

# Mac OS X Forensics

Luca Mariani e Andrea Nardinocchi

# Capitolo 1

## L'analisi

Estrapolare informazioni e prove dalla macchina dell'indiziato è l'arduo compito dell'analista forense. L'integrità primaria del disco è pressochè fondamentale per cui la creazione di immagini forensi (o dump certificati) del contenuto della macchina è il primo passo in assoluto (il *come* lo vedremo nel capitolo successivo dedicato alla suite di software in circolazione diretti proprio allo svolgimento di queste operazioni). Una volta che i dati sono memorizzati e assicurati da hash MD5 che svolgono il compito di checksum sull'immagine disco per garantirne l'integrità, l'analista mette in pratica tutta la sua esperienza.

In questo capitolo analizzeremo i segreti del mestiere dell'analista forense italiano e cercheremo di comprendere come tratta nello specifico le famose macchine made in Cupertino: *i Mac*.

### 1.1 L'avvio, la protezione dalle viscere del Firmware

Il criminale. Ci troviamo di fronte ad un soggetto che statisticamente non è particolarmente preparato in materia sicurezza. Forza gioco, l'eccezione va comunque tenuta in considerazione e perciò partiremo dal presupposto che il nostro indagato abbia deciso di vendere cara la sua pelle. Attraverso l'utilizzo del disco di installazione del sistema operativo della sua

macchina (la versione è indifferente: Mac OS X permette di impostare la password del firmware dalla prima release ad oggi) ha deciso di impostare una password che permette di impedire qualsiasi azione differente dall'avvio del disco da lui stesso impostato come *benedetto* (bless, quasi sempre il disco della macchina stessa): il criminale ha impostato una protezione direttamente nel Firmware della macchina. *Il Firmware* che con rispetto parlando può essere paragonato al BIOS di qualsiasi altra macchina non Apple, è un sistema memorizzato in un registro del computer che ha lo scopo di avviare il disco benedetto e permettere all'utente di eseguire operazioni preliminari (come ad esempio avviare la macchina da CD o DVD). A questo punto è facile dell'accensione potrebbe essere sgradevole anche per attacchi semplici come un brute force. Risolvere il problema è invece piuttosto banale: avere fisicamente sotto mano la macchina è un vantaggio di inestimabile valore. Sin dalla generazione degli iBook Clamshell (stiamo parlando di macchine precedenti addirittura al 2000, difficile reperire computer sospetti limitrofi a quella data) esiste un exploit che permette di resettare definitivamente la password del Firmware di ogni Mac in modo da autorizzare l'analista ad eseguire qualsiasi operazione esso voglia. Modificando la quantità di RAM presente nella macchina (rimuovendo o aggiungendo uno o più moduli dal sistema) non apparirà più una richiesta di password dal Firmware, fino a quando non verrà ripristinato il quantitativo di memoria originale. A questo punto, avviando il computer premendo la combinazione di tasti mela-alt-P-R sarà possibile eseguire il reset totale della *PRAM*, una memoria di una manciata di byte contenenti dati necessari al Firmware come: volume delle casse (per il suono caratteristico del sistema operativo all'avvio), il fuso orario e, appunto, la password del Firmware. A questo punto, dopo aver spento nuovamente la macchina e ripristinato il quantitativo di memoria installata la password del Firmware non sarà più una minaccia.

## 1.2 Accesso ai dati delle applicazioni

L'accesso a molti dei dati e delle informazioni memorizzate all'interno del sistema è veramente semplice se ci mettessimo a cercare quei particolari file di configurazione che l'applicazione usa per lo storage. Diverse aziende si sono lanciate nel quasi vuoto mercato forense dedicato al Mac rilasciando software costosissimi (oltre i 2000 dollari) che altro non fanno che elencare questi files e mostrarli attraverso l'uso di interfacce carine e piacevoli permettendo poi di organizzarli in base a filtri proposti dall'analista. La struttura del filesystem del Mac è, a differenza di quello che vogliono far credere le suddette aziende, più facile di quel che appare: essendo un sistema unix-based ci capiterà di trovare nella root diverse cartelle già viste in qualsiasi distribuzione Linux presente nel web e non, dal 1994 ad oggi, comprese nella directory Users, tutte le cartelle contenenti le home dei rispettivi utenti (spero che i fanatici più incalliti non la prendano a male se avessi dovuto fallire nel riportare questa data). In realtà potremo ignorare tutto e concentrare i nostri sforzi nelle directory in cui in realtà si annidano *quasi* tutti i file di configurazione di ogni software installato:

- /Library e /System/Library, contenente tutti i file condivisi dalle varie applicazioni e dal sistema operativo stesso;
- /Users/<utente>/Library/Application Support, contenente i file di configurazione e le informazioni di ciascuna applicazione presente nella macchina nei rispettivi formati;
- /Users/<utente>/Library/Safari, contenente l'history del browser installato di default (*Safari* appunto), i download effettuati e le pagine preferite dell'utente nel formato PLIST. Il formato PLIST è puro e semplice XML per cui analizzabile tramite qualsiasi editor di testo o, ancora meglio, attraverso l'editor PLIST integrato nel pacchetto Developers del Mac;
- /Users/<utente>/Library/Mail, contenente le e-mail inviate e ricevute da ogni account dell'utente nel formato unix *mbor*;

- /Users/<utente>/Library/Mail Downloads, contenente gli allegati di ciascuna e-mail nel rispettivo formato.

Queste directory contengono quindi sostanzialmente lo scibile di qualsiasi applicazione presente nella macchina e possono essere esplorate procedendo tramite distribuzione live (l'analista solitamente predilige una distribuzione live di Linux), avviando la macchina in *Target Disk Mode*<sup>1</sup> oppure sfruttando alcuni dei software presentati nel capito successivo.

### 1.2.1 Keychain

Potremmo partire dal presupposto quasi certo che il nostro indagato si colleghi a protocolli come MSN, AIM, ICQ o Talk per chiacchierare con alcuni dei suoi amici. Potremmo supporre che abbia un account su uno o alcuni dei maggiori social network come Facebook, Twitter o Myspace dove intrattenga rapporti sociali quotidiani. Potremmo pensare che si colleghi periodicamente a forum di varia natura o che abbia un server FTP in cui faccia backup dei propri files o carichi materiale per il suo personale sito web. Ora supponiamo con un certo livello di sicurezza che l'utente abbia risposto sì alla domanda *vuoi che il sistema memorizzi le password in modo che tu non debba riscriverle ogni volta?*: il Mac ha messo in gioco l'applicazione *Keychain* e l'utente ha acconsentito a sfruttarla. Keychain si occupa di memorizzare le informazioni sensibili (come password o certificati di posta elettronica) in modo che l'utente non debba ogni volta reinserire questo genere di dati quando richiesto. Ovviamente è da tenere in considerazione che ogni password memorizzata in questa applicazione è protetta dalla password dell'account quindi depredare la macchina dei file di configurazione di Keychain non avrebbe alcuna utilità: dovremo entrare nel sistema scassinando la password utente e poi tentare un attacco di brute force su tale applicazione. Ma come fare?

---

<sup>1</sup>Tenendo premuto il tasto T all'avvio il Mac entrerà nella modalità Target Disk Mode (TDM): attraverso il collegamento con un cavo firewire sarà possibile a questo punto vedere la macchina da qualsiasi altro computer come un ingombrante hard disk esterno e sarà quindi possibile accedere liberamente al suo contenuto.

**Violiamo l'account dell'utente <user>:** Per violare l'account abbiamo la necessità di sfruttare la *single user mode* del sistema accessibile premendo all'avvio della macchina mela-S. La single user mode è una modalità caratteristica dei sistemi Unix (è possibile immaginarla su Windows come la modalità safe) utilizzata dall'amministratore per eseguire operazioni di mantenimento o ripristino del sistema in caso di emergenza. Ci troveremo di fronte un terminale con i permessi dell'utente root ed il filesystem montato in sola lettura per cui immutabile: attraverso il comando **mount -uw** / rimontiamo il filesystem in lettura/scrittura in modo da poter effettuare delle modifiche poi, attraverso il comando **passwd <user>** modifichiamo la password dell'utente scelto, mandiamo a questo punto un **sync** per eseguire un flush dei dati dalla cache del disco al disco stesso e poi **reboot** per riavviare. A questo punto potremo eseguire il login dell'utente <user> con la password appena impostata.

**Violiamo Keychain:** Sarebbe meraviglioso per noi se ora Keychain ci concedesse di dare una sbirciata a tutto il suo contenuto con la password utente appena impostata. Ovviamente questo non accadrà mai: grazie al cielo (per l'utente) la sicurezza del software impone l'inserimento della vecchia password per verificare che il cambio della stessa sia stato fatto dall'utente proprietario del sistema. Ci resta da tentare un attacco di brute force su questa applicazione tramite un tool il cui codice è composto da una ventina di righe o poco più: *OSX-keychain-brute*. OSX-keychain-brute esegue un brute force con dizionario (o sfruttando il vocabolario integrato del mac in caso venga omissso) su keychain fino a quando non trova la parola chiave: è estremamente rapido (analizza circa 60.000 termini in un minuto e trenta secondi) e di facile utilizzo e può essere gratuitamente scaricato e provato da questo indirizzo: <http://sourceforge.net/projects/potaru-pentest/files/osx-keychain-brute> .

### 1.3 Accesso ai dati dell'utente

Ovviamente non è tutto oro quello che luccica: seppur possiamo essere sicuri di dove le applicazioni memorizzino i propri dati, non potremo mai essere sicuri di dove l'utente abbia memorizzato ciò che deve tenere nascosto, soprattutto di primo acchito, all'analista. In questo caso altro non serve che olio di gomito e una ricerca a trecentosessanta gradi all'interno del filesystem. Anche in questo caso possono venirci in aiuto alcuni software, spesso volte realizzati dall'analista stesso in C oppure script in Bash, per la ricerca all'interno del filesystem di file che rispettino particolari criteri tra cui: tipo di file (immagine, audio, documento, archivio o video), contenuto di tags sospetti nel nome e all'interno del file e posizione (se si dovesse trovare in una directory nascosta sarebbe ancora più sospetto, non trovate?). Il seguente sorgente C ad esempio, scritto in meno di 20 minuti e dal costo pari a 0\$, consente di eseguire una ricerca all'interno delle directory nascoste di tutto il sistema (le directory nascoste Unix hanno un punto di fronte al nome) di file con particolari estensioni e particolari tags presenti nel nome e segnalare le directory che presentano un livello di file sospetti superiori ad una certa percentuale impostata dall'utente. È da notare la semplicità del sorgente:

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <sys/types.h>
#include <sys/dir.h>
#define _dseparator "/"
char *pointerextensions, defaultextensions[] = ".JPG.JPEG.GIF.PNG.RAW.PDF.
    MP3.WAV.MPG.MKV.AVI.DIVX.XVID.TIF.TIFF.TGA";
static char const * const defaultkeywords[] = { "PORN", "SEX", "CHILD", "
    NULL" };
int searchkey (const char *singleton) {
    int index = 0;
    while ((strcasecmp(defaultkeywords[index], "NULL")) != 0) {
```

```

    if (strcasestr(singleton, defaultkeywords[index]))
        return index;
    index++;
}
return -1;
}

int expand (const char *startpath, unsigned int deepsearch, float limit) {
    DIR *directory;
    struct dirent *informations;
    char *completepath, *extension;
    unsigned int completesize, elements = 0;
    int backupkeyword;
    float suspect = 0, percentage;
    if ((directory = opendir(startpath))) {
        while ((informations = readdir(directory))) {
            if ((strcmp(informations->d_name, ".") != 0) && (strcmp(informations->
                d_name, "..") != 0)) {
                if (informations->d_type == DT_DIR) {
                    completesize = strlen(startpath)+strlen(_dseparator)+strlen(
                        informations->d_name);
                    if ((completepath = (char *) malloc (completesize+1))) {
                        /* generate the "complete" path name from the selected file so
                           we can call it recursively */
                        snprintf(completepath, completesize, "%s%s%s", startpath,
                            informations->d_name, _dseparator);
                        expand(completepath, (informations->d_name[0]!='.'), limit);
                        free(completepath);
                    } else return 1;
                } else if (deepsearch) { // searching for file's type
                    if (((extension = strrchr(informations->d_name, '.')) && (
                        strcasestr(pointerextensions, extension))) suspect++;
                    else if ((backupkeyword = searchkey(informations->d_name)) > 0)
                        suspect++;
                }
            }
        }
    }
}

```



```

        elements++;
    }
}
if (deepsearch) {
    if (elements > 0) {
        percentage = ((float)(100 * suspect)/(float)elements);
        if (percentage > limit)
            printf("\n[DIRECTORYSEARCH: notice]\n%s\n[total elements: %d]
                percentage %.1f of suspicious material\n\n", startpath,
                elements, ((100.0*suspect)/elements));
    }
}
closedir(directory);
} else return 1;
return 0;
}

int main (int argc, char *argv[]) {
    if (argc >= 4) {
        if ((pointerextensions = (char *) malloc (strlen(argv[3])+1)))
            strcpy(pointerextensions, argv[3]);
    } else {
        if ((pointerextensions = (char *) malloc (strlen(defaultextensions)+1)))
            strcpy(pointerextensions, defaultextensions);
    }
    printf("[DIRECTORYSEARCH] Exploring entire filesystem (this will take
        several minutes) (%s <starting path> <minimum suspicious>)\n", argv[0])
        ;
    expand((argc>=2)?argv[1]:"/", 0, (argc>=3)?atof(argv[2]):0.0);
    printf("\n[DIRECTORYSEARCH] ended\n");
    free(pointerextensions);
    return 0;
}

```



## Capitolo 2

# I software utilizzati

Esistono molti tools che permettono la creazione di copie forensi di dischi (sfruttando un altro disco fisico) o la creazione di immagini del disco. Altri software si occupano invece dell'analisi specifica, la ricerca di determinati tipi di file e estensioni. Infine, anche il bloccare la scrittura sul disco stesso, per evitare l'inquinamento delle prove, è uno dei maggiori problemi di un analista forense.

In questa parte della trattazione, ci si concentrerà sull'analisi di quei software che permettono questo genere di analisi, sia che essi siano gratuiti o a pagamento, cercando di fare una panoramica il più possibile esaustiva degli strumenti che un analista forense ha a disposizione.

### 2.1 BlackBag

La BlackBag Technologies Inc. fornisce soluzioni per l'analisi forense di sistemi Mac e dati eDiscovery per privati. La sede è situata a Silicon Valley, offre ai clienti un pacchetto di servizi, software e anche corsi di formazione.

Ad oggi, fornisce servizi ad una vasta clientela, tra cui federali, forze dell'ordine statali e locali così come a settori di sicurezza privata.

BlackBag mette a disposizione di analisti forensi esperti mezzi per creare immagini, bloccare la scrittura per preservare le informazioni e per l'analisi effettiva.

I maggiori prodotti della BlackBag possono essere suddivisi in tre categorie:

- Creazione Immagini, attraverso un software basato direttamente sul sistema operativo OS X, *MacQuisition*;
- Analisi, con il software *BlackLight*, in grado di analizzare sistemi iOS (iPhone, iPad), i dati su Mac OS X e Classic (OS 9), assicurando alti livelli di accuratezza;
- Blocco di scrittura, con *SoftBlock*, un tool forense basato sul kernel che identifica i dispositivi al momento della connessione e, a seconda delle preferenze dell'utente, li monta sia in sola lettura, che nella configurazione lettura-scrittura.

Si passerà ora ad analizzare in maniera più approfondita questi tre software.

**MacQuisition** Testato per oltre dieci anni su diversi sistemi, attualmente è funzionante su più di 185 piattaforme Mac conosciute. Offre agli esaminatori una soluzione affidabile che permette di evitare complicate ed estremamente lunghe operazioni.

Basandosi direttamente sull'OS, esso utilizza il sistema operativo del sospetto per creare l'immagine. Può essere facilmente utilizzato sia in laboratorio che sul campo da un ampio raggio di utenti. Infatti, consente la creazione di immagini in cinque semplici passi per coloro che si avvicinano per la prima volta al suo utilizzo, mentre mette a disposizione degli analisti più esperti un gran numero di opzioni per la personalizzazione.

Grazie all'ultima versione del software, supporta i sistemi basati su Intel i5 e i7, oltre che gli ultimi modelli di MacBook Air. Non include però l'ultima generazione di MacBook Pro (MacBookPro8,X, rilasciati nel 2011).

I metodi per creare un immagine di un Mac sono essenzialmente tre:

- Rimuovere l'hard disk e usare un adattatore per creare l'immagine del drive:
- Usare un CD/USB bootable come Raptor e/o Helix
- Creare un drive Mac OS X bootable, fare il boot da esso e utilizzare il comando *dc3dd*.

Ovviamente questi tre metodi possono non risultare estremamente comodi. MacQuisition è stato specificatamente ideato per creare immagini di computer Mac. Usa una versione di OS X autorizzata della Apple, così da non violare la licenza.

Lavora firewire e USB, cosicchè risulta possibile creare immagini di quasi tutti i sistemi Apple.

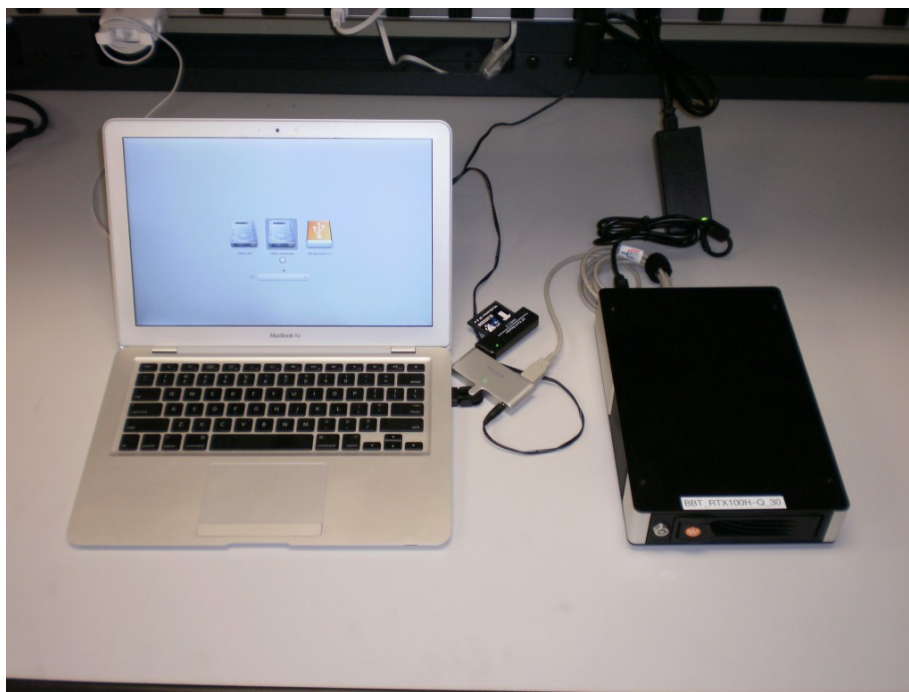


Figura 2.1: MacQuisition in fase di creazione dell'immagine di un MacBook Air.

In cinque semplici passi, il programma permette di creare l'immagine del sistema:

1. Passo 1: L'utente seleziona la periferica di cui si intende creare l'immagine e toglie la protezione dalla periferica in cui si intende salvare l'immagine dalla lista di tutte le periferiche disponibili;

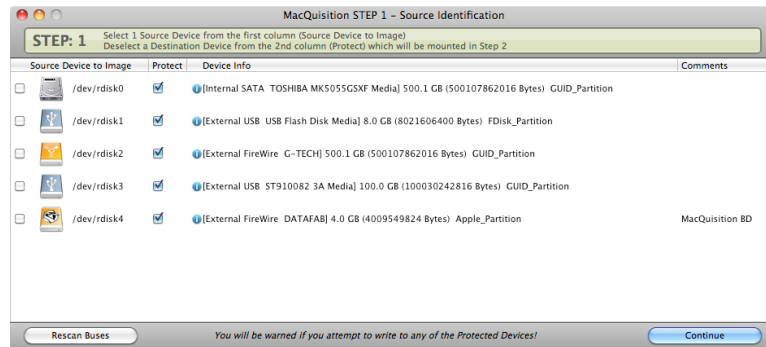


Figura 2.2: MacQuisition: Fase 1.

2. Passo 2: Selezione della periferica di destinazione tra le periferiche non protette;

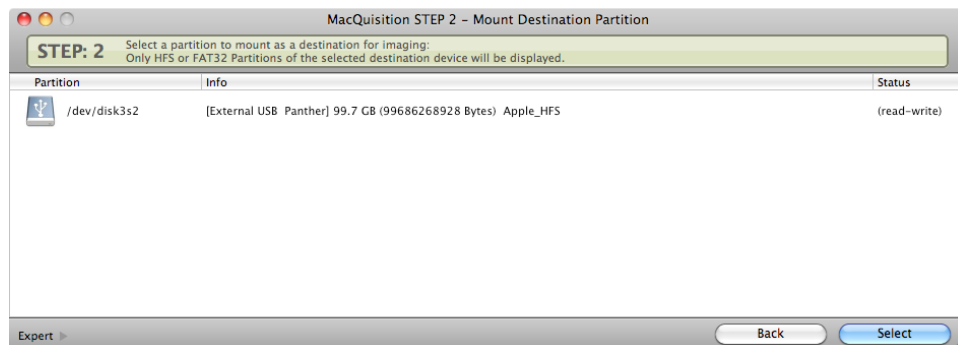


Figura 2.3: MacQuisition: Fase 2.

3. Passo 3: L'utente inserisce alcuni dettagli, tra i quali il nome, il time zone, etc. ;
4. Passo 4: In questa finestra saranno contenuti i dettagli selezionati nel passo precedente. Si richiede di controllare la loro accuratezza e di dare conferma per procedere;

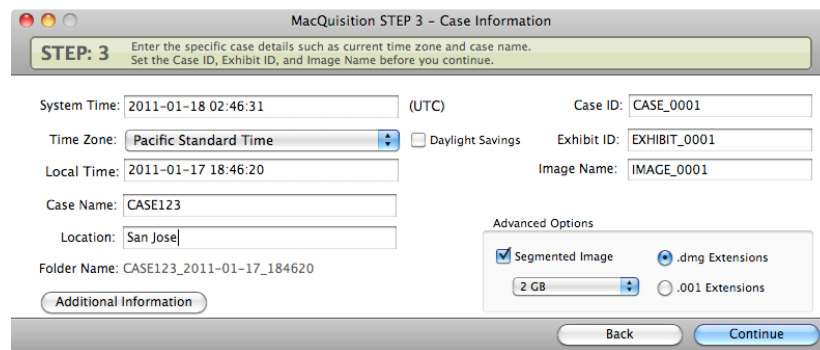


Figura 2.4: MacQuisition: Fase 3.

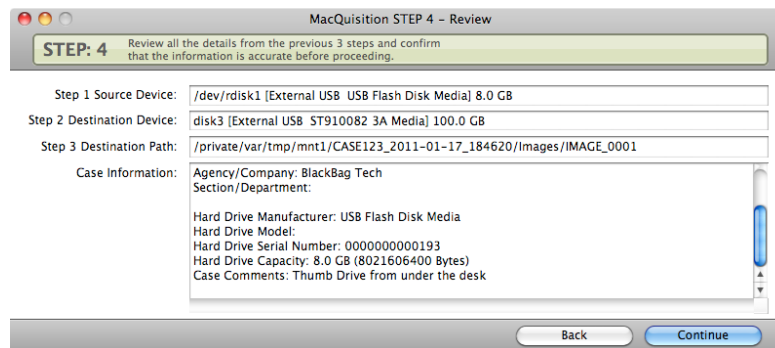


Figura 2.5: MacQuisition: Fase 4.

5. Passo 5: Si dà inizio al procedimento di creazione dell'immagine premendo il tasto *IMAGE*.

Questo software si può trovare, alla modica cifra di 599.00\$, online al sito <https://www.blackbagtech.com/foren5.html>.

**BlackLight** Questo software offre all'analista una piattaforma intuitiva per cercare, trovare, analizzare e riportare dati da Mac e iOS in un'unica applicazione.

Offre la possibilità di compiere un backup completo per iPhone e iPad, compresi un automatico hash dei file, analisi dei file conosciuti e analisi delle firme dei file. Crea una lista di ogni genere di file, compresi SMS, Voicemail, note, cronologia delle chiamate, calendario, contatti e ogni genere di informazione contenuta all'interno di questi supporti. Permette, inoltre, di ricreare, attraverso la SQLite recovery feature, file eliminati dal sospettato su centinaia di applicazioni iOS.

Identifica, separa e mostra immediatamente la lista dei file attivi e quella dei file in spazio non allocato, permettendo agli analisti di condurre un'investigazione più rapida e mirata. Inoltre identifica partizioni e supporti automaticamente, dando informazioni dettagliate su ogni supporto.

La funzione principale, ovviamente, resta quella di identificare determinate file a seconda di estensioni particolari, fornendo una lista di dati sul loro utilizzo molto dettagliata (cartella, data di creazione, data dell'ultima modifica e data dell'ultimo accesso), separando file di sistema conosciuti da quelli sospetti (funziona correttamente per le versioni di Mac OS X dalla 10.0.0 alla 10.6.7).

Per quanto riguarda le immagini grafiche, il software le analizza e le identifica automaticamente, mostrandole e offrendo numerose opzioni di ordinamento tra le quali anche l'analisi del colore della pelle, utile per organizzare l'ordine in base alla quantità di pelle presente all'interno delle fotografie (e identificare così più facilmente le foto di nudi).

E' anche disponibile una funzione di stop dell'analisi nel momento in cui, ad esempio, l'analista ha bisogno di una pausa. Questa funzione è attiva anche nel momento in cui il sistema dovesse crashare, evitando così la perdita dei dati ricavati fino a quel momento.

Oltre che il controllo di file presenti all'interno del computer, il programma può effettuare analisi sulla cronologia di Safari, Firefox e GoogleChrome, fornendo una rapida visione degli URL includendo data della visita, il tempo passato sul sito e eventuale testo scritto



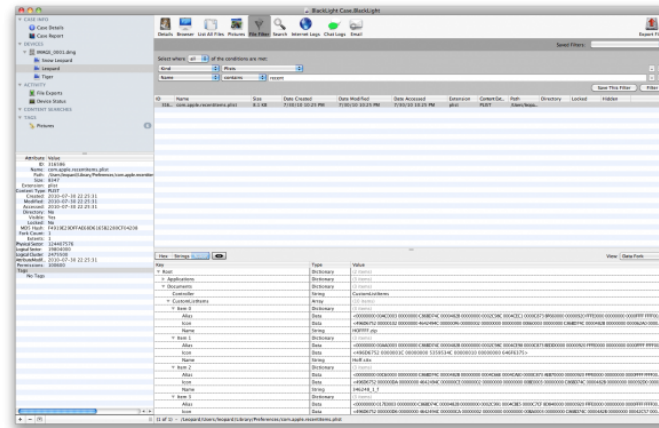


Figura 2.6: BlackLight: La funzionalità di filtraggio dei file personalizzabile dall'utente.

attraverso la ricerca nei Cookies.

Supporta anche i formati chat di Adium e iChat, mostrando automaticamente i file con il contenuto delle conversazioni, le date, i nomi e eventuali file scambiati.

Il prezzo di questo software è 2,499.00\$, è possibile acquistarlo al sito <https://www.blackbagtech.com/forensics/blacklight-6.html>.

**SoftBlock** Prodotto ideato per un utilizzo a larga scala, sia in laboratori forensi che per investigatori privati, per necessità di analizzare e avere una preview di uno o più eventuali supporti probatori. Permette un rapido e semplice montaggio e smontaggio di dispositivi multipli, se necessario.

Ha un'interfaccia intuitiva che permette di effettuare rapide preview e quindi determinare

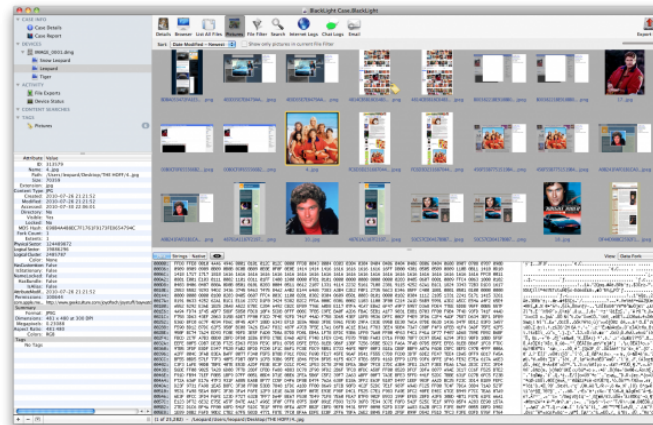


Figura 2.7: BlackLight: Identificazione e catalogazione delle immagini.

se sono necessari analisi approfondite o meno.

In pratica, racchiude in sé la possibilità di montare e dare un primo sguardo a possibili prove, senza il bisogno di installare ulteriori programmi che blocchino la scrittura sull'hardware per non andare a intaccare l'analisi.

E' acquistabile al sito <https://www.blackbagtech.com/forensics/softblock/softblock.html>, il prezzo di vendita è 239.00\$

## 2.2 MacForensicsLab

Prodotto della SubRosaSoft.com Inc., è considerato il più potente e il più bilanciato in fatto di qualità-prezzo tra i software per l'analisi forense. Funziona su ogni tipo di piattaforma, fornendo una vasta gamma di opzioni che racchiudono in questo programma la maggior parte delle funzionalità di cui un analista forense ha bisogno.

Veloce, con un alta tolleranza d'errore e la possibilità di verificare le acquisizioni, è in grado di produrre una replica esatta del supporto originale, massimizzando il recupero dei dati anche in presenza di dati corrotti. Le immagini forensi sono create mediante un hashing integrato. Consente inoltre la duplicazione delle immagini e dei file hash associati, in modo da ridurre il tempo che l'analista forense impiega nella fase di acquisizione dei dati.

Permette ai professionisti dell'analisi forense di trovare e recuperare i file associati e eliminati, fornendo un anteprima prima dell'effettivo procedimento di recupero. In questa operazione può essere esaminato anche lo spazio non allocato, trovando così eventuali prove di cui si ha bisogno.

Il software è ottimizzato per l'utilizzo attraverso i vari standard SQL del database server, consentendo la collaborazione investigativa permettendo l'accesso simultaneo e l'elaborazione di ogni caso specifico. Ogni azione effettuata viene memorizzata nei registri, esplorabili in maniera personalizzabile.

Crea anteprime di immagini grafiche, automaticamente filtrate in base ai toni della pelle, al tipo di immagine e alla dimensione del file. E' in grado, inoltre, di effettuare una ricerca in base a parole chiave, catalogando le prove in base a differenti lingue. Vi è anche una funzione di ricerca di eventuali numeri di codice fiscale e di carte di credito.



Figura 2.8: MacForensicsLab: Creazione dell'immagine di un supporto.

L'utente è, inoltre, in grado di personalizzare al massimo la ricerca, per ridurre i tempi della ricerca variandola a seconda del tipo di reato in esame. Funziona dagli iMac di prima generazione fino alle ultime macchine con processore Intel, oltre che su Windows, Linux e altri sistemi operativi.

Il costo di questo software è di 1,495.00\$; acquistabile online al sito [www.macforensiclab.com](http://www.macforensiclab.com) assieme ad altri prodotti più specifici.

## 2.3 The Sleuth Kit

Si tratta di un progetto Open Source per fornire tools digitali per l'investigazione forense. Fornisce una raccolta di linee comando utilizzabili su Windows e sui sistemi Unix (come



Figura 2.9: MacForensicsLab: Strumento di ricerca delle immagini.

Linux, OS X, Cygwin, FreeBSD, OpenBSD e Solaris), per l'analisi di file system di vario tipo, tra i quali NTFS, FAT, HFS+, Ext2, Ext3, UFS1 e UFS2.

Due pacchetti vengono forniti all'utente, scaricabili dal sito [www.sleuthkit.org](http://www.sleuthkit.org):

- The Sleuth Kit, una libreria C e un insieme di linee comando;
- Autopsy, l'interfaccia grafica per lo Sleuth Kit.

Si andrà ora ad analizzarli nello specifico.

**The Sleuth Kit** Le linee comando contenute in questo kit, permettono di analizzare e di trovare eventuali prove. Inizialmente il progetto fu avviato in collaborazione con @stake (Symantec), per poi proseguire indipendentemente sotto il controllo di un gruppo di sviluppatori, risultando ora totalmente indipendente da organizzazioni commerciali e accademiche.

Basato su *The Coroner's Toolkit* ([www.porcupine.org/forensic/tct.html](http://www.porcupine.org/forensic/tct.html)), solo nel marzo

2001 è stata implementata l'interfaccia grafica ad esso correlata.

Può essere utilizzato in due diversi modi: la libreria in C può essere incorporata in molti altri software forensi per incrementarne le capacità e le linee comando possono essere utilizzare direttamente dall'utente.

Le prove sono rintracciabili all'interno del file system. Le linee comando contenute nel pacchetto, consentono differenti approcci ai dati, tra i quali troviamo la ricerca, il recupero di file cancellati e molto altro.

In seguito viene riportato il procedimento da effettuare per ricercare un file. Ulteriori informazioni sulle potenzialità del prodotto sono descritte nel sito [wili.sleuthkit.org](http://wili.sleuthkit.org).

Viene ricercata, in questo esempio, lo spazio non allocato dell'immagine wd0e.dd per la stringa abcdefg. Il primo passo è quello di estrarre l'unità disco non allocata utilizzando il tool blkls.

```
# blkls images/wd0e.dd > output/wd0e.blkls
```

Successivamente, utilizzando l'utility *UNIX string*. Nel caso in cui è da ricercare un unica stringa, questo passaggio può essere saltato, mentre nel caso in cui è necessario cercare differenti stringhe, esso sicuramente rende la ricerca molto più breve.

```
# strings -t d output/wd0e.blkls > output/wd0e.blkls.str
```

Si utilizza poi l'utility *UNIX grep* per ricercare la stringa.

```
# grep abcdefg output/wd0e.blkls.str | less
10389739: abcdefg
```

Si nota che la stringa è posizionata al byte 10389739. Successivamente, si determina quel frammento. Per farlo si usa il tool *fsstat*.

```
# fsstat openbsd images/wd0e.dd
<...>
CONTENT-DATA INFORMATION
-----
Fragment Range: 0 - 266079
Block Size: 8192
Fragment Size: 1024
```

Questo ci mostra che il frammento ha lunghezza 1024 byte. Usando una calcolatrice, è facile controllare che 10389739 diviso per 1024 fornisce come risultato 10146. Questo significa che la stringa *abcdefg* è allocata nel frammento 10146 del file generato da *blkls*. Questo non ci aiuta, poichè questo file non è realmente un file del sistema. Per vedere l'intero frammento dall'immagine, è possibile utilizzare il comando *dd*.

```
# dd if=images/wd0e.dd bs=1024 skip=10146 count=1 | less
```

Successivamente vogliamo identificare dove si trova il frammento nell'immagine originale. Utilizzeremo il tool *blkcalc*, che ci fornirà in output la locazione nell'immagine.

```
# blkcalc -u 10146 images/wd0e.dd
59382
```

Per vedere poi il contenuto del frammento trovato, è possibile utilizzare *blkcat*.

```
# blkcat images/wd0e.dd 59382 | less
```

Tutta questa sequenza di passaggi, può essere effettuata automaticamente mediante *Autopsy*.

**Autopsy** Autopsy è un interfaccia grafica che mostra dettagli riguardo a dati rimossi e alla struttura del file system.

Procede attraverso due tipi di analisi: la *Dead Analysis* che avviene nel momento in cui un sistema di analisi dedicato viene utilizzato per esaminare i dati da un sistema sospetto, la *Live Analysis* che avviene nel momento in cui il sistema sospetto deve essere analizzato mentre è in funzione.

Permette inoltre di effettuare una serie di operazioni, basate sempre sul The Sleuth Kit, tra le quali troviamo la catalogazione dei file, il controllo dei contenuti, ricerca di parole chiave, analisi delle immagini e molti altri.

## 2.4 dcfldd

L'ultimo tool analizzato è il *dcfldd*, un tool unix basato sul comando GNU *dd*, che copia dati in blocchi, eventualmente effettuando conversioni.

Scaricabile dal sito <http://dcflfl.sourceforge.net>, esso è freeware. Uno dei maggiori vantaggi rispetto a *dd* è il fatto di supportare l'hashing di dati quando viene creata un immagine del disco, permettendo la verifica del fatto che l'immagine non sia stata modificata dalla sua acquisizione.

Scritto da Nicholas Harbous, impiegato del *Department of Defence Computer Forensics Lab* (DCFL), è in grado di verificare se sono state eliminate immagini, proteggere l'integrità



dei dati, dare più risultati contemporaneamente dividendoli in maniera personalizzabile, etc. .

Nel sito, è inoltre presente un manuale che descrive appieno le potenzialità di questo prodotto.