0 (ZERO) [FAT32] => Byte 36-39	SI
19-20	Tot. settori FS [se > 65.536 => 0; usare Byte 32-35]	SI
21	Media Type [f8 - dischi fissi, f0 - disp. removibili]	NO
22-23	Dimensione FAT (settori) [FAT12/16] 0 (ZERO) [FAT32]	SI
24-25	Nr. settori per traccia INT.13h	NO
26-27	Nr. Head dispositivo INT.13h	NO
28-31	Nr. settori prima dell'inizio della partizione	NO
32-35	Tot. settori FS [se < 65.536 => 0; usare Byte 19-20]	NO

Schema del boot sector che si trova nel primo settore del file system. I primi 36 byte sono uguali per le varie tipologie di FAT File System.

Dal byte 36, per i file system **FAT12** e **FAT16**, fino ad arrivare al byte 511, si sviluppa diversamente dal **FAT32**, questa è la struttura:

Byte	Description	Es.
36	BIOS INT.13h	NO
37	Non usato	NO
38	Extended boot signature: identifica se i successivi tre valori sono validi [29]	NO
39-42	Volume Serial Number [Windows lo genera utilizzando la data di creazione]	NO
43-53	Etichetta Volume (ASCII) [scelto dall'utente\tool al momento della creazione del FS]	NO
54-61	File System type (ASCII) [FAT, FAT12, FAT16]	NO
62-509	Non usato [boot code]	NO
510-511	Signature [AA55]	NO

Mentre questo è come si struttura il **FAT32**:

Byte	Description	Es.
36-39	Dimensione della FAT (settori)	SI
40-41	Nr. di FAT [se bit[7]=1 solo una delle FAT bit[0-3] è attiva, altrimenti mirror]	SI
42-43	Nr. di versione	SI
44-47	Posizione root directory (cluster)	SI
48-49	Posizione della struttura FSINFO (settori)	NO
50-51	Copia di backup del Boot Sector (settori) [6]	NO
52-63	Riservati	NO
64	BIOS INT.	NO
65	Non usato	NO
66	Extended boot signature: identifica se i successivi tre valori sono validi [29]	NO
67-70	Volume SN [Windows lo genera utilizzando la data di creazione]	NO
71-81	Etichetta Volume (ASCII)	NO
82-89	File System type (ASCII) [FAT32]	NO
90-509	Non usato [boot code]	NO
510-511	Signature [AA55]	NO

15.1.4 Boot Sector - Analisi

Analisi manuale, tramite un editor esadecimale, del boot sector per trovare informazioni salienti su un'immagine con file system **FAT32**. Per facilitare l'analisi leggendo a blocchi l'esadecimale possiamo utilizzare il comando **blkcat**.

```
root@caine:/# blkcat -f fat fat-4.dd 0 | xxd
00000000: eb58 904d 5344 4f53 352e 3000 0202 2600 .X.MSDOS5.0...&
00000016: 0200 0000 00f8 0000 3f00 4000 c089 0100 .....?@.....
0000032: 4023 0300 1d03 0000 0000 0000 0200 0000 @#.....[...]
```

Si posiziona su un singolo settore, sul settore zero per la precisione, come notiamo prima della pipeline.

Byte	Description	Value
3-10	OEM Name (ASCII)	MSDOSS.0
11-12	Dim. settore (Byte)	0200 (512)
13	Dim. Cluster (Settori)	2
14-15	Dim. Reserved Area (Settori)	0026 (38)
16	Nr. di FAT	2
17-18	Max nr. File in root directory	0
19-20	Tot. settori FS	0
21	Media Type	f8(disco fisso)
22-23	Dim. FAT (settori)	0
28-31	Nr. settori prima dell'inizio della partizione	000189c0 (100.800)
32-35	Tot. settori FS	00032340 (205.632)
Byte	Description	Value
36-39	Dimensione della FAT (settore)	00031d (797)
44-47	Posizione root directory (cluster)	00000002 (2)
48-49	Posizione della struttura FSINFO (settore)	0001 (1)
50-51	Copia di backup del Boot Sector (settori)	0006 (6)
67-70	Volume SN	4c194603
71-81	Etichetta Volume (ASCII)	«NO NAME »
82-89	File System type (ASCII) [FAT32]	«FAT32 »
510-511	Signature	AA55

```
root@caine:/# fsstat -f fat fat-4.dd
FILE SYSTEM INFORMATION
-----
File System type: FAT
OEM Name: MSDOS5.0
Volume ID: 0x4c194603
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory): FAT DISK
File System Type Label: FAT32
Backup Boot Sector Location: 6
FS Info Sector Location: 1
Next Free Sector (FS Info): 1778
Free Sector Count (FS Info): 203836
Sectors before file system: 100800

File System Layout (in sectors)
Total Range: 0 - 205631
* Reserved: 0 - 37
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 38 - 834
* FAT 1: 835 - 1631
* Data Area: 1632 - 205631
** Cluster Area: 1632 - 205631
*** Root Directory: 1632 - 1635

CONTENT-DATA INFORMATION
-----
Sector Size: 512
Cluster Size: 1024
Total Cluster Range: 2 - 102001
[...]
```

Uno strumento piuttosto utile a velocizzare l'analisi del boot sector è **fsstat**, questo raccoglie info della File System Category.

15.1.5 FSINFO

Byte	Description	Es.
0-3	Signature [41615252]	NO
4-483	Non usato	NO
484-487	Signature [61417272]	NO
488-491	Nr. di Cluster liberi	NO
492-495	Prossimo Cluster libero	NO
496-507	Non usato	NO
508-511	Signature [AA550000]	NO

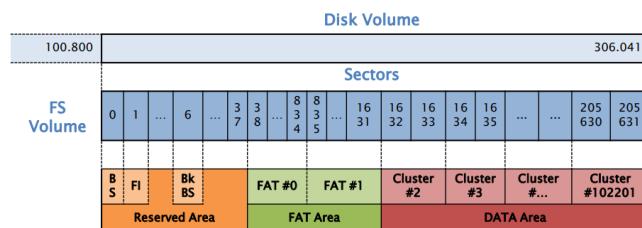
FSINFO situato nel settore UNO tiene traccia di informazioni per l'allocazione di nuovi cluster.

15.1.6 FSINFO - Analisi

root@caine:/# blkcat -f fat fat-4.dd 1 xxd		
Byte	Description	Value
0-3	Signature	41615252
484-487	Signature	61417272
488-491	Nr. di Cluster liberi	00018e1e (101.918)
492-495	Prossimo Cluster libero	0000004b (75)
508-511	Signature	AA550000

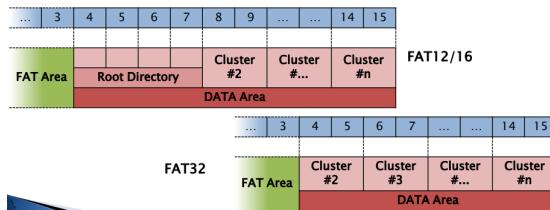
Grazie a **blkcat** riusciamo a posizionarci sul settore UNO. Come informazioni utili ci sono le tre signature.

15.1.7 Physical Layout Updated



15.2 Content Category

Qui ricadono dati riguardanti file e directory, qui i cluster possono contenere un numero di settori che è una potenza di due, fino ad un massimo di 64 settori, ovvero un cluster è al massimo 32KB. Ogni cluster ha un proprio indirizzo e il primo inizia dal DUE. I settori sono organizzati in cluster solo nella data area, nelle altre parleremo solo di settori. Un informazione utile in fase di analisi potrebbe essere l'indirizzo del primo cluster, questo cambia a seconda della tipologia di fat.

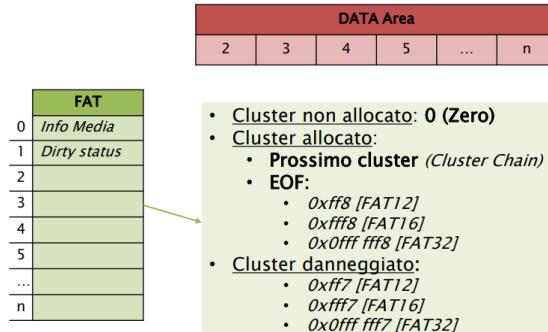


15.2.1 FAT Structure

L'identificativo dello stato di allocazione dei cluster si trova nella struttura dati FAT all'interno della FAT Area. Questa struttura ha due compiti principali, uno è quello dello stato di allocazione dei cluster e poi ha il compito di indicare il loro successivo cluster in cui è memorizzata la successiva parte di un file/directory. La struttura FAT ha entry di dimensioni uguali, a seconda del tipo di fat, se abbiamo **FAT12** allora saranno da 12 byte, **FAT16** allora 16 byte, mentre per **FAT32** avremo 32 byte. L'indirizzamento tra i cluster e le entry avviene in maniera diretta, ovvero hanno stesso indirizzo. Dato che i cluster iniziano dall'indirizzo DUE, le prime due entry non fanno riferimento ai cluster.

- **Indirizzamento diretto:**
 - La prima entry ha indirizzo 0 ZERO
 - Indirizzo entry = Indirizzo Cluster: *Ese. Entry[10]=Cluster[10]*
 - Entry[0]: informazione del media
 - Entry[1]: dirty status
 - Entry[2] -> Cluster[2]
 - Entry[n] -> Cluster[n]

Valori contenuti nella **FAT Table**.



15.2.2 FAT - Analisi

```
root@caine:/# blkcat -f fat fat-4.dd 38 | xxd
[...]
0000288: 4900 0000 4a00 0000 4c00 0000 0000 0000 I....J....L.....
0000304: 4d00 0000 ffff ff0f 4f00 0000 ffff ff0f M.....O.....
0000320: 5100 0000 5200 0000 ffff ff0f ffff ff0f Q...R.....
0000336: ffff ff0f 0000 0000 0000 0000 0000 0000 .....
0000352: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

Entry: 32Byte
(Offset/4)

Entry #	Byte	Valore
72	288-291	00000049 (73)
73	292-295	0000004a (74)
74	296-299	0000004c (76)
75	300-303	00000000 (0)
76	304-307	0000004d (77)
...
85	340-343	00000000 (0)

Stiamo analizzando un **FAT32** come possiamo vedere abbiamo entry di 32 byte. Con **blkcat** ci posizioniamo esattamente all'inizio della FAT Area, a partire quindi dal settore 38, ed iniziamo a leggere la prima struttura FAT che troviamo.

Qualora volessimo analizzare il contenuto di uno dei cluster bisogna capire prima a quale settore è indirizzato, ovvero da dove inizia il cluster. Per fare ciò è sufficiente un calcolo, ovvero:

$$\text{SETTORE} = (\text{cluster_address} - \underbrace{2}_{\text{indirizzo di inizio cluster}}) * \text{n_sect_cluster} + \text{sect_cluster_2}$$

Ad esempio: consideriamo il cluster 75

$$(75 - 2) * 2 + \text{sect_cluster_2}$$

Per calcolare il **primo settore data area** ci basta fare:

$$\underbrace{38}_{\text{dim.reserved area}} + \underbrace{1594}_{\text{dim.fat area}} = 1632$$

$$(75 - 2) * 2 + 1632 = 1778 \rightarrow \text{Settore di inizio del cluster 75}$$

Attraverso **blkstat** possiamo andare velocemente a recuperare l'informazione inversa, ovvero trovare il cluster conoscendo il settore esatto.

```
root@caine:/# blkstat -f fat fat-4.dd 1778
```

```
Sector: 1778
Not Allocated
Cluster: 75
```

15.3 Metadata Category

Qui rientrano informazioni sui file e directory, come ad esempio la posizione del primo cluster, ovvero da dove parte la cluster chain, ed altre informazioni temporali e di permessi sui file. Tutte queste informazioni sono memorizzate all'interno di una struttura dati detta **Parent Directory** che è a sua volta composta da **Directory Entry** che vengono allocate per ogni file o directory ed hanno uno spazio di 32KB, questa struttura è posizionata nella data area all'interno dei cluster, inoltre questa struttura gioca un ruolo anche per il file name category memorizzando anche i nomi, supportando fino ad un massimo di 8 caratteri per il nome e 3 caratteri per l'estensione.

15.3.1 Directory Entry

Byte	Description	Es.
0	- Primo carattere del filename (ASCII) - 0x5 o 0x0 [non allocato]	SI
1-10	Caratteri da 2 a 11 del filename (ASCII)	SI
11	Attributo File	SI
12	Riservato	NO
13	Ora di creazione (decimi di secondo)	NO
14-15	Ora di creazione (ora, minuti, secondi)	NO
16-17	Data di Creazione	NO
18-19	Data di Accesso	NO
20-21	- Indirizzo del primo cluster (High Byte) - 0 (ZERO) [FAT12/16]	SI
22-23	Ora di Modifica (ora, minuti, secondi)	NO
24-25	Data di Modifica	NO
26-27	Indirizzo del primo cluster (Low Byte)	SI
28-31	- Dimensione del file - 0 (ZERO) per le directory	SI

Attributo File [Byte 11]		
Flag Value bit	Description	Es.
0000 0001 (01)	Sola lettura	NO
0000 0010 (02)	File nascosto	NO
0000 0100 (04)	File di sistema	NO
0000 1000 (08)	Etichetta volume	SI
0000 1111 (0F)	Long File name	SI
0001 0000 (10)	Directory	SI
0010 0000 (20)	Archive	NO

15.3.2 Directory Entry - Analisi

Tramite il comando **blkcat** ci posizioniamo sul settore 1632 che sappiamo essere la root directory grazie al boot sector.

```
root@caine:/# blkcat -f fat fat-4.dd 1632 | xxd
0000000: 4641 5420 4449 534b 2020 2008 0000 0000 FAT DISK .....
0000016: 0000 0000 0000 874d 252b 0000 0000 0000 .....M%+.....
0000032: 5245 5355 4d45 2d31 5254 4620 00a3 347e RESUME-1RTF ..4~
0000048: 4a30 8830 0000 4a33 7830 0900 f121 0000 .0.0....0....!..
```

Byte	Description	Value
0	Fila Name - Primo carattere	F
1-10	Fila Name - Dieci caratteri	«AT DISK »
11	Attributo File	08 (0000 1000) [Etichetta Volume]
22-23	Ora di Modifica	4d87
24-25	Data di Modifica	2b25
Byte	Description	Value
22-23	Ora di Modifica	4d87
24-25	Data di Modifica	2b25
0 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1		
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Anno (0-127) + 1980	Mese (1-12)	Giorno (1-31)
2001	9	5
0 1 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1		
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Ora (0-23)	Minuti (0-59)	Secondi (0-29) x 2
9	44	14

```
root@caine:/# blkcat -f fat fat-4.dd 1632 | xxd
0000000: 4641 5420 4449 534b 2020 2008 0000 0000 FAT DISK .....
0000016: 0000 0000 0000 874d 252b 0000 0000 0000 .....M%+.....
0000032: 5245 5355 4d45 2d31 5254 4620 00a3 347e RESUME-1RTF ..4~
0000048: 4a30 8830 0000 4a33 7830 0900 f121 0000 .0.0....0....!..
```

Byte	Description	Value
0	Fila Name - Primo carattere	R
1-10	Fila Name - Dieci caratteri	«ESUME-1.RTF»
11	Attributo File	20 (0010 0000) [Archive]
13	Ora di Creazione (decimi s)	a3 (163)
14-15	Ora di Creazione	7e34 (15:49:40)
16-17	Data di Creazione	304a (10/02/2004)
20-21	Indirizzo Primo cluster File	0000 0009 (9)
26-27	Dimensione del file	000021f1 (8.689)

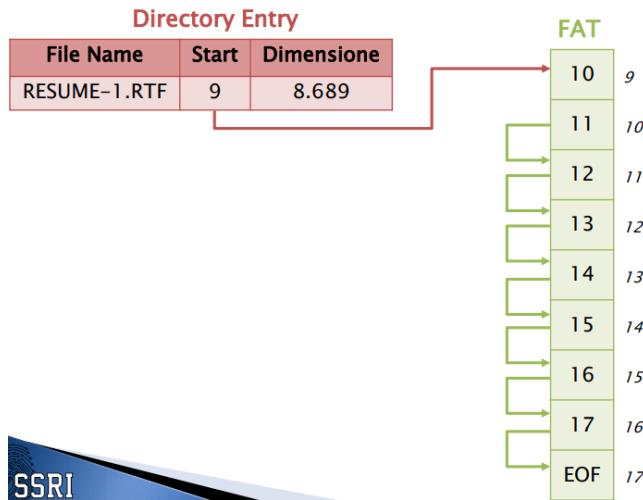
Notiamo il parametro compreso tra il byte 28 – 31, ovvero la dimensione del file, questa ci fa capire che il file è suddiviso in più cluster. Facendo un calcolo veloce, sono circa Nove cluster 8689/1024.

Per analizzare gli ulteriori cluster e sapere dove si trovano dobbiamo cercare nella **FAT Table** posizionandoci sul settore 36.

```
root@caine:/# blkcat -f fat fat-4.dd 38 | xxd
[...]
0000032: ffff ff0f 0a00 0000 0b00 0000 0c00 0000 .....
0000048: 0d00 0000 0e00 0000 0f00 0000 1000 0000 .....
0000064: 1100 0000 ffff ff0f 1300 0000 1400 0000 .....
```

Entry #	Byte	Valore
9	36-39	0000000a (10)
10	40-43	0000000b (11)
11	44-47	0000000c (12)
12	48-51	0000000d (13)
13	52-55	0000000e (14)
14	56-59	0000000f (15)
15	60-63	00000010 (16)
16	64-67	00000011 (17)
17	68-71	0fffffff (EOF)

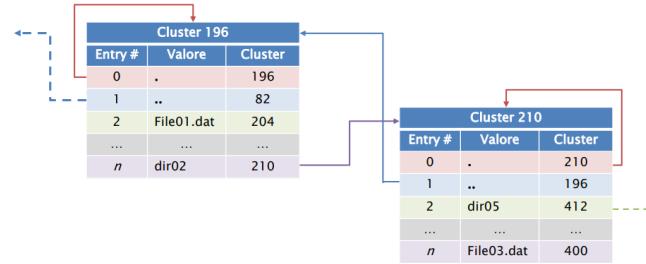
15.3.3 FAT - Cluster Chain



Per ricostruire il contenuto del file bisogna poi vedere ogni cluster all'interno della data area.

15.3.4 Metadata Category - Directory

Quando una nuova directory viene creata, un cluster viene allocato ed il suo contenuto viene pulito, quindi viene creato un cluster che contenga la nuova struttura di directory entry. Per conoscere la dimensione di una directory basta partire dal cluster iniziale e ricostruire la cluster chain dalla struttura FAT. Analizzando il cluster di una directory notiamo:



La prima entry rappresenta se stessa ed è indicata con il punto. La seconda entry invece è un puntatore al suo parent indicato con due punti. Dalla entry numero DUE in poi abbiamo il contenuto della directory stessa.

15.3.5 Metadata Category - Informazioni Temporali

Essendo le informazioni temporali di carattere non essenziale la loro alterazione o non modifica potrebbe essere problematica. Per i sistemi Windows, la **Data di Creazione** viene impostata al momento in cui deve essere allocata una nuova entry per un nuovo file, ad esempio se sposto o rinomino un file viene creata una nuova directory entry in cui la data di creazione è però copiata dalla precedente; Invece con creazione o copia di nuovi file la data di creazione è creata nuova.

Per quanto riguarda la **Data di Modifica** invece in caso di copia/-spostamento/rinomina del file la data viene semplicemente copiata. La **Data di Accesso** viene aggiornata anche semplicemente visualizzando le proprietà del file.

- Data di creazione (Windows)
 - Nuovo File/Copia File => Nuova data
 - Sposto/Rinomino => copia della data
- Data di Modifica (Windows): modifica del contenuto
 - Copia/Sposto/Rinomino File => copia della data
- Data di Accesso (Windows):
 - Modificata anche visualizzando le proprietà

15.4 File Name Category

Lo scopo di analisi dei dati in questa categoria è quello di mappare la struttura metadata con un nome, ovvero assegnare i nomi ai file. Il FAT File System non fa uso di una struttura dedicata alla categoria filename ma in realtà esse sono inglobate nell' stessa struttura dati che contiene informazioni sui metadata file ovvero nella directory entry.

Nota Bene: Nel FAT File System per nomi superiori agli OTTO caratteri si fa uso di una struttura chiamata **Long File Name** sempre grande 32byte per rientrare nella directory entry.

Cluster 196		
Entry #	Valore	Cluster
0	.	196
1	..	82
2	FileSys.TXT	204
3	TextFileFAT	204
4	TE021F~1.TXT	204
...

Il nome più lungo viene rappresentato da più directory entry, la prima avrà con sé anche tutte le altre informazioni, come quelle temporali, e tutte punteranno allo stesso cluster.

Byte	Description	Es.
0	Nr. sequenza (bit)	SI
1–10	Nome File [caratteri da 1 a 5]	SI
11	Attributo file [0f]	SI
12	Reserved	NO
13	Checksum	SI
14–25	Nome File [caratteri da 6 a 11]	SI
26–27	Reserved	NO
28–31	Nome File [caratteri da 12 a 13]	SI

16 Lezione 17

16.1 NT File System

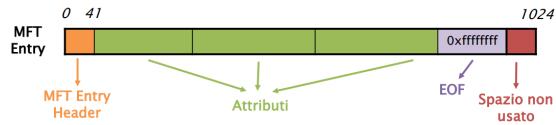
New Technologies File System (NTFS) fu disegnato da Microsoft nel '93 e divenne di default per i suoi sistemi operativi. NTFS è scalabile e fa uso di generiche strutture dati che poi possono cambiare ogni qualvolta vengono fatte richieste al file system. Una nota importante di questo file system è che esso gestisce tutto mediante l'uso di file, ad esempio, i dati che riguardano la gestione del file system sono localizzati dovunque all'interno del volume e sono organizzati in file, questo è possibile in quanto NTFS non fa uso di un layout rigido, rispetto a quello del FAT File System, e in realtà l'intero file system è considerato come data area se paragonato ad esso, quindi ogni settore può essere allocato per contenere file. Il layout di NTFS descrive in maniera rigida solo i primi settori del volume che servono a contenere il boot sector ed il boot code.

- ▶ Ogni cosa è un file:
 - **\$MFT:** *Master File Table*
 - **\$MFTMirr:** *backup della MFT*
 - **\$Boot:** *boot sector*
 - **\$Volume:** *informazioni del volume*
 - **\$Bitmap:** *stato di allocazione dei cluster*
 - **\$AttDef:** *definizione degli attributi*
 - **\$BadClus:** *elenco dei cluster danneggiati*
 - **\$Secure:** *descrittore di sicurezza*
 - **\$I30:** *Index*
 - ...

16.1.1 Master File Table

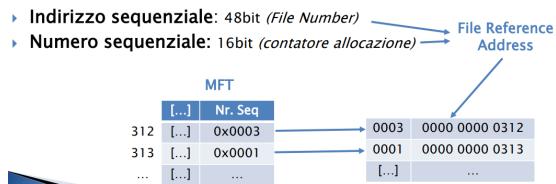
Il cuore di NTFS è la **Master File Table (\$MFT)**, ovvero la struttura portante nella quale vengono contenute tutte le informazioni che riguardano file e directory, ogni file o directory ha almeno una entry in questa tabella, Microsoft ha deciso di nominare le entry **File Record**, ma le chiameremo per comodità **MFT Entry**, queste sono grandi 1024byte, ma solo i primi 42byte hanno uno scopo predefinito, mentre i restanti servono a memorizzare gli **Attributi** che sono delle piccole strutture dati che hanno uno scopo specifico. Ogni entry ha il proprio indirizzo e la prima entry parte da zero. La prima entry di MFT descrive se stessa ed è chiamata **\$MFT** e descrive la posizione su disco di dove sono immagazzinati i dati di MFT. La posizione iniziale di MFT la troviamo nel boot sector.

Come per il FAT File System anche in NTFS un gruppo di settori è chiamato **Cluster**.



16.1.2 MFT Entry

- **Dimensione:** 1024byte (Default) definita nel boot sector (starter cluster) i primi 42byte sono per l'header e sono prestabiliti e strutturati in 12 campi ed i restanti byte non sono strutturati, serviranno a contenere gli attributi.
- **Signature:** rappresentata dalla stringa "FILE" e se la entry è danneggiata troveremo "BAAD".
- **Stato di Allocazione:** attributo \$BITMAP nella entry[0] \$MFT.
- **Indirizzo Sequenziale:** se un file non può essere descritto con una sola entry, allora la MFT utilizzerà più entry, in questo caso la prima entry viene chiamata **Base MFT Entry** e le successive entry conterranno l'indirizzo della base entry. Ogni entry ha un indirizzo sequenziale su 48bit, la prima entry ha indirizzo ZERO e quello massimo cambia a seconda della MFT ed è determinato dividendo la dimensione della MFT per la dimensione massima di ciascuna entry.
- **Numero Sequenziale:** numero a 16bit che aumenta ogni volta che la entry viene allocata.



File Reference Address: è l'unione tra indirizzo sequenziale e numero sequenziale. Questo meccanismo è utile in fase di recupero di file cancellati.

Byte	Description	Es.
0-3	Signature (ASCII) [FILE BAAD]	NO
4-5	Offset to fixup array	YES
6-7	Number of entries in fixup array	YES
8-15	\$LogFile Sequence Number	NO
16-17	Sequence value	NO
18-19	Link count	NO
20-21	Offset to first attribute	YES
22-23	Flags [01:in use 02:directory]	YES
24-27	Used size of MFT entry	YES
28-31	Allocated size of MFT entry	YES
32-39	File reference to base record	NO
40-41	Next attribute ID	NO
42-1023	Attributes and fixup values	YES

Struttura di una MFT Entry.

16.1.3 MFT - Analisi

```
root@caine:/# icat -f ntfs ntfs1.dd 0-128 | xxd
0000000: 4649 4c45 3000 0300 4ba7 6401 0000 0000 FILE0...K.d.....
0000016: 0100 0100 3800 0100 b801 0000 0004 0000 ....8.....
0000032: 0000 0000 0000 0000 0600 0000 0000 0000 .....;.....
0000048: 5800 0000 0000 1000 0000 6000 0000 X.....;.....
[...]
00000496: 3101 b43a 0500 0000 ffff ffff 0000 5800 1:.....X.
00000512: 0000 0000 0000 0000 0000 0000 0000 0000 .....;.....
[...]
0001008: 0000 0000 0000 0000 0000 0000 0000 5800 .....;.....
```

Analisi di un'immagine con NT File System tramite l'uso di **icat**. Ci focalizziamo sulla prima entry.

Byte	Description	Value
0-3	Signature (ASCII)	«FILE»
16-17	Sequence value	0001 (1)
18-19	Link count	0001 (1)
20-21	Offset to first attribute	0038 (56)
22-23	Flags [01:in use 02:directory]	0001 (1)
32-39	File reference to base record	0
40-41	Next attribute id	0006 (1)
42-1023	Attributes and fixup values	

16.2 File System Metadata

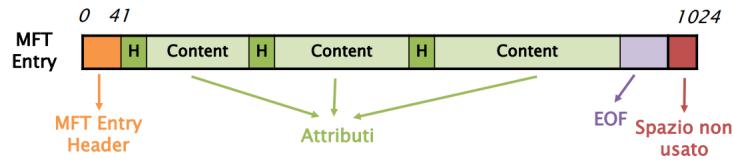
Dato che ogn byte all'interno del volume è utilizzato per contenere file allora ci saranno diversi file nel file system allocati per l'amministrazione del file system. Le prime 16 entry nell'ntf sono riservate ai file system metadata file, solo 12 entreranno però utilizzate, 4 sono lasciate per future implementazioni. Di solito queste tipologie di dati sono localizzate nella root directory e vengono nascosti all'utente; Ognuno di questi file inizia con il simbolo di dollaro ed una lettera maiuscola.

0	\$MFT	MFT Entry
1	\$MFTMirr	MFT Backup
2	\$LogFile	Journal
3	\$Volume	Volume Info
4	\$AttrDef	Attribute info
5	.	Root directory
6	\$Bitmap	Allocation status
7	\$Boot	Boot Sector, BootCode
8	\$BadClus	Cluster that have bad sector
9	\$Secure	Security Info
10	\$Upcase	Uppercase version of every Unicode character
11	\$Extend	Application category

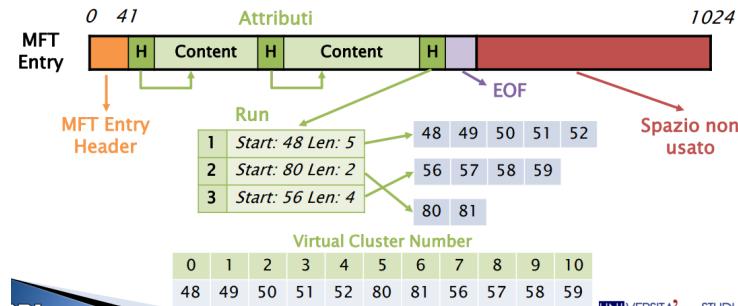
16.3 Attributi

Come abbiamo detto, una MFT Entry è composta da una piccola struttura interna predefinita (**MFT Entry Header**) e da una parte più grande utilizzata per memorizzare gli attributi, che abbiamo detto essere delle strutture dati per memorizzare determinati tipi di dato. Ci sono diversi tipi di attributi ed ognuno di essi ha la propria struttura interna, queste strutture servono a memorizzare ad esempio il nome del file, informazioni temporali o il contenuto del file. Tutto gli attributi sono costituiti da due parti, un **Header** ed il **Contenuto**. L'header è generico e sarà standard per ciascun attributo ed il contenuto sarà specifico per ogni tipo di attributo e può avere qualsiasi dimensione. L'header dell'attributo descrive l'attributo, il tipo, la dimensione, il nome e memorizza inoltre:

- **ID**: Identificatore univoco dell'entry 16bit.
- **Type ID**: Identificatore tipo di attributo.
- **Offset**: Attribute Content.

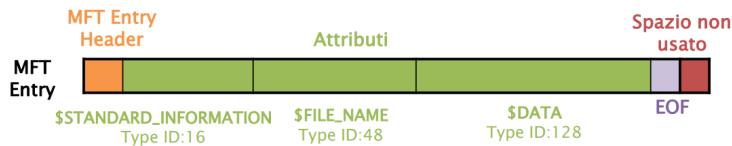


Il contenuto di un attributo, abbiamo detto, può avere dimensione variabile e può essere di qualsiasi formato, ad esempio l'attributo per memorizzare il contenuto di un file può essere di enormi dimensioni (MB, GB), quindi capiamo non essere pratico memorizzare questo tipo di dati nella entry MFT, quindi per risolvere questa problematica NTFS mette a disposizione due posizioni per memorizzare il contenuto degli attributi, ovvero se la dimensione del contenuto è piccola, l'attributo viene indicato come **Residente** e viene memorizzato nella stessa entry, al contrario se la dimensione è troppo grande allora l'attributo viene indicato come **NON Residente** e memorizzato in un cluster esterno. Informazioni sulla residenza o meno dell'attributo si trovano nell'header e nel caso l'attributo fosse non residente l'header avrà l'indirizzo al cluster esterno a cui fa riferimento, il contenuto dei cluster esterno è invece memorizzato una struttura detta **Cluster Run**.



16.3.1 Tipi di Attributi standard

Per ogni tipo di attributo è definito un numero che è il “*Type ID*” e Microsoft ordina gli attributi all’interno della entry proprio mediante questo numero. Ogni attributo standard ha un valore di default assegnato ma che potrebbe essere ridefinito dagli sviluppatori attraverso il file system metadata file nominato **\$ATTRDEF**. In aggiunta al numero ogni attributo ha un nome che inizia con un dollaro \$ e la lettera maiuscola. Le entry definite principali devono contenere obbligatoriamente gli attributi **\$FILENAME** e **\$STANDARD_INFORMATION**.



16	\$STANDARD_INFORMATION	<i>General information, such as flags; the last accessed, written, and created times; and the owner and security ID</i>
32	\$ATTRIBUTE_LIST	<i>List where other attributes for file can be found</i>
48	\$FILE_NAME	<i>File name, in Unicode, and the last accessed, written, and created times</i>
64	\$VOLUME_VERSION	<i>Volume information</i>
64	\$OBJECT_ID	<i>A 16-byte unique identifier for the file or directory</i>
80	\$SECURITY_DESCRIPTOR	<i>The access control and security properties of the file</i>
96	\$VOLUME_NAME	<i>Volume name</i>
112	\$VOLUME_INFORMATION	<i>File system version and other flags</i>
128	\$DATA	<i>File contents</i>
144	\$INDEX_ROOT	<i>Root node of an index tree</i>
160	\$INDEX_ALLOCATION	<i>Nodes of an index tree rooted in \$INDEX_ROOT attribute</i>
176	\$BITMAP	<i>A bitmap for the \$MFT file and for indexes</i>
192	\$SYMBOLIC_LINK	<i>Soft link information</i>
192	\$REPARSE_POINT	<i>Contains data about a reparse point</i>
208	\$EA_INFORMATION	<i>Used for backward compatibility with OS/2 applications (HPFS)</i>
224	\$EA	<i>Used for backward compatibility with OS/2 applications (HPFS)</i>
256	\$LOGGED.Utility_STREAM	<i>Contains keys and information about encrypted attributes</i>

16.4 Base/Non-Base MFT Entry

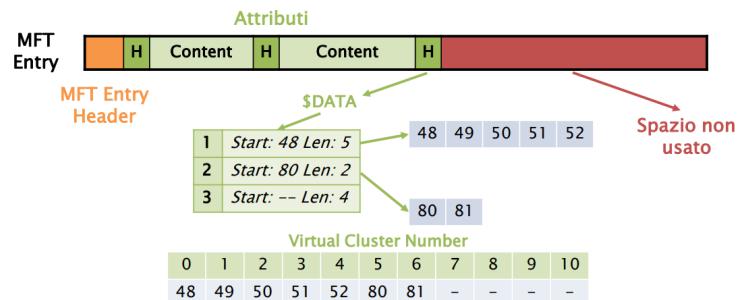
Un file può essere descritto fino ad un massimo di 65536 attributi, valora dato dai 16bit di identificatore. Quando una MFT Entry non ha sufficiente spazio per tutti i suoi attributi questi sono posizionati in altre entry. Per le base entry, oltre agli attributi di base viene anche aggiunto l’attributo **list** che conterrà le entry dove sono memorizzati

gli altri attributi. Le entry non-base non avranno gli attributi di base, quelli sono localizzati nella entry principale.



16.5 Sparse Attributes

NTFS può risparmiare spazio sul disco per memorizzare il contenuto di un attributo di tipo **\$DATA** se esso ad esempio contiene dei cluster vuoti. Se quindi NTFS trova dei cluster vuoti da dover memorizzare esso può indicare nei cluster run la presenza di questi cluster vuoti senza andarli ad allocare ed occupare spazio.



Nota Bene: NTFS permette anche la compressione degli attributi di tipo **\$DATA** e solo quando sono attributi non residenti. NTFS fa inoltre uso di molte strutture indirizzate, dove un indice NTFS è una collezione i attributi memorizzata in maniera ordinata (B-TREE).

16.6 Attribute Header

Esistono due tipi di Header, uno per attributi **residenti** ed un altro per quelli **non residenti**.

- **Residente:** I primi 16byte sono uguali in entrambe le tipologie di header. Dal byte 16 al 21 invece ci sono delle differenze. Il byte di partenza dell'attributo residente è il byte 56.

Byte	Description	Es.
0-3	Attribute type ID	YES
4-7	Length of attribute	YES
8	Non-resident flag	YES
9	Length of name	YES
10-11	Offset to name	YES
12-13	Flags	YES
14-15	Attribute identifier	YES
16-19	Size of content	YES
20-21	Offset to content	YES

Flags	
0x0001	compressed
0x4000	encrypted
0x8000	sparse

Resident Attribute

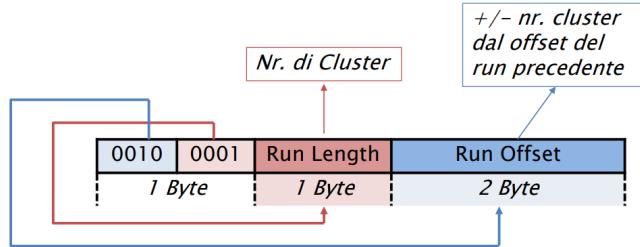
0000000: 1000 0000 6000 0000 0000 1800 0000 0000`.....		
0000016: 4800 0000 1800 0000 305a 7a1f f63b c301 H.....0Zz.;...		
Byte	Description	Value
0-3	Attribute type ID	00000010 (16) \$STANDARD_INFORMATION
4-7	Length of attribute	00000060 (96)
8	Non-resident flag	00 (0)
9	Length of name	00 (0)
12-13	Flags	0000 (0)
14-15	Attribute ID	0000 (0)
16-19	Size of content	00000048 (72)
20-21	Offset to content	0018 (24)

- **Non Residente:** Qui troviamo quelli che si chiamano **VCN Iniziali e Finali** che sono utilizzati quando sono necessarie più entry MFT per descrivere il singolo attributo.

Byte	Description	Es.
0-15	General Header	YES
16-23	Starting Virtual Cluster Number (VCN) of the runlist	YES
24-31	Ending VCN of the runlist	YES
32-33	Offset to the runlist	YES
34-35	Compression unit size	YES
36-39	Unused	NO
40-47	Allocated size of attribute content	NO
48-55	Actual size of attribute content	YES
56-63	Initialized size of attribute content	NO

Non-Resident Attribute

16.6.1 Run



Il primo byte è obbligatorio in una run, questo viene suddiviso in 4bit superiori e 4 bit inferiori. I 4 meno significativi servono ad esprimere la grandezza in byte del campo **Run Length**, i 4 più significativi invece esprimono il numero di byte del campo **Run Offset**.

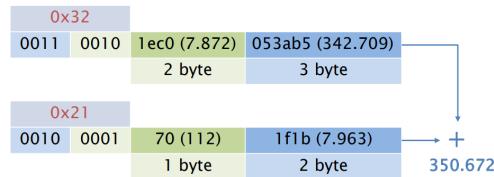
00000000: 8000 0000 6000 0000 0100 4000 0000 0100@.....
00000016: 0000 0000 0000 0000 ef20 0000 0000 0100@.....
00000032: 4000 0000 0000 0000 00c0 8300 0000 0000 @.....
00000048: 00c0 8300 0000 0000 00c0 8300 0000 0000@.....
00000064: 32c0 1eb5 3a05 2170 1b1f 2290 015f 7e31 2...!p.."-~1
00000080: 2076 ed00 2110 8700 00b0 6e82 4844 7e82 v...!.n.HD-

Byte	Description	Value
0-3	Attribute type ID	00000080 (128) SDATA
4-7	Length of attribute	00000060 (96)
8	Non-resident flag	01 (1)
9	Length of name	00 (0)
12-13	Flags	0000 (0)
14-15	Attribute identifier	0001 (1)
16-23	Starting VCN runlist	0
24-31	Ending VCN runlist	20ef (8.431)

00000000: 8000 0000 6000 0000 0100 4000 0000 0100@.....
00000016: 0000 0000 0000 0000 ef20 0000 0000 0100@.....
00000032: 4000 0000 0000 0000 00c0 8300 0000 0000 @.....
00000048: 00c0 8300 0000 0000 00c0 8300 0000 0000@.....
00000064: 32c0 1eb5 3a05 2170 1b1f 2290 015f 7e31 2...!p.."-~1
00000080: 2076 ed00 2110 8700 00b0 6e82 4844 7e82 v...!.n.HD-

Byte	Description	Value
32-33	Offset to the runlist	0040 (64)
40-47	Allocated size of attribute content	0083c000 (8.634.368)
48-55	Actual size of attribute content	0083c000 (8.634.368)
56-63	Initialized size of attribute content	0083c000 (8.634.368)

00000000: 8000 0000 6000 0000 0100 4000 0000 0100@.....
00000016: 0000 0000 0000 0000 ef20 0000 0000 0100@.....
00000032: 4000 0000 0000 0000 00c0 8300 0000 0000 @.....
00000048: 00c0 8300 0000 0000 00c0 8300 0000 0000@.....
00000064: 32c0 1eb5 3a05 2170 1b1f 2290 015f 7e31 2...!p.."-~1
00000080: 2076 ed00 2110 8700 00b0 6e82 4844 7e82 v...!.n.HD-



16.7 File System Category

Abbiamo detto che NTFS memorizza i dati della file system category in file detti **File System Metadata (MFT File)** questi servono a descrivere in maniera generica il file system. Il file più importante di questa categoria è il file **\$MFT** che contiene la **Master File Table** che a sua volta contiene ciascuna entry per ogni file e directory. L'indirizzo del primo cluster MFT è contenuto nel boot sector, mentre il suo layout è descritto dalla entry numero zero che è un riferimento all'MFT stesso. La **entry[0]** di MFT contiene:

- **\$DATA**: che indica i cluster usati per la struttura dati.
- **\$BITMAP**: indica lo stato di allocazione delle entry.

Analisi della prima entry del file **\$MFT**.

```
root@caine:/# istat -f ntfs ntfs1.dd 0
[...]
$STANDARD_INFORMATION Attribute Values:
Flags: Hidden, System
Owner ID: 0 Security ID: 256
Created: Thu Jun 26 10:17:57 2003
File Modified: Thu Jun 26 10:17:57 2003
MFT Modified: Thu Jun 26 10:17:57 2003
Accessed: Thu Jun 26 10:17:57 2003
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
Type: $FILE_NAME (48-3) Name: N/A Resident size: 74
Type: $DATA (128-1) Name: $Data Non-Resident size: 8634368
342709 342710 342711 342712 342713 342714 342715 342716
342717 342718 342719 342720 342721 342722 342723 342724
[...]
443956 443957 443958 443959 443960 443961 443962 443963

Type: $BITMAP (176-5) Name: N/A Non-Resident size: 1056
342708 414477 414478 414479
```

16.7.1 \$MFTMIRR File

Il file **\$MFT** è molto importante per ritrovare tutti gli altri file, questo potrebbe però essere il punto principale di fallimento in caso di corruzione del boot sector o della entry MFT, per risolvere allora questo problema esiste una copia di backup delle entry MFT più importanti che può essere usata per effettuare un ripristino. La entry[1] descrive questo backup e viene chiamata **\$MFTMIRR** che è un attributo di tipo non residente e che contiene una copia di backup delle prime 16 entry MFT.

```
root@caine:/# istat -f ntfs ntfs1.dd
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
Type: $FILE_NAME (48-2) Name: N/A Resident size: 82
Type: $DATA (128-1) Name: $Data Non-Resident size: 4096
514064 514065 514066 514067
```

16.7.2 \$BOOT File

Un altro file system metadata file molto importante è **\$BOOT** che contiene il boot sector del file system ed è descritto dalla entry numero 7, ha una posizione statica come il suo attributo **\$DATA**, si trova al primo settore del file system. Esiste anche una copia del boot sector che si trova all'ultimo settore del volume o nel mezzo del volume.

Byte	Description	Es.
0-2	Istruzioni assembly per saltare al bootcode	NO
3-10	OEM Name (ASCII)	NO
11-12	Dimensione settore (Byte)	YES
13	Dimensione Cluster (Settori)	YES
14-15	Settori riservati	NO
16-20	Non usati	NO
21	Descrizione Media	NO
22-23	Non usati	NO
24-31	Non usati	NO
32-35	Non usati	NO
36-39	Non usati	NO
40-47	Tot. settori FS	YES
48-55	Indirizzo del cluster iniziale di MFT	YES
56-63	Indirizzo del cluster iniziale di MFT Mirror	NO

Byte	Description	Es.
64	Dimensione delle entry MFT	YES
65-67	Non usati	NO
68	Dimensione dei record dell'index	YES
69-71	Non usati	NO
72-79	Serial Number	NO
80-83	Non usati	NO
84-509	Boot Code	NO
510-511	Signature (0xaa55)	NO

```
root@caine:/# istrat -f ntfs ntfs1.dd 7
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 48
Type: $FILE_NAME (48-2) Name: N/A Resident size: 76
Type: $SECURITY_DESCRIPTOR (80-3) Name: N/A Resident size: 104
Type: $DATA (128-1) Name: $Data Non-Resident size: 8192
0 1 2 3 4 5 6 7
```

Analisi della entry 7 per osservare il **\$BOOTFILE**.

16.7.3 **\$VOLUME** File

Un altro file system metadata file interessante è il **\$VOLUME** e viene descritto dalla entry numero 3 dell'MFT e contiene l'etichetta del volume e la sua versione. Ha due attributi che sono unici, nessun altro file dovrebbe averli, questi sono **\$VOLUME_NAME** e **\$VOLUME_INFORMATION**. Ovviamente come tutti i file c'è anche un attributo di tipo **\$DATA** ma esso sarà di ZERO byte.

Byte	Description	Es.	Flags
0-7	Unused	NO	
8	Major version	YES	
9	Minor version	YES	
10-11	Flags	NO	0x0001 Dirty 0x0002 ResizeLogFile 0x0004 Upgrade volume next time 0x0008 Mounted in NT 0x0010 Deleting change journal 0x0020 Repair object IDs 0x8000 Modified by chkdsk

```
root@caine:/# istrat -f ntfs ntfs1.dd 3
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 48
Type: $FILE_NAME (48-1) Name: N/A Resident size: 80
Type: $OBJECT_ID (64-6) Name: N/A Resident size: 16
Type: $SECURITY_DESCRIPTOR (80-2) Name: N/A Resident size: 104
Type: $VOLUME_NAME (96-4) Name: N/A Resident size: 22
Type: $VOLUME_INFORMATION (112-5) Name: N/A Resident size: 12
Type: $DATA (128-3) Name: $Data Resident size: 0
```

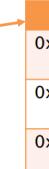
Leggendo la entry [3] possiamo analizzare il file **\$VOLUME**. Notiamo come l'attributo **\$DATA** sia vuoto.

16.7.4 \$ATTRDEF File

Un altro file system metadata file interessante è **\$ATTRDEF** che è descritto dalla entry[4] MFT, l'attributo **\$DATA** di questo file definisce i nomi e gli identificatori di tutti gli attributi presenti nel file system. Senza conoscere la posizione dell'attributo **\$DATA \$ATTRDEF** non si può conoscere la posizione degli altri attributi, questo potrebbe essere un problema.

Byte	Description	Es.
0–127	Name of attribute	YES
128–131	Type identifier	YES
132–135	Display rule	NO
136–139	Collation rule	NO
140–143	Flags	YES
144–151	Minimum size	NO
152–159	Maximum size	NO

Flags	
0x02	Attribute can be used in an index
0x04	Attribute is always resident
0x08	Attribute can be non-resident



```
root@caine:/# istat -f ntfs ntfs1.dd 4
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 48
Type: $FILE_NAME (48-2) Name: N/A Resident size: 82
Type: $SECURITY_DESCRIPTOR (80-3) Name: N/A Resident size: 104
Type: $DATA (128-4) Name: $Data Non-Resident size: 2560
342701 342702 342703
```

Questo è ciò che riscontriamo leggendo la entry[4].

16.8 File System Category - Analisi

La prima cosa utile da fare è processare il primo settore del file system dove si trova il boot sector e lì troveremo le info su dove è posizionata la MFT e quanto è grande ogni entry MFT, questo ci aiuta a capire come processare la MFT. Se qualche dato risulta corrotto basta esaminare il file di backup **\$MFTMIRR**. Ci sarebbe poi da processare i file **\$VOLUME**, per conoscere la versione del file system o l'etichetta ed infine **\$ATTRDEF** che ci definisce la presenza di possibili attributi speciali.

16.9 Content Category

Un NTFS non è altro che una raccolta di attributi, alcuni dei quali sono di tipo residente e quindi memorizzano il contenuto in cluster esterni. Un cluster invece sappiamo essere un gruppo di settori consecutivi, ed il numero di settori per un cluster è una potenza di due. Ogni cluster ha un indirizzo che parte da ZERO e il cluster ZERO inizia col primo settore del file system, molto meno confusionario rispetto al FAT File System.

16.9.1 \$BITMAP File

All'interno di questo tipo di file system metadata file vengono memorizzate le informazioni sulla allocazione di un cluster o meno. **\$BITMAP** viene descritto dalla entry[6] MFT e l'attributo **\$DATA** di questo file ha un bit per ogni cluster del file system. → **BIT[x] = CLUSTER [x]**

```
root@caine:/# istat -f ntfs ntfs1.dd 6
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
Type: $FILE_NAME (48-2) Name: N/A Resident size: 80
Type: $DATA (128-1) Name: $Data Non-Resident size: 128520
514113 514114 514115 514116 514117 514118 514119 514120
514121 514122 514123 514124 514125 514126 514127 514128
[...]
```

16.9.2 \$BADCLUS File

NTFS tramite l'utilizzo del file system metadata file **\$BADCLUS** tiene traccia dei cluster con settori danneggiati e li alloca all'attributo **\$DATA** di questo file che viene descritto dalla entry[8] MFT. Per evitare l'utilizzo di cluster danneggiati da parte del file system basta dichiararli come danneggiati ed allocare l'attributo **\$DATA** di questo metadata file che si chiama **\$BAD**.

```
root@caine:/# istrat -f ntfs ntfs1.dd 8
[...]
Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
Type: $FILE_NAME (48-3) Name: N/A Resident size: 82
Type: $DATA (128-2) Name: $Data Resident size: 0
Type: $DATA (128-1) Name: $Bad Non-Resident size: 1052803072
```

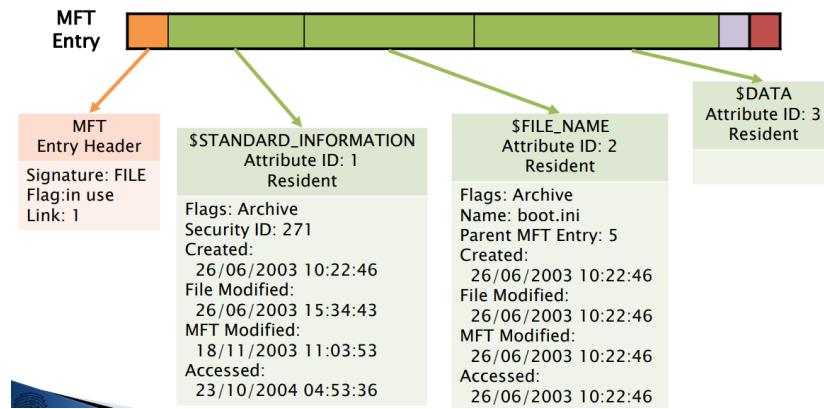
Ci sono due tipi di attributo **\$DATA**, il primo è vuoto e il secondo, che sarebbe **\$BAD** non serve a contenere il dato del file ma bensì serve a descrivere ed etichettare i file danneggiati.

17 Lezione 18

17.1 Metadata Category

I dati che ricadono in questa categoria sono quelli che descrivono file o directory. In realtà tutti i metadati sono memorizzati in uno degli attributi, un tipico file ha tre attributi, ovvero:

- **\$STANDARD_INFORMATION.**
- **\$FILE_NAME.**
- **\$DATA.**



17.1.1 \$STANDARD_INFORMATION Attribute

Questo attributo esiste per tutti i file e directory e contiene i metadati principali, qui infatti troviamo il primo set non esclusivo di timestamp, informazioni sulla proprietà, sulla sicurezza e sulla gestione delle quote. Nulla di questo attributo rientra nei dati essenziali per la memorizzazione del file, ma molte di queste funzionalità a livello applicativo dipendono dal sistema operativo. L'ID Default Type è il numero 16 ed ha una dimensione statica di 72byte.



Byte	Description	Es.	Flags
0-7	Creation time	NO	
8-15	File altered time	NO	
16-23	MFT altered time	NO	
24-31	File accessed time	NO	
32-35	Flags	NO	
36-39	Maximum number of versions	NO	
40-43	Version number	NO	
44-47	Class ID	NO	
48-51	Owner ID	NO	
52-55	Security ID	NO	
56-63	Quota Charged	NO	
64-71	Update Sequence Number (USN)	NO	

Flags	
0x0001	Read Only
0x0002	Hidden
0x0004	System
0x0020	Archive
0x0040	Device
0x0080	#Normal
0x0100	Temporary
0x0200	Sparse file
0x0400	Reparse point
0x0800	Compressed
0x1000	Offline
0x2000	Content is not being indexed for faster searches
0x4000	Encrypted

Ci sono 4 valori temporali (timestamp), che sono:

- **Data di Creazione:** creazione del file.
- **Data di Ultima Modifica:** modifica del contenuto degli attributi \$DATA ed \$INDEX.
- **Data di Ultima Modifica MFT:** modifica dei metadati del file.
- **Data di Ultimo Accesso:** accesso al contenuto del file.

17.1.2 \$FILE_NAME Attribute

Ogni file e directory ha almeno un attributo che si chiama **\$FILE_NAME** nella propria entry MFT. In realtà ogni file e directory ha almeno un'altra istanza di un attributo sempre chiamata **\$FILE_NAME** all'interno dell'indice della directory principale. **\$FILE_NAME**, nella entry MFT, ha un ID Type che è 48 ed ha una dimensione variabile dipendente dalla lunghezza del file, di base la dimensione è data da 66byte + Lunghezza nome. Questo attributo contiene il nome del file codificato in UTF-16, il nome inoltre deve trovarsi in uno spazio dei nomi ben preciso.

Namespace	
0	POSIX: The name is case sensitive and allows all Unicode characters except for '/' and NULL.
1	Win32: The name is case insensitive and allows most Unicode characters except for special values such as '/', '\', ';', '>', '<', and '?'.
2	DOS: The name is case insensitive, upper case, and no special characters. The name must have eight or fewer characters in the name and three or less in the extension
3	Win32 & DOS: Used when the original name already fits in the DOS namespace and two names are not needed.

Byte	Description	Es.	Flags
0–7	File reference of parent directory	NO	0x0001 Read Only
8–15	File creation time	NO	0x0002 Hidden
16–23	File modification time	NO	0x0004 System
24–31	MFT modification time	NO	0x0020 Archive
32–39	File accessed time	NO	0x0040 Device
40–47	Allocated size of file	NO	0x0080 #Normal
48–55	Real size of file	NO	0x0100 Temporary
56–59	Flags	NO	0x0200 Sparse file
60–63	Reparse value	NO	0x0400 Reparse point
64	Length of name	NO	0x0800 Compressed
65	Namespace	NO	0x1000 Offline
66+	Name	NO	0x2000 Content is not being indexed for faster searches
			0x4000 Encrypted

Oltre al nome del file questo attributo contiene anche un riferimento alla parent direcotry, un informazione così utile può essere usata per trovare l'intero percorso di un entry MFT che si sta leggendo. Contiene inoltre molti valori duplicati dallo **\$STANDARD_INFORMATION**, alcuni non vengono aggiornati.

17.1.3 \$DATA Attribute

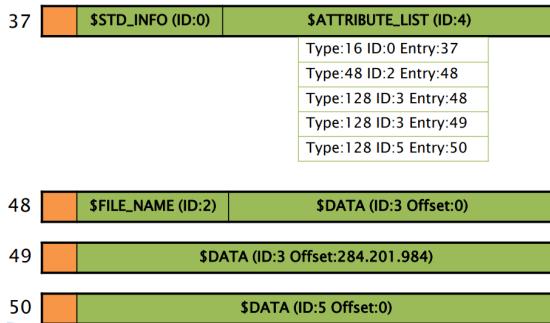
Viene utilizzato per memorizzare il contenuto del file, quindi memorizza qualsiasi forma di dati, non ha un formato o dei valori definiti. L'ID Type è 128 e può avere qualsiasi dimensione, incluso lo zero, se la sua dimensione è maggiore di 700byte i dati sono memorizzati in cluster esterni e l'attributo è di tipo non residente. Un attributo \$DATA è assegnato ad ogni file e quello di default non ha un nome. Un una entry MFT possono essere allocati più attributi \$DATA per lo stesso file ma in questo caso gli altri attributi di tipo \$DATA sono nascosti da occhi indiscreti, ovvero non sono visibili da chiunque ma servono strumenti specifici.

17.1.4 \$ATTRIBUTE_LIST Attribute

Questo attributo viene utilizzato quando un file o una directory ha bisogno di più di una entry MFT per memorizzare tutti i suoi attributi. Ha un ID Type che è 32, essendo basso ricadrà nella MFT di base, quindi si troverebbe, qualora esistesse, subito dopo lo \$STANDARD_INFORMATION.

Byte	Description	Es.
0–3	Attribute type	YES
4–5	Length of this entry	YES
6	Length of name	YES
7	Offset to name (relative to start of this entry)	YES
8–15	Starting VCN in attribute	YES
16–23	File reference where attribute is located	YES
24	Attribute ID	YES

Elenco di tutti gli attributi di un file.



17.1.5 \$SECURITY_DESCRIPTOR Attribute

Questo attributo è esclusivo delle versioni di NTFS precedenti alla 3.0, per le versioni successive esiste ma viene utilizzato solo per caso di retrocompatibilità. Windows usa questo tipo di attributi per descrivere i criteri di controllo dell'accesso che devono essere applicate a un file o una directory. Questo attributo ha ID Type 80. Le versioni più recenti di NTFS memorizzano i descrittori di sicurezza in un unico file esterno principale, in questo modo da risparmiare spazio. Questo file, per le versioni a partire dalla 3.0 in poi è chiamato **\$SECURE** File ed è descritto dalla entry[9] MFT che è composta da:

- **Indice \$SDH**: fa riferimento ai descriptor.
- **Indice \$SII**: fa riferimento ai descriptor.
- **Attributo \$DATA**: contiene i secure descriptor effettivi.

17.2 Metadata Category - Algoritmi di Allocazione

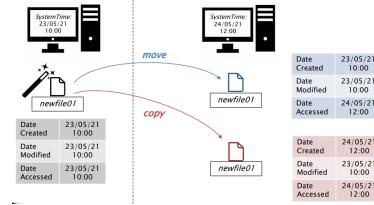
Esistono TRE strategie per l'allocazione dei metadati.

Per quanto riguarda l'allocazione delle entry MFT, microsoft, fa uso della strategia del **Primo Disponibile**, partendo dalla entry 24 il Sistema Operativo ricerca delle entry disponibili da poter usare e se non ce ne sono più allora ne crea una nuova. Quando un file viene cancellato la entry che lo descriveva non viene modificata, bensì viene modificata la flag, nell'header della entry, che identifica lo stato d'uso, passando dallo stato "*IN USO*" allo stato "*NON IN USO*" e quindi questo ci fa capire che è ancora possibile recuperare le informazioni temporali. Quando viene allocato un nuovo file, e viene quindi usata una entry "*NON IN USO*", questa viene prima pulita e poi il suo stato viene cambiato in "*IN USO*", così non si avranno quei dati di slack. Per quanto riguarda l'allocazione degli attributi, microsoft ordina le entry in base al tipo di attributo e se l'attributo **\$DATA** alla fine della entry è di tipo residente e la sua dimensione diminuisce, il suo vecchio contenuto può essere ancora trovato all'interno della entry nella parte finale perché viene solo spostato il marcatore di end-of-file. Quando un attributo aumenta la sua dimensione costringendo a diventare da "**RESIDENTE**" a "**NON RESIDENTE**", quello che era presente nella entry MFT esiste ancora fin quando non viene sovrascritto.

17.2.1 Aggiornamento informazioni temporali

Abbiamo detto che le info temporali si trovano all'interno degli attributi **\$STANDARD_INFORMATION** e **\$FILE_NAME**. Le informazioni temporali di **\$FILE_NAME** vengono aggiornate solo quando viene creato o spostato il file. Per le informazioni temporali di **\$STANDARD_INFORMATION** funziona più o meno come per FAT, ovvero il tempo di creazione è impostato per un nuovo file, se si crea un nuovo file da zero o si copia un file, l'ora di creazione del nuovo file viene impostata sull'ora corrente, se invece muovo un file, anche su un volume differente, l'ora di creazione non varia. La data di ultima modifica invece viene aggiornata ogni qual volta viene modificato il valore di un qualsiasi attributo di tipo **\$DATA** oppure **\$INDEX_ROOT** o **\$INDEX_ALLOCATION**.

- Data di creazione: creazione nuovo file o copia
- Data di ultima modifica: variazione degli attributi DATA, \$INDEX_ROOT o \$INDEX_ALLOCATION
- Data di ultima modifica MFT: modifica degli attributi
- Data di accesso: viene fatto accesso alla entry (metadati o contenuto)



17.3 Metadata Category - Analisi

La metadata category viene analizzata o su file o su una directory precisa, questo processo comporta l'individuazione di una entry MFT e l'elaborazione del suo contenuto. Per individuare una entry specifica bisogna prima individuare la MFT utilizzando l'indirizzo iniziale che si trova nel boot sector, dopo aver individuato la entry ne analizziamo gli attributi, qualora il file o la directory disponga di più entry esse verranno processate seguendo l'attributo **\$ATTRIBUTE_LIST**. Per processare una entry MFT ne elaboriamo prima l'header, poi il primo attributo, analizzandone l'header, il tipo e poi il contenuto in modo appropriato, e poi di conseguenza i successivi attributi della entry.

17.4 File Name Category

Categoria che include i dati per collegare il nome di un file al suo contenuto. Come correlare i nomi ai file ?

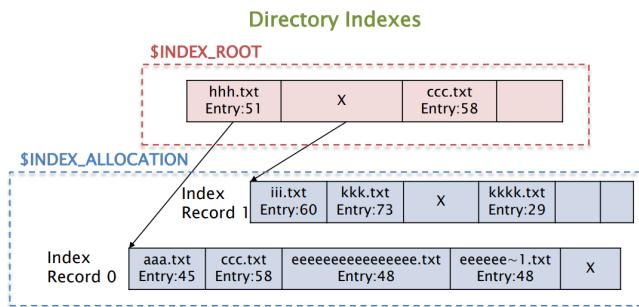
NTFS usa degli indici per organizzare i contenuto, un indice non è altro che una raccolta di strutture dati ordinate per chiave. L'albero B-TREE che si crea ha uno o più nodi che sono chiamati:

- **\$INDEX_ROOT**: indica sempre la radice dell'albero.
- **\$INDEX_ALLOCATION**: memorizza gli indici che servono ad archiare gli altri nodi.
- **\$BITMAP**.

Questi indici vengono chiamati **\$i30**.

Byte	Description	Es.
0–7	MFT file reference for file name	YES
8–9	Length of this entry	YES
10–11	Length of \$FILE_NAME attribute	NO
12–15	Flags	YES
16+	\$FILE_NAME Attribute	YES
Last 8	VCN of child node in \$INDEX_ALLOCATION	YES

Directory Index Entry Data Structure.



17.4.1 Root Directory

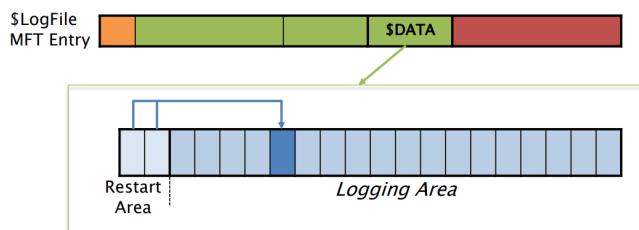
Con qualsiasi file system è fondamentale conoscere la posizione della **Root Directory** soprattutto se si desidera trovare un file in base al suo percorso completo. In NTFS la directory principale si trova sempre descritta nella entry[5] MFT ed è nominata con un punto “.”, in questa directory risiedono tutti i file system metadata file.

17.5 Application Category

Le funzionalità aggiunte con questo tipo di file system sono quelle che manipolano i dati della application category. NTFS fornisce supporto a molte funzionalità a livello applicativo, per questo risulta unico, funzionalità che devono essere racchiuse in un file system, ma permettono alle applicazioni di funzionare in modo più efficiente. Queste funzionalità non sono essenziali rispetto all'uso di un file system, un utente può quindi decidere di disattivare alcune delle funzionalità. Un file system può avere il supporto delle quote di spazio su disco, ovvero limitare lo spazio allocato da un utente, parte delle informazioni sulla quota sono archiviate come dati all'interno del file system, mentre altri dati sono memorizzati in un file a livello applicativo, come il registro di windows.

17.6 Logging/Journaling

Per migliorare l'affidabilità di un file system, microsoft, ha aggiunto il **Journaling**, di solito chiamato **Logging**, questa procedura permette di mantenere il file system in uno stato di consistenza, se ad esempio il S.O. viene spento durante un importante operazione di scrittura su disco, alla riaccensione il S.O. sarà in grado di portare il sistema in uno stato noto. Questo file è descritto nella entry[2] MFT e viene chiamato **\$LOGFILE**.



L'area di riavvio contiene due copie di una struttura dati che aiuta il S.O. quali transizioni devono essere esaminate. Contiene in realtà un contatore nell'area di logging per l'ultima transazione nota riuscita. Microsoft descrive due record:

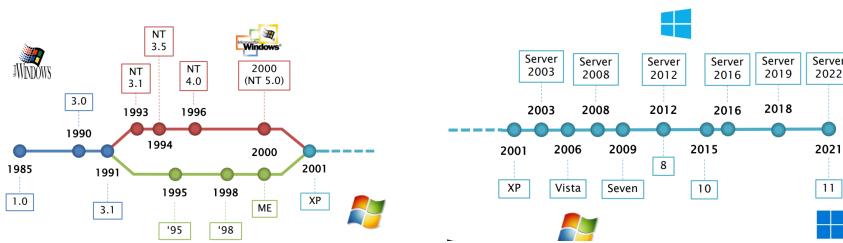
- **Record di Aggiornamento:** più utilizzato e descrive una transazione nel file system prima che si verifichi e dopo che è stata eseguita.
- **Record del Checkpoint:** ovvero da dove nel file di registro dovrebbe iniziare il S.O. per verificare la consistenza del file system. [Viene creato ogni 5 secondi]

18 Lezione 19

18.1 Sistemi Operativi

18.2 Microsoft Windows

18.2.1 Windows - Storia



18.2.2 Users

Gli account possono essere suddivisi in due categorie principali per quanto riguarda windows, sono:

- **Account Locali:** ovvero gli account locali alla macchina, permettono di accedere solo alla macchina su cui sono stati configurati. Possono essere di tipo semplice o anche di livello amministratore.
- **Account di Dominio:** sono quegli account che vengono autenticati tramite un domain controller che risiedono in un database detto **active directory** e ci possono essere diversi livelli di appartenenza e possono anche essere gestiti da una chace locale ad esempio per permettere l'autenticazione dell'utente anche in mancanza di un collegamento diretto tra il pc e il domain controller di riferimento.

Con l'arrivo di windows 8 venne introdotto anche un altro tipo di account, ovvero:

- **Account Online:** viene legato un utente ad un account microsoft; Ogni computer windows è come se facesse parte di un dominio comune che fa capo a microsot per la gestione dell'account. È possibile avere un account online e associarlo poi ad uno o più computer windows in modo tale da avere sincronizzate tutte le preferenze/impostazioni.

18.2.3 Secure Boot

Con l'arrivo di Windows 8, microsoft ha introdotto anche l'adozione di un nuovo **boot manager**, integrato nei sistemi avanzati del firmware UEFI, che è il **secure boot**, esso permette l'esecuzione solo di un sistema operativo trusted ovvero il proprio S.O. Windows. Questa opzione è disabilitabile nel bios qualora ad esempio si voglia eseguire una distro live forensics per effettuare delle analisi.

18.2.4 Registro di Sistema

Un'altra cosa peculiare è che Windows custodisce tutte le impostazioni ed opzioni del sistema operativo, come in parte anche alcune app installare, all'interno di una struttura dati detta **Registro di Sistema**, questa struttura dati è memorizzata grazie all'uso di più file che sono variati nel tempo a seconda della versione di Windows.

► Windows 95/98:

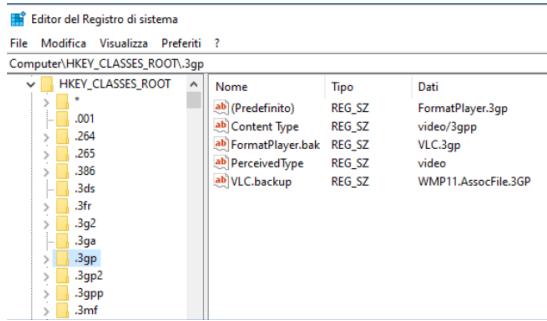
- User.dat:
 - \Windows
 - \Windows\Profiles\[user_name]
- System.dat:
 - \Windows

► Windows ≥ XP:

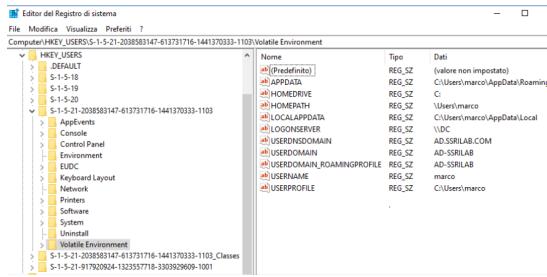
- Software, System, SAM, Security, Default:
 - \Windows\system32\config
- NTUser.dat:
 - \Documents and Settings\[user_name] (Windows XP)
 - \Users\[user_name] (Windows ≥ Vista)

A livello di struttura dati il registro di sistema non è altro che un albero composto da 5 sotto-alberi principali chiamati **Hive**, questi sotto-albero sono:

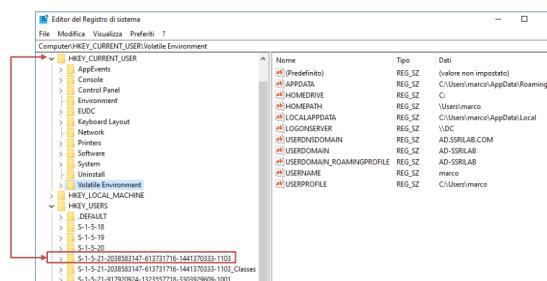
- **HKEY_CLASSES_ROOT**: contiene principalmente due tipi di informazioni, ovvero, contiene l'associazione tra i file e lo specifico programma che li gestisce e poi contiene alcuni dati di configurazione per alcuni componenti, come VisualBasics.



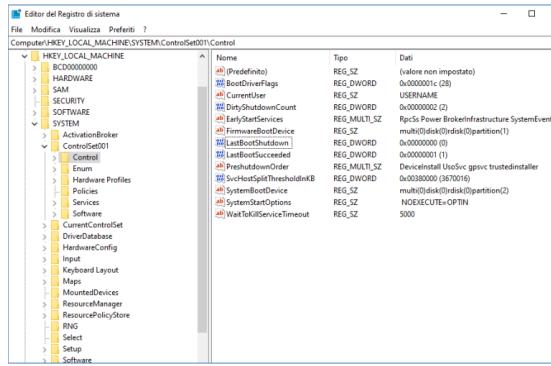
- **HKEY_USERS:** contiene le impostazioni di tutti i profili utenti configurati all'interno del sistema, ovvero le diverse informazioni che si trovano nei vari file “*NTUSER.dat*”.



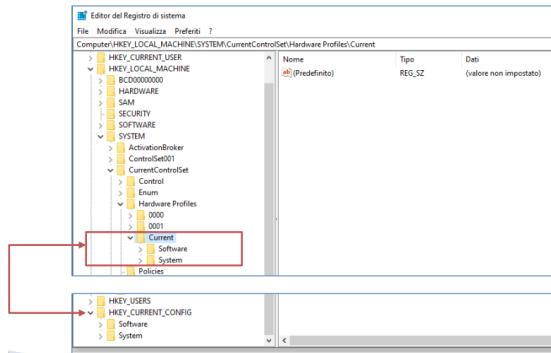
- **HKEY_CURRENT_USER:** contiene solamente un puntatore allo specifico profilo utente presente in **HKEY_USERS** che è attualmente loggato nel sistema.



- **HKEY_LOCAL_MACHINE**: contiene le informazioni relative alla configurazione del PC.



- **HKEY_CURRENT_CONFIG**: contiene esclusivamente un puntatore ad uno dei precisi sotto-alberi situati nel precedente **Hive** (HKEY_LOCAL_MACHINE).



Ogni nodo all'interno dell'albero è composto da:

- **CHIAVE**: è una coppia di valori espressa come (**NOMECHIAVE - VALORE**).
- **SOTTOCHIAVE**.

I valori di ogni chiave possono essere:

Tipi di chiavi	
Tipo	Descrizione
REG_SZ	NUL-terminated string
REG_EXPAND_SZ	NUL-terminated string (variabili di ambiente)
REG_BINARY	Dati binari
REG_DWORD / REG_DWORD_LITTLE_ENDIAN	4Byte (intero senza segno) [little endian]
REG_DWORD_BIG_ENDIAN	4Byte (intero senza segno) [big endian]
REG_LINK	Collegamento ad un'altra chiave
REG_MULTI_SZ	Array di NUL-terminated string
Tipi di chiavi	
Tipo	Descrizione
REG_RESOURCE_LIST	Elenco di risorse per un driver
REG_FULL_RESOURCE_DESCRIPTOR	Un descrittore di risorsa utilizzata da un driver
REG_RESOURCE_REQUIREMENTS_LIST	Un elenco requisiti delle risorse di un driver
REG_QWORD / REG_QWORD_LITTLE_ENDIAN	8Byte (intero senza segno) [little endian]
REG_NONE	Nessun tipo

18.2.5 Registro di Sistema - Analisi

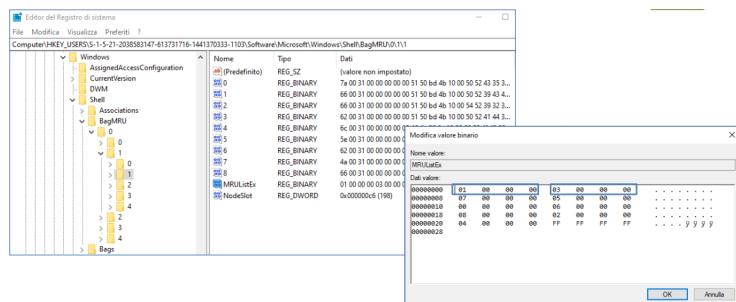
Risulta utile analizzare un registro di sistema dato che molte delle informazioni al suo interno sono rilevanti, tra le varie cose possiamo ritrovare i file recentemente aperti dall'utente, le configurazioni dell'utente oppure le marcature con cui windows ha contrassegnato i dispositivi USB collegati al sistema. Ogni chiave del registro di sistema ha in realtà associata una serie di valori, tra cui anche quello di ultima modifica, quindi è possibile ricostruire l'utilizzo del registro di sistema da parte del sistema stesso o dall'utente, è possibile introdurre ulteriori parametri per consolidare una timeline più elaborata. Lo strumento atto alla manipolazione del registro di sistema è **REGEDIT** per i sistemi windows, questo risulta utile per l'individuazione di qualche chiave ma per un'indagine risulta inadeguato, ci sono strumenti open source che permettono di cercare nel registro i dati o le keywords. [**Windows Registry Recovery**]

18.2.6 Thumbnails

Un'altra peculiarità di windows sono i **Thumbnails**, da Windows98 in poi il sistema crea dei file, chiamati **THUMBS.db**, nelle directory dove sono presenti delle immagini. Questo file contiene delle miniatura delle immagini presenti all'interno della cartella, lo scopo a livello di sistema è permettere una visualizzazione più veloce, le anteprime sono salvate nel file system. Nei file **THUMBS.db** le immagini sono memorizzate come una variante dei file bitmap. Da Windows Vista in poi questo database è stato centralizzato, quindi invece di essere presente in ogni cartella contenente immagini è stato centralizzato all'interno di ogni profilo utente, nel path **"%userprofile%\AppData\Local\Microsoft\Windows\Explorer"**, ed è chiamata **thumbcache_[NUM].db** dove NUM è la dimensione delle anteprime[96, 256, 1024], l'analisi di un database del genere potrebbe aiutarci a ritrovare artefatti di file che non sono più presenti nel sistema, questo perché il sistema difficilmente esegue controlli di coerenza tra i file in questo database ed i file ancora presenti nel sistema.

18.2.7 Shell Bag

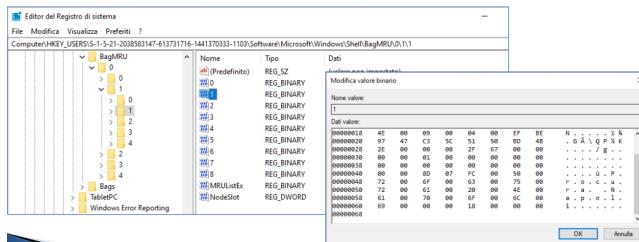
Un altro elemento che potrebbe essere utile durante un'analisi sono le **ShellBag**, è un meccanismo insito nella shell grafica di windows e consente di tenere traccia delle scelte fatte dall'utente in materia di personalizzazione della visualizzazione delle cartelle, ad esempio il tipo di ordinamento oppure il comportamento dell'explorer nel mostrare una singola cartella. La cosa più interessante sono le ShellBag che si inizializzano quando l'utente entra almeno una volta all'interno di una cartella, scorrendole si può ricostruire una sorta di timeline precisa relativa all'utente e alla sua esplorazione del file system. Le ShellBag possono tornare utili in diverse situazioni e sono situate nel registro di sistema, più precisamente nelle chiavi dei sotto-alberi di **HKEY_USERS**.



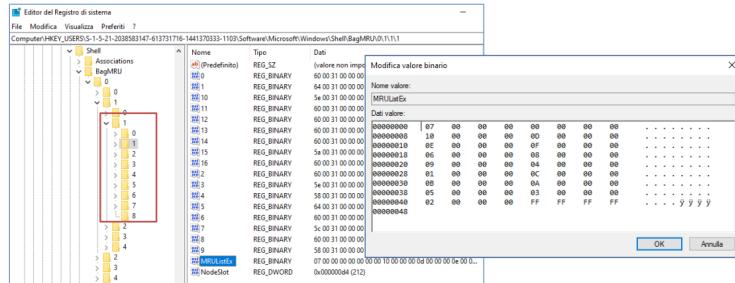
All'interno dei nodi di questo sotto-albero vi sono una serie di sottocartelle e quelle più importanti sono:

- **BAGMRU**: memorizza lo storico di tutte le cartelle visualizzate dall'utente durante l'uso delle macchine. Queste cartelle sono elencate tramite dei numeri progressivi.
- **BAGS**: contiene le impostazioni di visualizzazione delle cartelle che sono contenute in **BAGMRU**.

All'interno di **BAGMRU** troviamo il file **MRULISTEX** questo rappresenta quella che è una traccia temporale che ci permette di determinare l'ordine con cui l'utente ha navigato attraverso il file system, così da ricostruire le azioni dell'utente anche se per caso le cartelle non dovessero più esistere. Come si nota i valori di **MRULISTEX** sono espressi in gruppi di 4 byte, la prima sequenza è la **01** poi abbiamo **03, 07, 05** e così via, se seguiamo l'ordine di lettura di questi valori in **BAGMRU** otteniamo l'ordine che riguarda la lettura delle ultime cartelle visionate dall'utente, in ordine decrescente. Come seguire le tracce nella ShellBag, leggiamo il valore contenuto in **mrulistex** e poi ci riporta nella cartella specifica in **BAGMRO**, seguendo l'ordine della prima sequenza ci posizioniamo nella cartella **01**.



Selezionando la cartella **1** vediamo il nome, in questo caso è **Procura Napoli**. Volendo poi vedere i file a cui ha avuto accesso l'utente in quella cartella ci basta identificare la cartella che si chiama **1** nel sotto-albero e poi leggere nuovamente il file **MRULISTEX** che ci identifica i file e sottocartelle visualizzati.

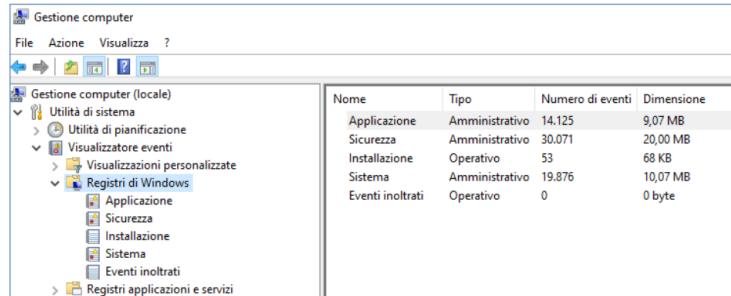


I dati estratti all'interno di **BAGMRU** sono:

- **BagNumber**: rappresenta la sottochiave bags che contiene le preferenze dell'utente.
 - **Registry Key Last Write Time**: ovvero o la data di primo accesso o di ultima modifica di una specifica cartella.
 - **Folder Name**: il nome della cartella.

18.2.8 Event Viewer

Questo è il sistema standard per il logging su Windows, esso produce una serie di file con estensione **EVT/EVTX** e possono essere utilizzati tramite il software con medesimo nome che troviamo nel sistema. La maggior parte degli eventi viene associata con un codice errore.



Codici Eventi.

ID Evento ≥ Vista	ID Evento < Vista	Descrizione
1102	517	Log di audit cancellato
4624	528/540	Accesso di un account completato
4625	529/537	Accesso non riuscito per un account
4634	538	Un account è stato disconnesso
4674	578	Operazione eseguita con privilegi elevati
4704	608	Assegnazione di un diritto per un utente
4719	612	Cambiamento nelle politiche di audit
4720	624	Aggiunta di un nuovo account
4722	626	Un account utente è stato abilitato
4726	630	Un account utente è stato eliminato
4732	636	Un account utente è stato aggiunto ad un gruppo locale
4738	642	Un account utente è stato modificato
4739	643	Cambiamento nelle policy di dominio.

18.2.9 Application Data

In windows per ciascun utente configurato c'è un posto dove vengono conservate alcune delle impostazioni dei programmi utilizzati dall'utente, ma anche alcuni file temporanei riguardo l'uso di questi programmi.

► Windows XP:

- \Documents and Settings\[nome_utente]\
 - Dati Applicazioni
 - Impostazioni Locali

► Windows ≥ Vista:

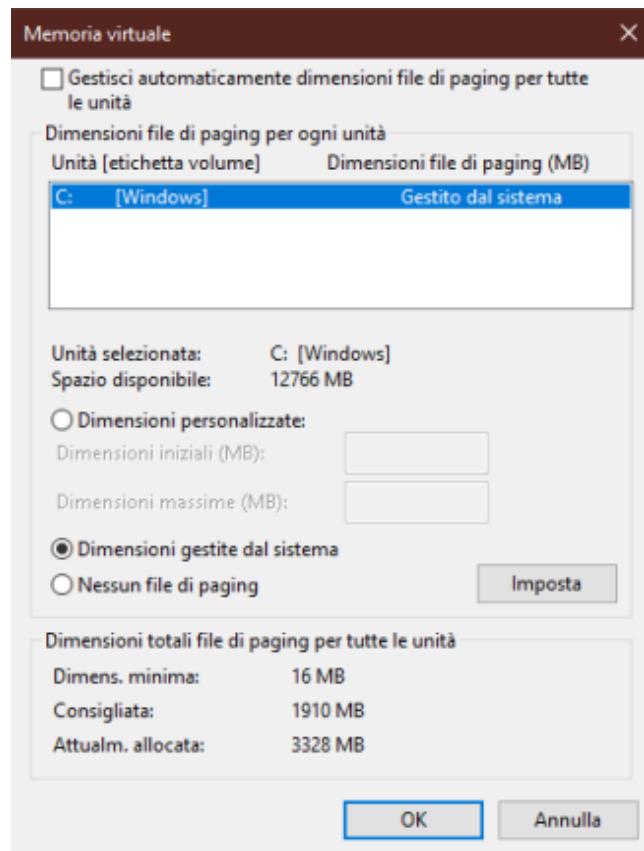
- \Users\[nome_utente]\AppData

In queste cartelle ritroviamo molti dati utili da analizzare, come:

- Posta Elettronica.
- Cache.
- Cronologia.
- Log.
- Configurazioni.

18.2.10 File Swap

È un'estensione di quella che è la memoria volatile (RAM), questa viene "eseguita" riservando parte del disco per la memorizzazione di alcuni dati che andrebbero in RAM. Gestendo Microsoft tutto attraverso i file, anche per lo swap viene impiegato un file, questo viene chiamato **Pagefile.sys** e di solito si trova sul disco di sistema. Il file di swap contiene porzioni della RAM, quindi qualora il digital forenses volesse eseguire un dump della RAM deve ricordare anche questo file.



Un ulteriore file alquanto utile è **Hyperfil.sys** che è generato in automatico dal sistema e contiene una copia della memoria RAM che viene prodotta ogni qualvolta il sistema viene messo in uno stato di sospensione/ibernazione.

18.2.11 Pro vs. Contro di Windows

- **Vantaggi**

- è diffuso.
- è molto documentato.
- essendo usato da moltissime persone è quasi impossibile non trovare supporto per una qualche categoria.

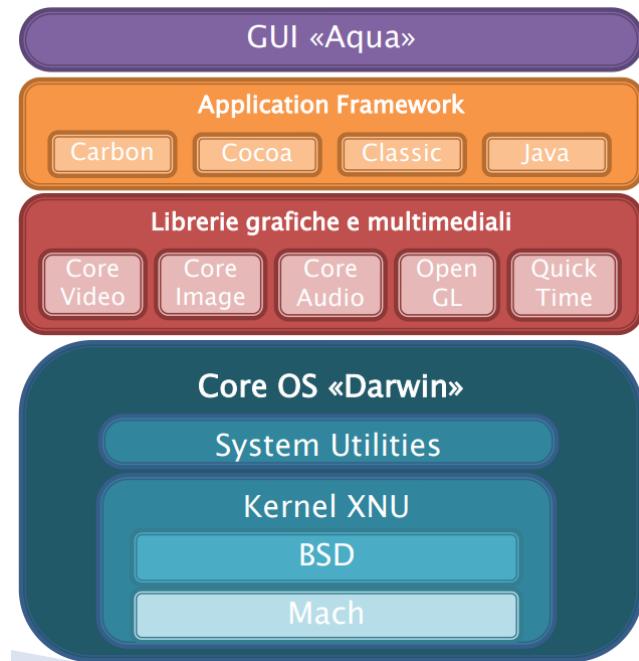
- **Svantaggi**

- pochi log, il numero di eventi che scatena un log è basso.
- presenza di antivirus che possono compromettere le info temporali.
- è un sistema commerciale.

18.3 Apple OSx/MacOS

18.3.1 Overview

Il sistema operativo **OSx** è formato da due strati distinti, uno è il sistema operativo vero e proprio, che è sviluppato da Apple ed è basato su una licenza opensource, ed è composto da un kernel ibrido chiamato **Kernel XNU**, è definito ibrido perché è l'unione di un **Kernel Mononitico di free BSD** e di un codice derivato da un microkernel chiamato **MACH 3.0**. Sopra questo kernel viene hostato un sistema di derivazione BSD che si chiama **free BSD** e si chiama **Darwin**, al di sopra di questo troviamo poi lo strato commercial, ovvero quello non opensource, e al suo interno troviamo tutte le **Librerie Grafiche e Multimediali** ed inoltre troviamo gli **Application Framework**. Leggermente più sopra c'è il livello utente che comprende tutto ciò che riguarda l'interfaccia grafica, che si chiama **Aqua**, quindi il sistema di gestione widget, la dashboard ed i sistemi di indicizzazione come spotlight.



18.3.2 Configurazione

Per quanto riguarda le impostazioni di configurazione, MacOS, fa uso di un database ad oggetti chiamato **NetInfo**, l'utente può usare il sistema ignaro delle configurazioni su questo DB. Esso è gestito da un app chiamata **Gestione NetInfo**. **NetInfo** controlla una serie di conigurazioni che sono puramente del sistema operativo e al contrario del registro di sistema Windows esso non gestisce alcuna configurazione a livello applicativo.

18.3.3 Configurazione Server

Dalla versione OSx Server 10.4 è stato introdotto un nuovo sistema di directory service che si chiama **Open Directory** il suo scopo è stato quello di integrare sotto un unico cappello tutte le funzioni principali di sistemi di directory service che erano presenti sul mercato, tra i quali NetInfo, in questo modo Apple ha fatto in modo che l'utente è in grado di interrogare tutti i diversi servizi di directory con gli stessi comandi di **Open Directory**. I tool di gestione server permettono quindi di manipolare il sistema Open Directory in modo agevole attraverso la G.U.I. I comandi messi a disposizione da OSx per la gestione dei servizi Open Directory sono:

Tool	Descrizione
dscl	Manipolazione e gestione dei servizi di directory
dsconfigldap	manipolazione degli alberi LDAP
dsconfigad	manipolazione dei sistemi Active Directory
dseditgroup	gestione di gruppi di utenti
dsenableroot	abilita/disabilita l'utente root in OpenDirectory
dscacheutil	regola le cache relative a OpenDirectory
dsmemberutil	Gestisce i gruppi di appartenenza di un oggetto OpenDirectory
dsexport	esporta oggetti da un albero OpenDirectory
dsimport	importa oggetti in un albero OpenDirectory

18.3.4 Cifratura

Attraverso il pannello di controllo di OSx è possibile accedere alla maggior parte delle funzioni utilizzate dal sistema operativo per mettere in sicurezza il sistema, una di queste è rappresentata dal **FileVault** che è la prima delle funzioni di sicurezza utilizzate da Apple, attraverso la quale la Home Directory dell'utente, ovvero **/Users/[nome_utente]**, viene cifrata con la password di LogIn. Con la versione OSx 10.5 l'archiviazione del FileVault è dipendente dal profilo dell'utente, questo significa che all'interno dello stesso sistema possono essere presentati sia utenti tradizionali, sia utenti con le Home Directory cifrate. Dalla versione 10.7 in poi, invece, fu introdotta la versione 2 di questo file, ovvero **FileVault2** e di comune alla precedente versione c'è solo il nome in quanto viene utilizzata la **Full Disk Encryption**.

18.3.5 File Swap

Come per Windows, anche OSx usa una serie di file per la gestione del meccanismo di **Paging** non mediante una partizione separata ma bensì grazie ad un file che contiene l'immagine della memoria per la funzione di *"sleep"* e sono conservati nella directory **/var/vm**. I file si swap sono un ottima fonte di informazioni.

18.3.6 Portachiavi

Un'altra funzione utile da analizzare è quella del **Portachiavi**, ovvero, ogni applicazione che necessita del chaching delle credenziali non fa uso delle proprie funzioni bensì ne utilizza una fornita dal sistema operativo stesso. Medianet delle API le credenziali sono salvate in uno specifico file cifrato con **AES-128** e salvato nella home directory dell'utente. Attraverso un'applicazione gestore portachiavi è possibile visionare il contenuto del file. Dalla versione 10.9 di OSx c'è stata una riscrittura della gestione del portachiavi che è in grado di funzionare anche con **iCloud**.

18.3.7 Analisi

Dato l'elevato numero di tecnologie proprietarie risulta difficile o incompleto per un digital forenses analizzare in modo corretto un sistema Apple. Il consiglio è di usare sempre sistemi Apple per l'analisi di loro stessi, ci sono software specifici per Apple e che permettono l'analisi di immagini forensi Apple, tra cui **BlackBag** che è una suite che adesso si chiama **BlackLight** ed inoltre in questa suite possiamo trovare un software di acquisizione forense che è **MacQuisition**.

Un altro software è **Mac Forensics Lab**, esso permette l'analisi sia di sistemi Mac OS che Windows. Esistono anche tool da riga di comando, come **Apple HDIutil**, tool fornito dalla stessa Apple su Mac OSx e permette la gestione delle immagini **DMG**, si possono eseguire copie logiche o copie full disk. Tutti i dati utente in un sistema Mac risiedono nella home directory dei singoli utenti ed anche tutte le applicazioni ce necessitano di salvare dati personali lo fanno all'interno della sottocartella **Library** della home directory.
[/Users/\[nome_utente\]/Library](#)

18.4 Linux

18.4.1 Overview

Linux è il nome del kernel che in realtà è in continuo sviluppo ed evoluzione **GNU/Linux**. Esistono decine di sistemi basati su questo kernel e queste sono dette **distribuzioni**, molte **distro** famose col tempo hanno cambiato approccio, ad esempio RedHat lanciò un progetto chiamato **Fedora** che è in realtà un laboratorio da cui si ricavano versioni desktop e server, poi c'è **Ubuntu** che possiede una grande quota del mercato. Per portare un minimo di standardizzazione nel mondo linux, nasce un progetto chiamato **Linux Standard Base (LSB)** che aveva lo scopo di fissare una serie di specifiche sulle quali poi basare lo sviluppo delle diverse distro linux, in questo modo si riusciva a garantire una certa omogeneità. Il sistema **GNU/Linux** puro non esiste, esso è la somma di alcune componenti come:

- **Kernel**: ogni sistema **GNU/Linux** contiene almeno una release del kernel linux. Questo kernel può essere di tipo vanilla, ovvero scaricato direttamente dall'repository ufficiale oppure può avere delle patch ad-hoc che migliorano o aggiungono funzionalità.
- **Librerie di Sistema**: il kernel è affiancato da una serie di librerie per le funzionalità di base.
- **Tool di base**: comandi fondamentali come tool di manipolazione del file system, gestione della rete, tool di autenticazione, shell di sistema, ecc.

► Distribuzioni commerciali:

- Red Hat Enterprise
 - Fedora
 - CentOS: versione libera senza supporto
 - Scientific Linux
- SUSE Linux Enterprise
 - openSUSE

► Distribuzioni gratuite:

- Debian: distribuzione ufficiale della Free Software Foundation
- Ubuntu



18.4.2 Sistema

Essendo un sistema nato tra gli anni 60/70 ha avuto ben 40/50 anni di lento debugging, questo lo ha reso altamente stabile, infatti Unix nasce come sistema **Multutente e Multitasking** infatti per la maggior parte tutte le distro permettono l'utilizzo da parte di utenti differenti di una o più shell fisiche. La struttura del file system è **rigida** ovvero che i file non possono essere posizionati a casaccio nel disco ma bensì ogni cosa ha il suo posto. Le directory in cui principalmente ricadono i dati sono le seguenti:

Directory	Contenuto
/bin	Binari d'uso comune nel sistema.
/boot	Kernel e file necessari al boot
/dev	device fisici e logici collegati al computer
/etc	File di configurazione del sistema
/home	File degli utenti
/lib	Librerie di sistema
/mnt	Punto di montaggio per media esterni
/opt	Punto dove sono installati programmi che richiedono complesse alberature per il loro funzionamento
/root	Home directory dell'utente root
Directory	Contenuto
/sbin	Binari riservati all'uso di root
/srv	File di dati per alcuni servizi server come web e server FTP
/tmp	Locazione generale per i file temporanei
/usr	Contiene programmi non indispensabili al sistema
/usr/local	Locazione per i programmi compilati dagli utenti
/usr/src	Sorgenti del kernel e dei vari pacchetti
/var	Parte variabile dei programmi. Contiene log, mail, spool di stampa, database e quanto può essere utile a un programma da tenere in una directory scrivibile
Device /dev	Contenuto
/hda	Disco ATA master collegato al canale primario
/hdd	Disco ATA slave collegato al canale secondario
/sda	Disco SCSI con l'ID più basso collegato alla catena
/hda1	Prima partizione del disco ATA master collegato al canale primario
/loop0	Loop device. Permette visualizzare un file immagine come se fosse realmente agganciato
/eth0	Prima scheda di rete collegata al sistema
/md0	RAID software generato da Linux

Linux, come ogni sua distro, ha un sistema semplice per i permessi che funziona su un qualunque file system nativo, questo utilizza 3 gruppi di lettere che sono:

- **R**: permesso di lettura.
- **W**: permesso di scrittura.
- **X**: permesso di esecuzione / permesso di accesso se si parla di una directory.

r	w	x	r	w	x	r	w	x
owner			group			public		

Sono in totale 9 lettere e vengono suddivise in questo modo. (L'utente **root** non ha di questi limiti)

18.4.3 Log

L'analisi di un sistema GNU/Linux può essere una sfida in quanto, da un lato il sistema fornisce una marea di informazioni, in più, rispetto a Windows o Mac OS, esistono decine di log differenti che ci regalano informazioni. Il sistema risulta più standardizzato rispetto all'anarchia di Windows, questo non ne preclude la libertà, l'amministratore di sistema infatti ha molta libertà, esso infatti può decidere di ricompilare il kernel qualora volesse, ed aggiungere patch ad-hoc, rendendo il tutto più complesso. L'analisi dovrebbe partire dagli elementi più specifici fino ad arrivare, nei casi peggiori, all'analisi dei sorgenti del kernel installato. Tutti i sistemi Unix utilizzano un sistema standard per la gestione dei **log**, questo sistema prende il nome di **syslog** e viene utilizzato dalla maggioranza dei software, di sistema o meno, per la scrittura dei log.

I sistemi più nuovi utilizzano **rsyslog** e **syslog_ng** in più integrano nuove funzionalità come la spedizione dei log tramite tunnel tcp/udp o il supporto ai database. **syslogd** è il daemon su cui si basa **syslog** ed ha il compito di effettuare il **data collecting** per l'intero sistema o per più sistemi qualora fosse configurato lato server, ed una volta ricevuto un messaggio di log il daemon segue le direttive riportare nel file di configurazione **/etc/syslog.conf** e decide in base a questo dove scrivere quelle informazioni. Le entru syslog sono gestite da due parametri con le quali il daemon decide cosa fare, questi parametri sono:

- **Facility**: indica il tipo di programma che sta scrivendo la entry.
- **Severity**: indica il grado di priorità della entry.

Facility code	Keyword	Description
0	kern	Kernel messages
1	user	User-level messages
2	mail	Mail system
3	daemon	System daemons
4	auth	Security/authentication messages
5	syslog	Messages generated internally by syslogd
6	lpr	Line printer subsystem
7	news	Network news subsystem
8	uucp	UUCP subsystem
9	cron	Clock daemon
10	authpriv	Security/authentication messages
11	ftp	FTP daemon
12	ntp	NTP subsystem
13	security	Log audit
14	console	Log alert
15	solaris-cron	Scheduling daemon
16–23	local0 – local7	Locally used facilities
Severity Value	Severity	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant conditions
6	Informational	Informational messages
7	Debug	Debug-level messages

Di norma i file di log si trovano in [/var/log](#) e il file più significativo è rappresentato dal file chiamato **messages** che contiene gli eventi relativi alla macchina. L'unico log nonrappresentato da un file di testo, nei sistemi Unix, è il file **WTMP** che contiene la registrazione degli accessi utilizzatori del sistema in formato binario. Come detto prima, non tutti i software fanno uso di syslog, alcuni processi agiscono per conto proprio per salvare i log, come per **Apace** o **Samba**, per essere sicuri di trovare tutti i log nel sistema si può utilizzare il programma **LogFinder**.

18.4.4 Configurazioni

I sistemi unix sono configurabili attraverso un unico programma, l'editor di testo. Tutte le configurazioni sono contenute nella directory `/etc` e sono rappresentate da una serie di file di testo. Grazie a questa struttura è possibile verificare lo stato di configurazione del sistema semplicemente usando dei programmi di ricerca come `grep` o `find`. Nella directory `/etc` sono contenuti i **file di configurazione di default**, alcuni dei quali sono:

- **inittab**: file di configurazione di boot.
- **passwd**: elenco degli utenti.
- **shadow**: password degli utenti.

Quando l'utente decide di modificare uno di questi comportamenti di default il programma salverà la particolare configurazione all'interno di un file ed una directory nascoste all'utente nella sua home directory.

Nota Bene: Linux può utilizzare anche esso un semplice file di appoggio per lo swap. Linux predilige che il file system dove sarà salvato questo file sia di tipo FAT oppure si trova un'apposita partizione di swap marcata con **0x83**.

18.4.5 Home Directory

Il concetto di **Home Directory** è più radicale in Linux rispetto agli altri sistemi. Banalmente è il posto dove l'utente ha il permesso di scrivere/leggere. In Unix esistono due tipologie di utenti:

- **root**: amministratore di sistema.
- **utente comune**.

C'è una differenza tra l'amministratore di un sistema Unix e quello di un sistema Windows, quest'ultimo non concede proprio tutti i permessi e quindi il libero accesso a tutto, come invece succede in Unix, il sistema da piena "fiducia" all'admin permettendogli di effettuare molte più operazioni.

18.4.6 /VAR Directory

Una delle parti più importanti durante l'analisi di un sistema Unix è la directory **/var**, questa racchiude tutti i dati che vengono modificati durante l'esecuzione del sistema, appunto "*Variabili*". Abbiamo dati come per esempio:

- log di sistema;
- spool di stampa;
- mail in transito e code;
- tablespace degli RDBM;
- cache di sistema;
- configurazione dei vari tool;
- database dei pacchetti installati;
- file di bind;
- database di LDAP;
- database di sistema di AFS;
- database di Kerberos.

Risulta essere necessario analizzare tutte le sottocartelle per avere un'analisi completa.

18.4.7 Analisi

Durante l'analisi di un sistema Unix le cartelle da analizzare per lo più sono:

- **/home**.
- **/var**.
- **/etc**.

Essendo Unix un sistema in costante aggiornamento, non sempre i tool di analisi forense riescono ad analizzare tecnologie nuove e magari ancora non supportate, d'altro canto, essend un sistema aperto sarà sicuramente possibile impiegare qualche distro Linux che dispone già di queste tecnologie per analizzarle. Per le **analisi live** ciò a cui dobbiamo essere attenti è:

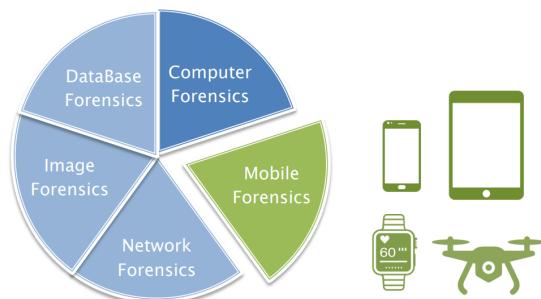
- **inittab/systemd**: controllare i servizi eseguiti in fase di boot.
- **autenticazione**: verificare la configurazione PAM, kerberos e openldap.
- **/etc/fstab**: verificare il montaggio dei file system all'avvio.

19 Lezione 20

19.1 Mobile Forensics

19.1.1 Overview

Le linee guida sono le stesse della Computer Forensics, ciò che cambia è l'oggetto d'indagine che è un dispositivo mobile in questo caso.



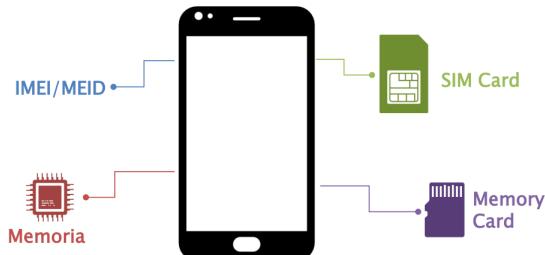
Come detto in precedenza, anche la **Mobile Forensics** è una branca della Digital Forensics. Questa categoria ricopre i dispositivi di telefonia mobile e non solo, ma anche smartwatch, droni, dispositivi IoT.

Le best practice da rispettare sono le medesime spiegate per la Computer Forensics e che sono alla base della Digital Forensics. Non sempre sarà però possibile accendere il dispositivo da analizzare e non sempre sarà possibile portare a casa una copia fisica della memoria. Quindi la copia forense andrebbe elaborata invocando l'accertamento tecnico di tipo irripetibile. [ART.360 C.P.P]

19.1.2 Evidence

Le informazioni presenti all'interno di un dispositivo mobile sono rappresentate da:

- **IMEI/MEID**: identificativo del dispositivo, è un codice che identifica il dispositivo nella rete **G.S.M/C.D.M.A.**
- **Memoria**: rappresenta la memoria interna del dispositivo.
- **Memory Card**: memoria esterna, può essere più grande rispetto alla memoria principale.
- **SIM Card**: rappresenta l'identificativo di un contratto di telefonia mobile ed ha una propria memoria.

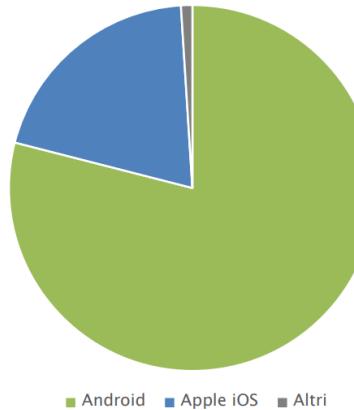


19.1.3 GSM/CDMA

I dispositivi mobili si distinguono per lo più in base alla tecnologia che adoperano per connettersi alla rete mobile, tra queste abbiamo:

- **G.S.M (GLobal System for Mobile communications)**: è il sistema più diffuso. In breve questa tecnologia suddivide la banda di comunicazione in frazioni temporali assegnate a ciascun utente, presuppone l'uso di una **SIM Card** per identificare il dispositivo nella rete mobile. Anche le **SIM Card** sono identificate da un numero univoco chiamato **ICCID** composto da 19/20 cifre, e da un altro codice detto **IMSI** che serve ad identificare la SIM Card nella rete mobile interna. I dispositivi con tecnologia G.S.M possiedono un **IMEI**.
- **C.D.M.A (Code Division Multiple Access)**: a differenza del G.S.M questa tecnologia offre all'utente tutto lo spettro della banda e per poter funzionare non è vincolato all'utilizzo della SIM Card. Fa uso di un identificatore univoco chiamato **MEID** che serve a riconoscere il dispositivo all'interno della rete mobile.

Nota Bene: per quanto riguarda gli smartphone bisogna fare una distinzione anche in base alla tipologia di sistema operativo.



19.1.4 La Raccolta

Nella fase di raccolta del reperto di interesse, quindi durante perquisizione e sequestro, bisogna essere consapevoli che i dispositivi mobili sono nati per essere sempre connessi, quindi la prima cosa di cui preoccuparsi è isolare il dispositivo dalla rete, utilizzando ad esempio le funzionalità messe a disposizione dal dispositivo stesso, ovvero la **Modalità Aereo**, sincerandosi che con essa vengano spente tutte le periferiche di rete, quali Bluetooth, Wi-Fi, GPS, NFC. Esistono delle sorta di borse che creano quella che è chiamata **Gabbia di Faraday** che è un sistema in grado di isolare il dispositivo al suo interno. I motivi dell'isolamento sono principalmente due, ovvero:

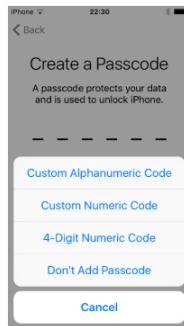
- Tutela del dispositivo da possibili operazioni di **Remote Wipe**.
- Evitare la ricezione di nuovi elementi, come chiamate, SMS, Chat, ecc; In modo da non alterare il sistema senza la presenza del proprietario.

L'utente può decidere o meno di attivare alcune funzioni di sicurezza sul dispositivo, queste sono distinte in base al sistema operativo. Ad esempio per Apple iOS si può bloccare l'accesso al dispositivo tramite:

- Passcode a 4 cifre.
- Passcode a 6 cifre (default).
- Passcode > 6 cifre.

- Password Alfanumerica.
- Face ID/ Touch ID.

Ognuno con un massimo di 10 tentativi.



Per Android OS invece:

- Passcode \geq 4 cifre.
- Password Alfanumerica.
- Pattern.
- Face ID/Touch ID.
- Password di Avvio.

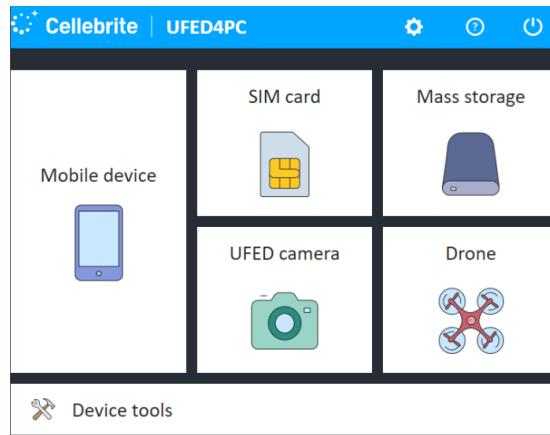
Massimo ? tentativi.

Inoltre è possibile ci siano ulteriori livelli di sicurezza sul dispositivo, protezioni implementate dalle app installate sul dispositivo, come WhatsApp o Telegram con codici di accesso, oppure sono presenti app di sicurezza che bloccano l'accesso ad alcuni dati oppure l'avvio di alcune app. Anche le SIM Card hanno delle protezioni, esse sono coperte da una passcode a 4 cifre detta PIN e si hanno a disposizione 3 tentativi alché verrà richiesto il recovery code per la SIM detto PUK che è composto da 8 cifre e permette un massimo di 10 tentativi. Dopo aver verificato queste possibilità di sicurezza si può procedere allo spegnimento del dispositivo. Alcuni dispositivi richiedono lo sblocco per essere spenti, c'è però la possibilità di bypassare il codice di sblocco ed eseguire il dispositivo in una modalità detta **AFU (After First Unlock)** ed effettuare così l'acquisizione.

19.1.5 Acquisizione - Strumenti

Per eseguire la copia forense esistono in realtà sul mercato diversi strumenti, sia hardware che software. **Cellebrite UFED** è il software su cui faremo affidamento.

Esso è composto da una parte software.



E da una parte hardware, comprensiva di cavi di vario genere per collegare le diverse porte dei dispositivi. Alcuni di questi cavi sono programmabili tramite UFED Adapter.



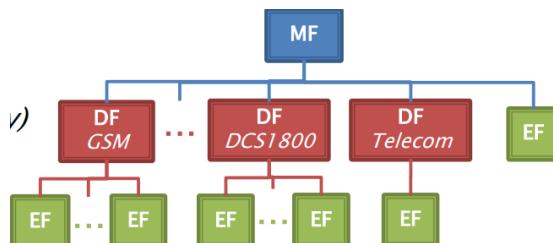
19.1.6 Acquisizione - Memory Card

Nei dispositivi mobili potrebbe esserci la presenza di una memory card, il suo scopo è quello di espandere la memoria del dispositivo. La prima cosa da fare quando c'è in ballo una memory card è effettuare una copia forense prima di procedere a quella del dispositivo stesso, questo perché non sappiamo se nella memory card ci sono oppure no dei file utili al funzionamento del device, allora è buona premura effettuare una copia a priori. La copia di una memory card viene effettuata come di prassi o con una distro forensics oriented o con un software come FTK Imager/WriteBlock Hardware.

19.1.7 Acquisizione - SIM Card

Questa non è altro che una smart card sulla quale gira un'app chiamata SIM e che viene impiegata per le reti GSM. Il file system di una SIM è organizzato in una struttura gerarchica ad albero in cui sono presenti:

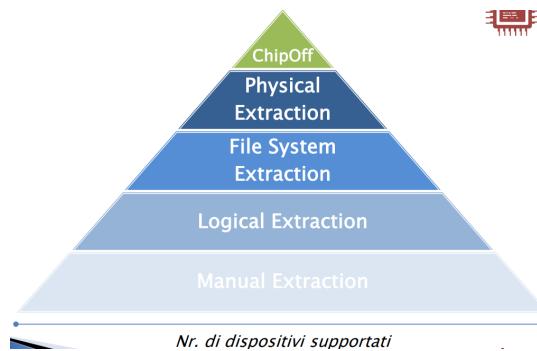
- **MF**: un **Master File** che rappresenta la root dl file system che può contenere uno o più dedicated file o altri elementary file.
- **DF**: i **Dedicated File** si possonov edere come delle directory in quanto possono contenere sia altri dedicated file ed anche elementary file.
- **EF**: gli elementary file sono file che contengono i vari tipi di formati di dati presenti nella SIM.



L'estrazione dei dati da una SIM può avvenire tramite un lettore di smart card ed un software che interopera con la SIM e ne estrae i dati, l'estrazione può avvenire sia a livello file system e quindi viene creata una copia del file system oppure può essere logica, quindi lo strumento si interfaccia con la SIM Card e mediante delle API richiede i dati di cui necessita, lo strumento che poi riceve questi dati li impacchetta in una propria struttura dati, ad esempio un XML, in modo che poi possa essere visionabile dal computer forensi.

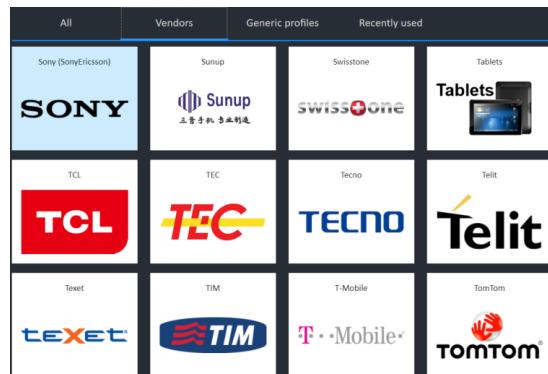
19.1.8 Tipologie di Acquisizione

Le metodologie di estrazione dei dati che possono essere fatte su un dispositivo sono cinque:

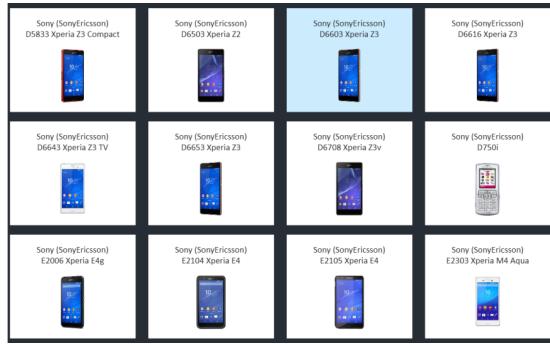


Esse sono disposte in ordine di numero massimo di dispositivi supportati. Il **ChipOff** supporta un numero ristretto di device ed è per lo più delle volte distruttivo in quanto viene rimosso proprio il chip dal dispositivo. L'estrazione manuale comprende il maggior numero di dispositivi.

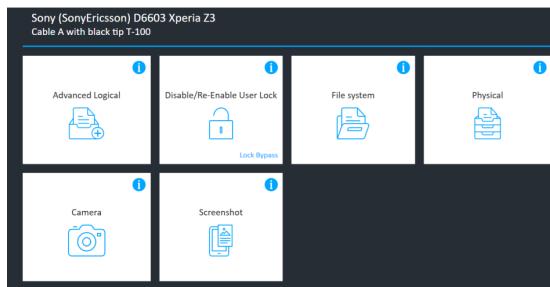
19.1.9 Cellebrite UFED - GUI



Richiesta al forensen del tipo di dispositivo da acquisire in base al suo produttore.



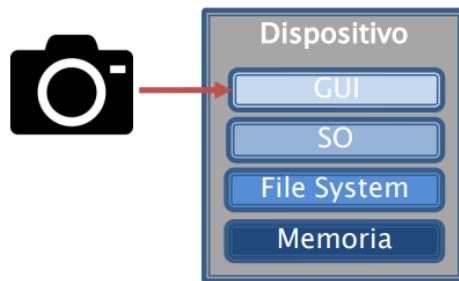
Viene poi richiesto il modello del device.



Ed infine viene chiesta la tipologia di acquisizione che si desidera effettuare.

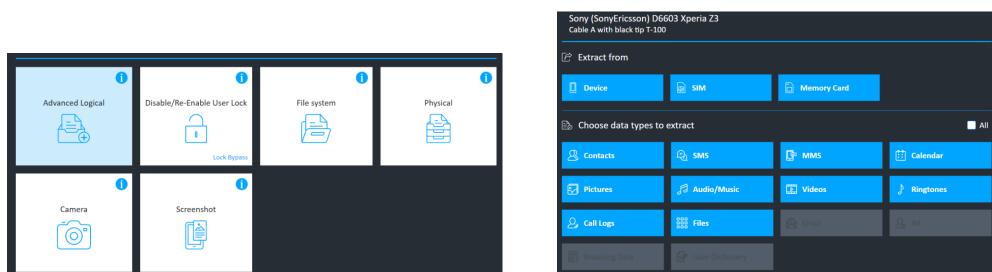
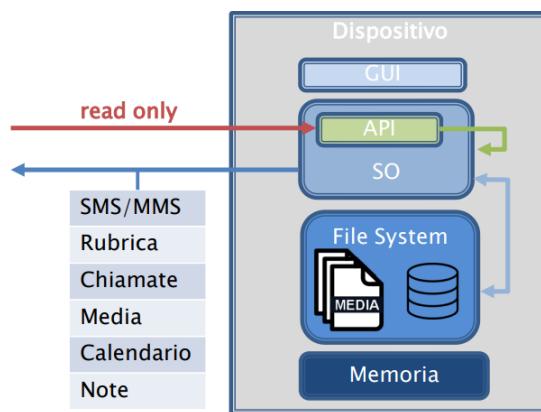
19.1.10 Manual Extraction

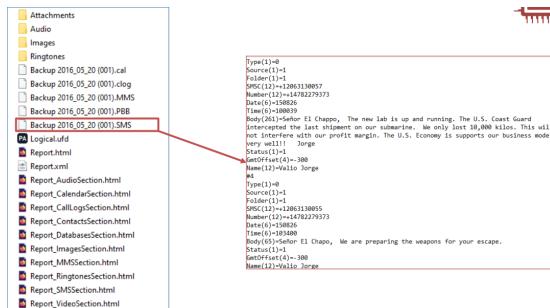
Questo metodo di estrazione prevede di visualizzare i dati presenti nel dispositivo tramite la medesima interfaccia dell'utente ed eseguire la così detta **repertazione fotografica** che consiste nello scattare foto o realizzare video dei dati di interesse presenti. Questo metodo può essere supportato da software di catalogazione foto e che aiuta il forense a separare le varie tipologie di dati che si sta acquisendo. Gli svantaggi di questo metodo possono essere diversi, ad esempio il processo può essere molto lungo se ci sono molti dati da estrarre, poi c'è il rischio di poter cancellare i dati di interesse, inoltre le info repertabili sono solo quelle mostrate dall'interfaccia. Ovviamente ci sono dei limiti all'applicabilità della procedura, ad esempio i dispositivi con i display non funzionanti ai quali non possono essere fatte foto, oppure dispositivi dei quali non si conosce il codice di sblocco, questi sono dispositivi non elegibili per questo tipo di procedura di acquisizione.



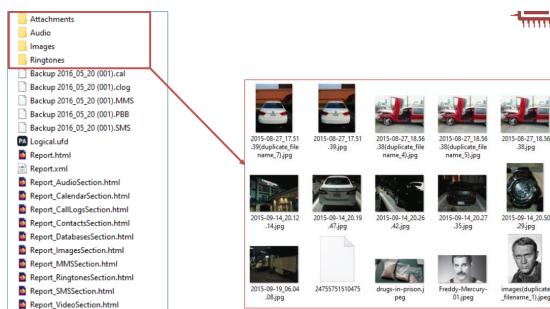
19.1.11 Logical Extraction

Questa metodologia prevede la connessione tra il dispositivo target e la workstation che deve eseguire l'estrazione e questa connessione può avvenire con un cavo USB, un cavo seriale o addirittura su alcuni device è possibile usare il Wi-Fi, il Bluetooth o gli infrarossi come tramite. Gli strumenti che eseguono questa estrazione fanno uso delle API che sono messe a disposizione dal sistema operativo del dispositivo. La maggior parte dei dati estratti con la logical extraction sono il risultato dell'impiego delle API del produttore del dispositivo, lo strumento forense effettua delle chiamate API, ad esempio, di sola lettura e il dispositivo risponde all'API estraendo i dati richiesti in maniera leggibile e lo strumento forense si farà carico di mostare i dati. I dati che vengono estratti sono limitati a quelli messi a disposizione dal dispositivo tramite le API, a volte questi possono essere incompleti, oppure potrebbero non essere implementate le API per un contenuto del dispositivo. Per eseguire questa estrazione si deve avere accesso al dispositivo.





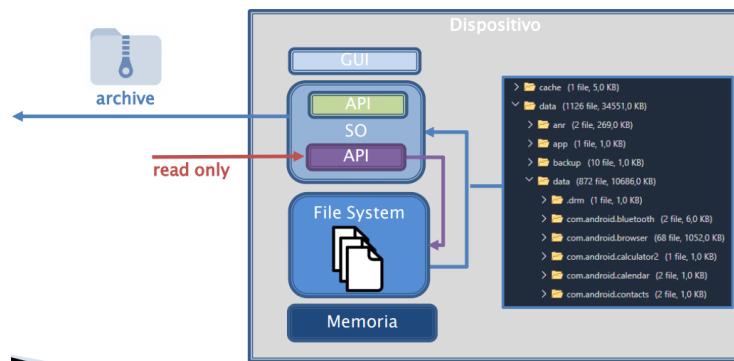
I dati passati dal dispositivo vengono collezionati e catalogati in specifici file.

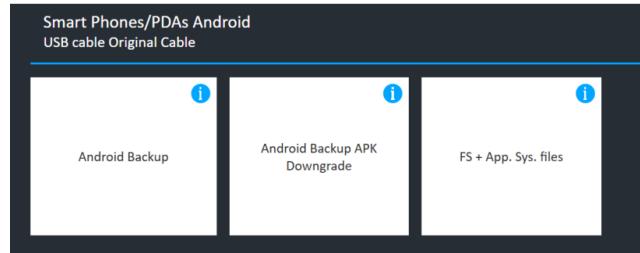


Nelle cartelle troviamo i file media estratti dal dispositivo tramite le API.

19.1.12 File System Extraction

Questo metodo di estrazione rientrerebbe nella estrazioni logiche, in questo caso lo strumento forense richiede al sistema operativo i file e non i dati presenti. Lo strumento fa uso di API differenti da quelle usate per la logical extraction e sono API con livello di amministratore nel dispositivo, possono variare in base alla versione del sistema operativo. Lo strumento forense chiede al S.O. la struttura delle cartelle proprio come se fosse un backup ed il S.O. invia questi file che potrebbero essere sia solo quelli di accesso all'utente oppure anche i file del S.O. stesso. Rispetto alla logical extraction non si ricevono i dati in formato leggibile bensì dei dati che andrebbero analizzati e parsati e interpretati, ad esempio invece di un singolo file con gli SMS in questo caso avremo tutto il DB degli SMS e questo va analizzato e vanno estratti gli artefatti. Visto l'utilizzo dei database è possibile ottenere degli artefatti cancellati ma ancora presenti nel database. I risultati che si ottengono con questa estrazione possono essere differenti a seconda dei permessi con cui vengono fatte le richieste, il risultato migliore è il **File System Completo** dove si ottengono tutti i file e cartelle della Live Partition, poi abbiamo **File System Parziale** dove vengono considerate solo alcune porzioni.

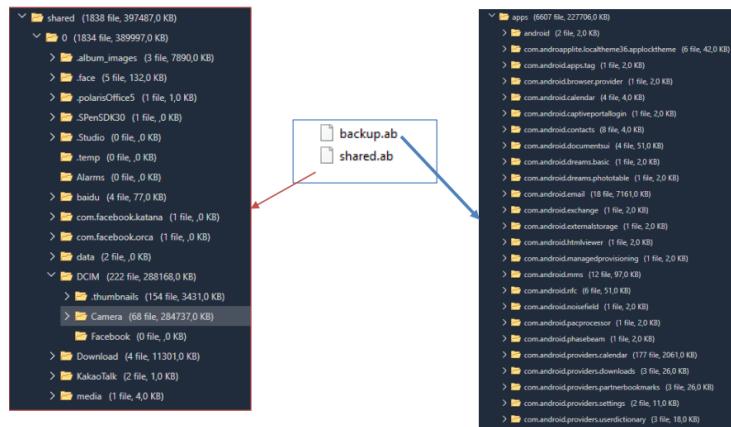




Smart Phones/PDAs Android
USB cable Original Cable

Connect the source device to the USB port on the computer. If the device is already connected, disconnect and then reconnect the device.

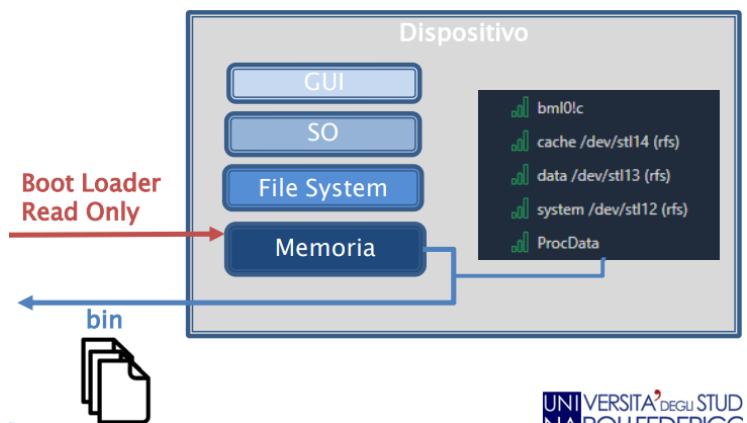
Android:
Important:
Verify that the device's Internet connectivity is disabled (Wi-Fi and mobile data) by entering into Airplane mode.
This method is supported for devices running Android version 4.1 and above and with Developer options enabled.
To enable the Developer options, go to Menu → Settings → About (information) → tap the "Build number" 7 times until it's enabled.
Under Developer options → enable the Android/USB debugging and Stay awake (if available).
Notice:
After pressing "Continue" the extraction will start automatically, DO NOT press anything.
If the extraction does not start you will be prompted to select "Back up my data" on the device.
Note:
On some devices the "Back up my data" button may be disabled.
To enable it, enter a password and then press the "Back up my data" button.
To decode the extraction, this password will also be required in Physical Analyzer.
On some devices, the "Back up my data" button is not clearly visible, and you may need to press the bottom-right corner of the device's screen to continue.



Il risultato di questa estrazione sono due archivi che contengono le strutture delle cartelle così come sono nel dispositivo Android che è oggetto di estrazione.

19.1.13 Physical Extraction

Questo metodo risulta più performante in termini di estrazione dei dati dal dispositivo, esso è paragonabile alla copia forense (copia bit a bit). I risultati ottenibili di solito dagli strumenti forensi che eseguono questa procedura sono dei file binari (RAW) e vengono etichettati con estensione **.bin**, questi file possono essere anche più di uno per singolo dispositivo acquisito magari se il device ha più supporti di memoria oppure se ci sono più partizioni. Con questa metodologia prendiamo l'intera memoria così com'è e quindi prendiamo anche lo spazio non allocato e quindi di conseguenza anche i file cancellati.

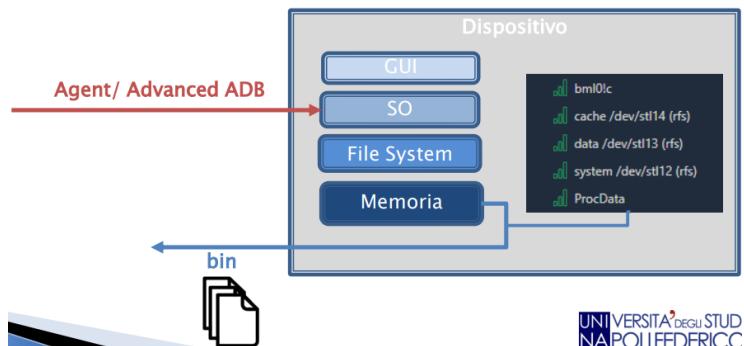


Una prima metodologia che permette di avere un'estrazione fisica della memoria del dispositivo fa uso del così detto **BootLoader**, ovvero si manda il dispositivo in una modalità detta recovery in cui è possibile inserire delle piccole parti di codice, chiamate appunto **BootLoader**, all'interno della RAM al momento in cui deve essere avviato il dispositivo. Questa metodologia è rischiosa se non si conosce il BootLoader che si sta andando ad usare.

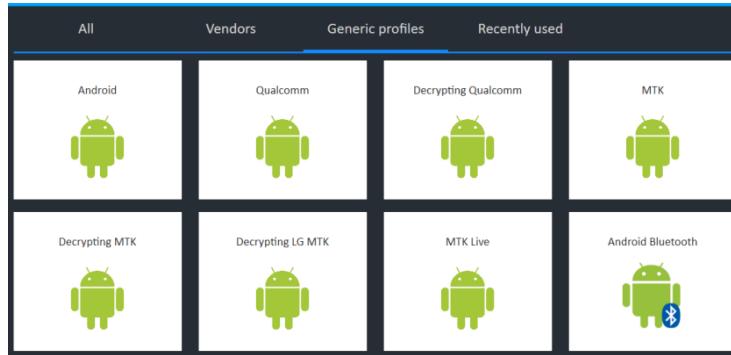
Errori in questa fase potrebbero compromettere il dispositivo. Questa modalità necessita di alcuni pre-settaggi del dispositivo, come ad esempio **USB Debug Mode**, e si deve essere a conoscenza delle impostazioni di sicurezza, se non c'è bisogno di determinati settaggi allora permettere al forense di bypassare i criteri di blocco del dispositivo ed eseguire una copia anche dei dispositivi con schermo rotto. Dato che i dispositivi più moderni garantiscono la cifratura della memoria, in questo caso non si riuscirebbe neanche a caricare il BootLoader in quanto il Boot Volume è cifrato; In realtà alcuni studi hanno riportato alcuni bug su alcuni chipset e che hanno permesso di inserire ed eseguire i così detti **Decrypted BootLoader** che riescono a cambiare l'avvio e l'esecuzione del S.O. appena prima

questo venga caricato ed appena dopo il rilascio della MasterKey in modo poi da avere una visione in chiaro della memoria.

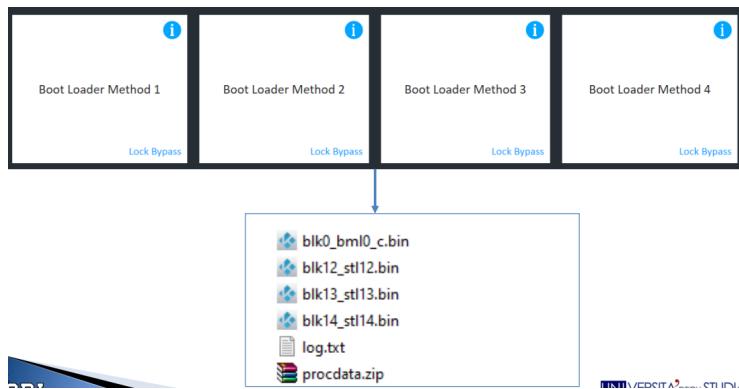
Esiste anche un altro metodo di estrazione quando non si può usare il BootLoader, viene in nostro soccorso un metodo che prevede installazione di un pezzo di codice sul sistema operativo sfruttando la presenza di alcuni bug; Questo codice si chiama **Agent**, questa metodologia viene utilizzata anche per la **Full File System Extraction** per effettuare un dump del file system. Questa metodologia comporta una manipolazione dell'evidence, quindi è buona norma documentare l'installazione dell'agent e cosa esso esegue. Questo modoto prevede che il dispositivo sia bloccato o almeno in modalità **AFU**. Per i dispositivi Android esiste un altro metodo per l'estrazione fisica, per lo più su vecchi device, ed è la **Advanced AndroidDebugBridge (ADB)**, è possibile eseguirla su Android fino alla versione 7.1 e con patch di sicurezza fino al 11/2016. Anche questa estrazione viene eseguita con l'ausilio di un agent e prevede di usare come destinazione del dump della memoria fisica o una memory card da inserire nel dispositivo target oppure la possibilità di estrarre i dati su una memoria estrena collegabile mediante cavo UTC.



L'output di questo tipo di estrazione va processato ed elaborato per visualizzare i dati presenti. A differenza della Full File System Extraction, il cui processo prevede l'analisi dei dati presenti nei diversi database, l'estrazione fisica esegue una copia per interno della memoria e quindi bisogna prima preoccuparsi di leggere il sistema di partizionamento e poi leggere il file system presente, da qui poi si procede in modo simile alla file system extraction a differenza che con l'estrazione fisica si può eseguire anche il recupero dei file cancellati (**Carving**, la physical extraction raggruppa un minor numero di device sul quale eseguirla poiché devono rientrare in una casistica più ristretta. Lo strumento UFED ci permette di selezionare anche il chipset che abbiamo all'interno del dispositivo.



Inoltre vengono fornite diverse tipologie di estrazione mediante BootLoader, ognuno con una diversa vulnerabilità.



19.1.14 ChipOff

Un altro metodo di estrazione molto invasivo è quello del **ChipOff** che prevede l'estrazione fisica dissaldando il chip di memoria dalla scheda madre del dispositivo per poter poi essere letta attraverso degli strumenti hardware particolari. Questa è considerata una tecnica estrema in quanto comporta la quasi distruzione del supporto e può essere quindi effettuata una sola volta, quindi solo nel caso non ci fossero ulteriori soluzioni all'estrazione dei dati dal device.

19.2 Analisi - I Sistemi Operativi

19.2.1 Android

Come sappiamo Android è un sistema basato su Kernel Linux che è stato sviluppato per l'uso su dispositivi mobili. La natura del suo kernel open source ha permesso la collaborazione di diversi sviluppatori e che in pochi anni hanno portato ad una rapida implementazione di nuove innovazioni, anche il market store delle app, che deriva dal repository utilizzato nel mondo Linux per scaricare applicativi e tenerli aggiornati, è una derivazione delle caratteristiche open source. Le app sono il punto forte di Android e molte società spendono molte risorse per la loro creazione e il loro mantenimento e le app non sono presenti esclusivamente sullo store ufficiale ma l'utente può scaricarle anche tramite store alternativi o dal web.

19.2.2 Apple

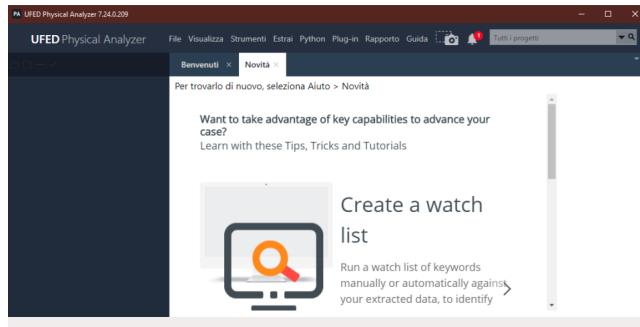
Nonostante la sua natura closed, Apple iOS ha raggiunto una buona fetta di mercato. Il mondo Apple OS è sorretto da un proprio market store per le app, detto **AppStore**, ma in realtà dalle ultime direttive della comunità europea, Apple dovrebbe garantire l'utilizzo di app scaricare da store di terze parti.

19.3 Analisi - App

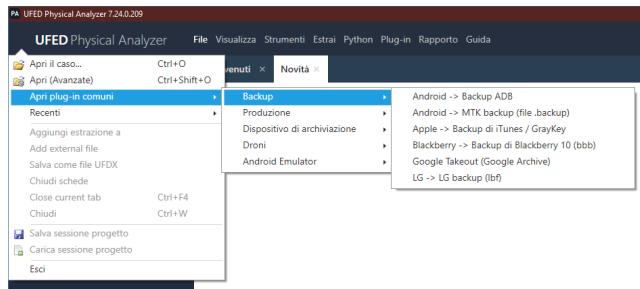
La cosa che sicuramente hanno in comune i due sistemi operativi è l'estensione delle loro funzionalità tramite le app, soprattutto le app di terze parti. Possiamo quindi capire che l'analisi forense di un dispositivo mobile è incentrata nel capire quali app vengono per lo più usate, in primis sicuramente quelle di messaggistica. Diciamo che la differenza con l'analisi forense svolta su un personal computer si evidenzia che in un PC la maggior parte delle informazioni cercate sono date da file. successivamente vengono cercate maggiori informazioni su un file, come i metadati, ecc. Mentre l'analisi forense di un dispositivo smartphone è incentrata alla ricerca di dati, questo perché il risultato delle interazioni tra utente e smartphone è rappresentato da dati e non file, ad esempio chiamate, messaggi, email ed altri dati di interazione con altre app. Per come sono stati implementati entrambi i S.O. le app rappresentano dei modi indipendenti cioè potrebbero essere analizzate a prescindere dal S.O. Questo perché di solito sono progettate per non interagire col S.O. se non per richiedere determinati permessi e per accedere ad alcune risorse.

19.4 Analisi - Strumenti

19.4.1 UFED Physical Analyzer



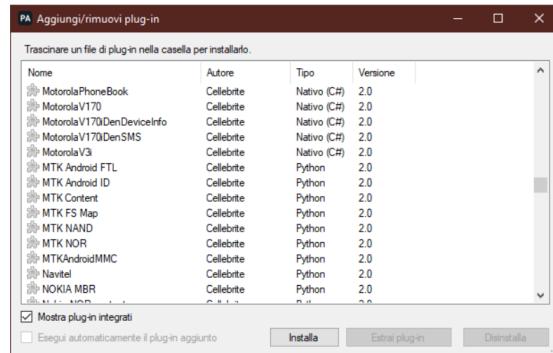
È sicuramente uno degli strumenti più conosciuti per l'analisi forense mobile. Analizza i dati estratti da un dispositivo mobile.



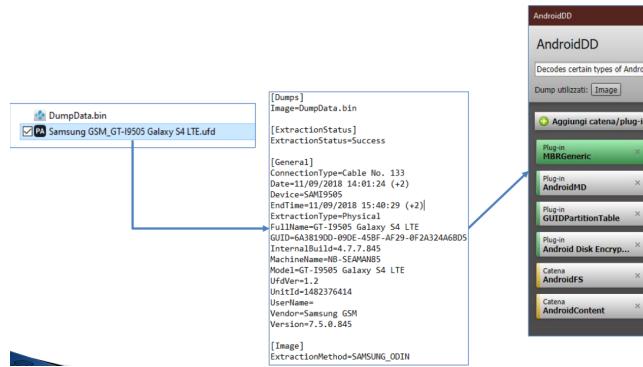
Permette l'analisi non solo dei dati estratti con il suo estrattore ma anche quello di terze parti ed inoltre permette di analizzare i backup di dispositivi mobili. Se durante l'analisi di un PC vi imbattete nel backup di un dispositivo mobile per analizzarlo basterebbe estrarlo dalla copia forense e poi darlo in pasto a questo software.

19.4.2 PlugIn

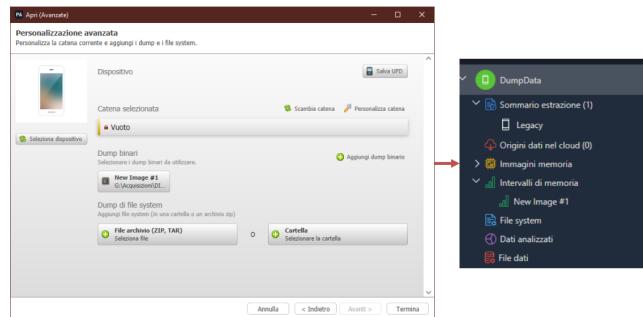
Tutte le operazioni automatiche di analisi eseguite dal software sono dirette dai plugin che sono scritti in python oppure in C#, inoltre ognuno può creare il proprio plugin ed installarlo.



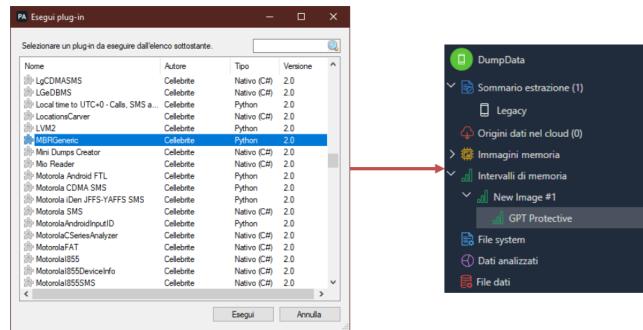
Ogni volta che si termina un'estrazione dati tramite UFED esso accompagna i dati estratti con un piccolo file con estensione **.ufd** nel quale sono racchiuse delle info sull'estrazione e che poi saranno usate dal physical analyzer per identificare quale plugin usare automaticamente.



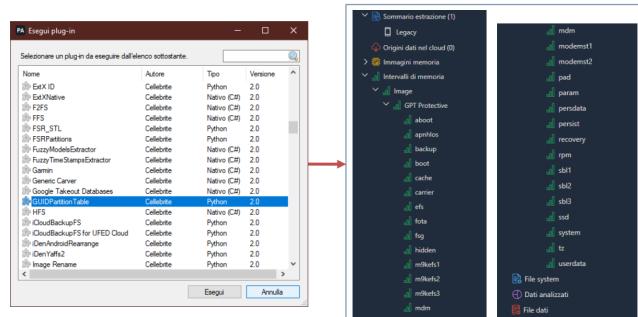
19.4.3 Creazione di una catena di PlugIn personalizzati



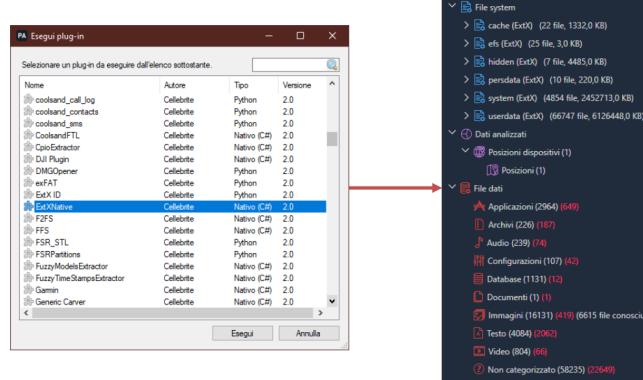
Si fornisce il **Dump Binario** essendo un acquisizione fisica e non si specifica una catena. Lo strumento forense senza alcuna catena caricherà soltanto l'immagine, crea un intervallo di memoria chiamato **New Image #1**.



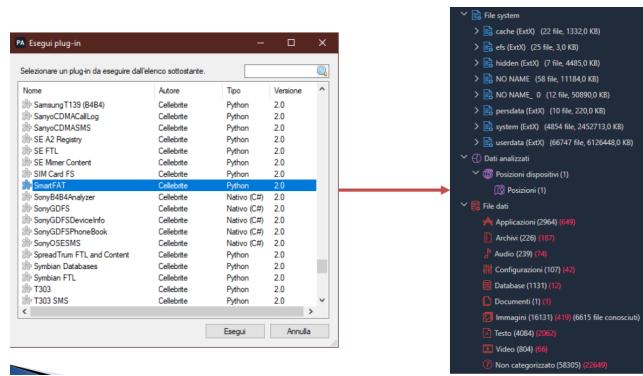
Essendo un acquisizione fisica la prima cosa da fare è iniziare a leggere il sistema di partizionamento, eseguire quindi quei plugin che si occupano di decodificare il sistema di partizionamento nel disco. Il plugin **MBRGeneric** riconosce una sola partizione GPT Protective.



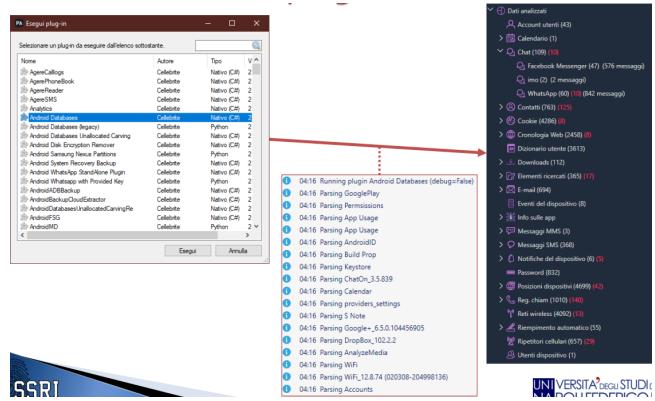
Eseguiamo quindi il plugin adatto alla partizione e decodifichiamo e suddividiamo tutte le partizioni nel volume.



Una volta divise le partizioni non resta altro che analizzare i file system presenti, essendo Android sarà il plugin **ExtXNative**. Dopo l'esecuzione di questo plugin si inizia a riempire la parte dati.



Si va alla ricerca di ulteriori partizioni di tipo FAT.



Ovviamente stiamo parlando di file, che nei dispositivi mobili vengono rappresentati in database. Tramite la catena di plugin [Android Databases](#) lo strumento fa il parsing di tutte le possibili app che possono trovarsi su un dispositivo Android ed andando così a decodificare i vari database.

#	Partecipanti	Ultima attività
40	4 2 491579 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	19/11/2016 20:04(UTC+0) 19/11/2016 23:11(UTC+0)
41	7 1 491579 [REDACTED]@whatsapp.net (proprietario)	06/09/2016 21:41(UTC+0) 21/10/2016 23:57(UTC+0)
42	5 2 491579 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	28/08/2016 17:59(UTC+0) 14/10/2016 19:20(UTC+0)
43	16 28 2 491579 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	24/08/2016 18:19(UTC+0) 27/12/2016 22:22(UTC+0)
44	46 49 2 491579 [REDACTED]@whatsapp.net +491579 [REDACTED]@whatsapp.net (proprietario)	22/08/2016 16:11(UTC+0) 04/10/2017 20:29(UTC+0)
45	6 1 491579 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	12/08/2016 11:24(UTC+0) 15/08/2016 18:49(UTC+0)
46	2 9 2 447481 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	10/08/2016 07:44(UTC+0) 23/05/2017 19:39(UTC+0)
47	27 2 491579 [REDACTED]@whatsapp.net 491579 [REDACTED]@whatsapp.net (proprietario)	08/08/2016 12:10(UTC+0) 15/08/2016 18:52(UTC+0)

Essendo dei database bisogna decodificare i dati in modo da poterli comprendere, ad esempio eseguendo le Join tra le tabelle come farebbe l'applicazione madre.

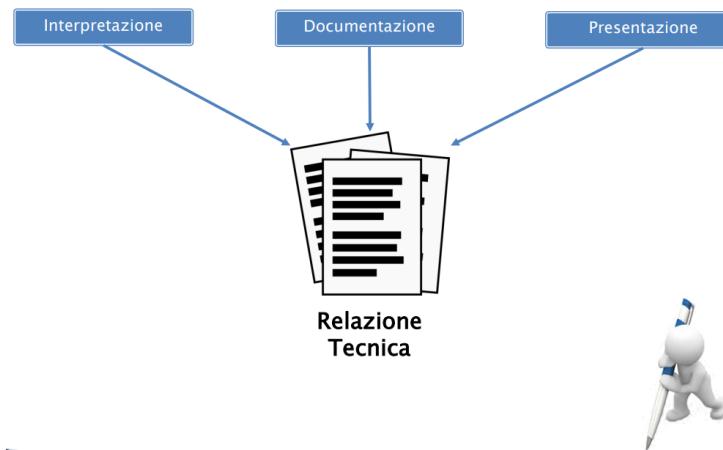
Numero righe	Nome
1	_jobqueue-WhatsAppJobQueue.db
1948	axolotl.db
6	chatsettings.db
0	chatsettingsbackup.db
3	Cookies
7713	emojidictionary.db
1	google_app_measurements.db
1	hsmpacks.db
1	location.db
2	media.db
1206	msgstore.db
399	wa.db
3	Web Data
0	web_sessions.db

Questi sono i databases dove risiedono la maggior parte dei dati di WhatsApp, le info estratte sono l'unione di più database.

20 Lezione 21

20.1 La relazione Tecnica - Le Fasi

La **relazione tecnica** non è rappresentata da un modell standard. Queste sono le varie fasi della relazione tecnica che possiamo individuare:



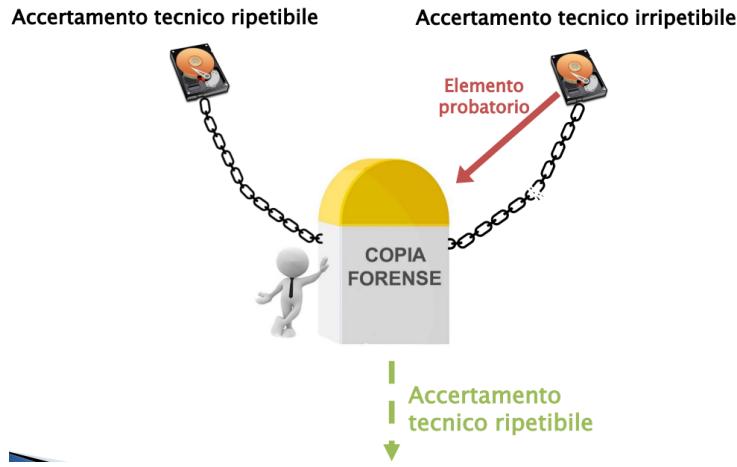
20.2 La prova digitale

Il fatto di lavorare con evidenze digitali ha dei pro e dei contro:

- **CONTRO:** il contro più grande è che la prova è facile da corrompere.
- **PRO:** come pro abbiamo il fatto che possiamo copiare il dato quante volte si vuole e senza perdere qualità o corromperne la forma.

20.3 Accertamenti

La parte più delicata riguarda la creazione della prima prova, ovvero quale che avviene sul supporto originale.



20.3.1 Accertamento Ripetibile

- Agire in modo da non alterare la prova.
- Agire in modo da documentare ogni azione compiuta su di esse.
- Porre la controparte in condizioni di poter replicare quanto fatto.

Abbiamo già visto come evitare di alterare una prova, quindi ora vedremo i successi due punti affinché un accertamento sia ripetibile. Ogni azione eseguita deve essere ben documentata, molto spesso però la lettura di alcune perizie porta ad una copiosa introduzione e conclude la documentazione dopo poche pagine senza prestare molta attenzione su cosa è stato compiuto dal consulente, fornendo poca trasparenza e risultando poco professionale.

Per quanto riguarda la fase conclusiva, ovvero concludere o valutare l'ipotesi iniziale. L'ipotesi è data da quanto richiesto dal P.M. o dal committente, ovvero la verifica è il metodo utilizzato per dare una risposta al quesito e le conclusioni dovrebbero fornire quanto richiesto o, in alcuni casi, smentire.

20.4 La Relazione Tecnica

La base di partenza è il quesito presentato e seguito da una serie di linee d'azine previste, una lista dell'hardware e del software impiegato, una descrizione di ogni azione intrapresa che abbia o meno portato al risultato. Lo scopo finale è quello di poter dare la possibilità a chiunque legga la relazione di giungere alle medesime conclusioni.

Esempio Pratico: indagine di pedopornografia (Approccio per strati).

1. Ricerca di un evidenza diretta, ovvero utilizzare un tool di ricerca per estrarre dal file system tutti i file grafici/video per poi farne una cernita. In caso di evidenze significative allora si sarebbe ottenuto il materiale per rispondere al quesito del magistrato con una semplice azione.
2. Ricerca più estesa, ovvero sempre all'interno del file system è possibile effettuare una ricerca più complessa per cercare ad esempio file compressi e cercare in questi archivi se ci sono file positivi.
3. Senza fermarsi solo ai file, ricerchiamo la presenza di programmi P2P e di sharing, con bittorrent si condividono anche i file che si stanno scaricano, quindi la presenza di qualche frame di un filmato pedoporno durante tutto il download di un file può dare riscontro positivo per delle prove.
4. Ricerca dei file cancellati.
5. Ricerca di periferiche di archiviazione agganciate.
6. Analisi steganografica, ovvero di dati occultati.

Nella relazione tecnica tutto deve essere descritto nei minimi dettagli anche ad esempio la configurazione hardware della macchina che viene usata per eseguire l'analisi, il sistema operativo, la versione dei tool utilizzati in analisi, in casi di programmi ad-hoc anche quelli vanno documentati con codice annesso. In conclusione quindi il contenuto della relazione tecnica, che si attiene strettamente ai quesiti, evitando il dilungarsi su questioni di poco conto, ove venga richiesto di essere più precisi nella descrizione o correlando la relazione con una documentazione fotografica/cinematografica, questa parte è definita **Parte Descrittiva** e deve essere ben distinta dalla **Parte Valutativa** che contiene le motivazioni del consulente e l'iter logico con il quale è arrivato a queste ultime. Inoltre il consulente dovrà evitare qualsiasi valutazione di tipo giuridico nella relazione, se ne occuperanno il magistrato o l'avvocato. Giuridicamente, inoltre, i pareri ed i fatti riportati dal consulente tecnico non sono vincolanti per il giudice che può disattenerle attraverso una valutazione critica e motivata.

20.5 Forma della Relazione Tecnica

Ad oggi la relazione assume due formati, uno digitale ed uno cartaceo. Viene inoltre composta da quattro parti:

1. **Parte Epigrafica:** parte nella quale il consulente tecnico indica gli estremi del P.P., P.M., del Giudice, le parti presenti ad un accertamento, le operazioni compiute e la descrizione dell'incarico.
2. **Parte Descrittiva:** parte in cui il consulente tecnico illustra gli accertamenti o le ricostruzioni effettuate.
3. **Parte Valutativa:** parte in cui il consulente tecnico risponde ai quesiti del magistrato, giudice oppure dell'avvocato, motivando esaustivamente le proprie conclusioni.
4. **Parte Riassuntiva:** qui il consulente tecnico espone in modo sintetico ogni risposta ad ogni singolo quesito posto, riepilogando le risultanze dell'analisi in poche righe.

Diciamo inoltre che la relazione tecnica deve essere redatta in modo **Chiaro ed Intellegibile** impiegando grafici, illustrazioni, tabelle o qualsiasi metodo grafico per illustrare al meglio i concetti.