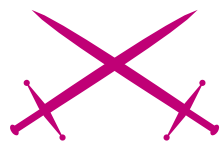


Hard Disk Forensics e Specifiche ATA



Attacco

Maurizio Anconelli 

Grado di difficoltà



Quando si parla di Computer Forensics la prima immagine a far da sfondo ai concetti è una scatoletta di alluminio non più grande di una mano, con qualche inserto nero ed una base intarsiata da circuiti, chip e saldature. Solitamente è ben nascosta all'interno dei case, e la presenza fisica è smascherata unicamente da un rumore quasi impercettibile ad ogni modifica di mac times.

Proprio il sommesso ronzio delle testine sarà la colonna sonora di questo articolo che ha come obiettivo la comprensione di alcuni aspetti spesso ignorati degli hard disk, mettendo in pratica sistemi che spesso sono affrontati solo a livello teorico.

Il sistema migliore per nascondere i dati su un supporto è ovviamente la crittografia. Esistono tuttavia metodi alternativi che sfruttano alcuni comandi degli standard ATA ed alcune caratteristiche degli hard disk spesso sconosciute al normale utente del pc, a causa della scarsa comodità di utilizzo.

Pur non avendo lo stesso grado di sicurezza ed inviolabilità dei sistemi crittografici, questi sistemi presentano caratteristiche in grado di modificare ed aumentare le possibilità di occultamento dei dati. Per questi motivi i concetti esposti nel seguente articolo rappresentano un bagaglio necessario sia per chiunque si occupi di computer forensics ed investigazione digitale che per chi si occupi di sicurezza informatica in generale, come a chiunque voglia comprendere aspetti particolari dell'hard disk.

Le procedure che andremo ad analizzare sono in gran parte manuali e presuppongono una discreta conoscenza della struttura fisico/

logica di hard disk e master boot record, del quale spiegheremo brevemente il funzionamento. Come per la maggior parte degli articoli tecnici, il sistema migliore per comprendere a fondo i concetti consiste nel tenere la mano sinistra sulla rivista e la destra sulla tastiera. Trattandosi di interazione a basso livello con l'hard disk e di procedure potenzialmente distruttive, il mio vivo consiglio è comunque quello di utilizzare un hard disk dedicato e lasciar perder il RAID 5 del mail server aziendale.

Dall'articolo imparerai...

- alcuni sistemi di occultamento dati su Hard Disk,
- le funzionalità ed i comandi ATA importanti in Computer Forensics,
- leggere il *Master Boot Record* con un editor esadecimale,
- utilizzare dd per la manipolazione delle partizioni nel *Master Boot Record*.

Cosa dovresti sapere...

- funzionamento di base dell'Hard Disk,
- nozioni di base su indirizzamento CHS, LBA,
- nozioni di base su partizionamento e filesystem.

Standard ATA

Gran parte di questo articolo è basato sulle caratteristiche ATA, che sta per *Advanced Technology Attachment* e consiste in una serie di standard e specifiche sviluppate dal T13 Technical Committee che controllano l'interfacciamento tra il sistema e gli hard disk, in particolare le caratteristiche fisiche, i protocolli ed i comandi che permettono la comunicazione tra gli stessi ed i dispositivi di controllo.

Alcuni dei comandi e delle caratteristiche sviluppate permettono un'interazione a basso livello con il device, nel nostro caso ne verranno utilizzate alcune per modificare le caratteristiche rilevate dai sistemi operativi. Bisogna comunque tener presente che come per ogni buon standard informatico che si rispetti, sebbene lo sviluppo delle specifiche sia ampiamente documentato, il supporto da parte dei produttori è assai

creativo, non limitandosi ad escludere o meno determinati parametri, ma modificandone a proprio uso e consumo le specifiche stesse.

Le caratteristiche ed i comandi che andremo ad utilizzare, quindi, potrebbero essere non supportati o gestiti diversamente da un particolare tipo di drive, a parità di versione ATA supportata.

Nel nostro caso andremo ad approfondire l'utilizzo di HPA,

Listato 1. Output dei comandi *fdisk*, *hdparm* e *mmls*

```
#fdisk -ul
Disk /dev/hda: 41.1 GB, 41110142976 bytes
255 heads, 63 sectors/track, 4998 cylinders, total
80293248 sectors

Units = sectors of 1 * 512 = 512 bytes
Device Boot Start End Blocks Id System
/dev/hda1 * 63 40965749 20482843+ 7 HPFS/
NTFS
/dev/hda2 40965750 80244674 19639462+ c W95
FAT32 (LBA)
/dev/hda3 80244675 80276804 16065 7 HPFS/
NTFS

#hdparm -gI
/dev/hda:
geometry = 65535/16/63, sectors = 80293248, start = 0
ATA device, with non-removable media
Model Number: Maxtor 6K040L0
Serial Number: K10Y3XXX
Firmware Revision: NAR61JA0

Standards:
Supported: 7 6 5 4
Likely used: 7

Configuration:
Logical max current
cylinders 16383 16383
heads 16 16
sectors/track 63 63
--
CHS current addressable sectors: 16514064
LBA user addressable sectors: 80293248
device size with M = 1024*1024: 39205 MBytes
device size with M = 1000*1000: 41110 MBytes (41 GB)

Capabilities:
LBA, IORDY (can be disabled)
Queue depth: 1
Standby timer values: spec'd by Standard, no device
specific minimum
R/W multiple sector transfer: Max = 16 Current = 16
Advanced power management level: unknown setting
(0x0000)
Recommended acoustic management value: 192, current
value: 254
DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4
*udma5 udma6
Cycle time: min=120ns recommended=120ns
PIO: pio0 pio1 pio2 pio3 pio4
Cycle time: no flow control=120ns IORDY flow
control=120ns

Commands/features:
Enabled Supported:
* NOP cmd
* READ BUFFER cmd
* WRITE BUFFER cmd
* Host Protected Area feature set
* Look-ahead
* Write cache
* Power Management feature set
Security Mode feature set
* SMART feature set
* FLUSH CACHE EXT command
* Mandatory FLUSH CACHE command
* Device Configuration Overlay feature set
* Automatic Acoustic Management feature set
SET MAX security extension
Advanced Power Management feature set
* DOWNLOAD MICROCODE cmd
* SMART self-test
* SMART error logging

Security:
Master password revision code = 65534
supported
not enabled
not locked
not frozen
not expired: security count
not supported: enhanced erase

HW reset results:
CBLID- above Vih
Device num = 0 determined by the jumper
Checksum: correct

#mmls -t dos -v -b /dev/hda
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

Slot Start End Length Size
Description
00: ----- 0000000000 0000000000 0000000001 0512B
Primary Table (#0)
01: ----- 0000000001 0000000062 0000000062 0031K
Unallocated
02: 00:00 0000000063 0040965749 0040965687 0019G
NTFS (0x07)
03: 00:01 0040965750 0080244674 0039278925 0018G
Win95 FAT32 (0x0C)
04: 00:02 0080244675 0080276804 0000032130 0015M
NTFS (0x07)
05: ----- 0080276805 0080293247 0000016443 0008M
Unalloca
```



DCO, Security Mode, G-List e P-list di un disco PATA (Parallel ATA), quello che più comunemente ed altrettanto erroneamente viene definito disco EIDE o IDE.

Programmi Utilizzati

Per la configurazione e l'analisi del disco abbiamo bisogno di uno strumento che agisca a basso livello. *Hdat2* è un programma freeware scaricabile dal sito <http://www.hdat2.com> per la diagnostica a basso livello degli hard disk che permette la modifica di parametri ATA quali DCO e HPA. È possibile scaricare il programma o un comodo eseguibile che crea il floppy disk di boot con il tool già inserito.

Per il resto utilizzeremo programmi a linea di comando come *fdisk*, *hdparm* e *mmls*, un tool dello sleuthkit (<http://sleuthkit.org>) già presente in *hakin9 live*.

Analisi del Disco

Il sistema migliore per ricavare le informazioni e le specifiche del disco è di eseguire il boot da un sistema Linux Live, il CD *hakin9 Live* allegato alla rivista è la scelta ideale, e utilizzare i comandi *fdisk*, *hdparm* e *mmls* (*sleuthkit*).

Attraverso *fdisk* visualizziamo una lista (*l*) dei dischi collegati con i relativi parametri. Il flag *u* permette di visualizzare il settore di partenza ed il settore finale al posto dei cilindri:

```
bash-3.1# fdisk -ul
```

Una volta visualizzato il drive interessato, lo passiamo come argomento ad *hdparm* che permette di ricavare

Listato 2. Output di *fdisk* e *mmls* dopo le midi fiche DCO

```
# fdisk -ul /devhda
Disk /dev/hda: 41.0 GB, 41085273600 bytes
255 heads, 63 sectors/track, 4995 cylinders, total 80244675 sectors
Units = sectors of 1 * 512 = 512 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/hda1    *            63    40965749    20482843+   7   HPFS/NTFS
/dev/hda2            40965750    80244674    19639462+   c   W95 FAT32 (LBA)
# mmls -t dos -v -b /dev/hda
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

   Slot  Start          End          Length      Size  Description
00: ----  0000000000    0000000000    0000000001  0512B Primary Table (#0)
01: ----  0000000001    0000000062    0000000062  0031K Unallocated
02: 00:00  0000000063    0040965749    0040965687  0019G NTFS (0x07)
03: 00:01  0040965750    0080244674    0039278925  0018G Win95 FAT32 (0x0C)
```

informazioni estese ed eseguire varie operazioni sull'HD mediate dal *device driver subsystem*.

Assieme all'identificativo del disco passiamo i parametri *l* (i maiuscola) che permette di ricavare informazioni direttamente dal drive, comprensive di specifiche ATA supportate e *g*, che mostra la geometria e le dimensioni del drive.

```
bash-3.1# hdparm -gI
```

Tra le informazioni, oltre a numero di serie e geometria, possiamo osservare alcune caratteristiche fondamentali per le modifiche che vogliamo effettuare:

- gli standard ATA supportati dal disco: *Standards: Supported: 7 6 5 4, Likely Used: 7*,
- la presenza di HPA e DCO tra le features ATA supportate: *Enabled Supported, Host Protected*

Area feature set, Device Configuration Overlay feature set,

- il supporto alle funzionalità ATA di sicurezza: *Security: Master password revision code = 65534, supported, not enabled, not locked, not frozen, expired: security count, supported: enhanced erase*.

Nel Listato 1 è riportato l'output completo dei due comandi.

Riassumendo l'Hard Disk Maxtor 6K040L0 ha la capacità di 39205 Mbytes (1024*1024), 40 GB circa, supporta gli standard ATA 7 e la configurazione di HPA e DCO ed è configurato con tre partizioni primarie, la prima NTFS è la partizione di sistema Win2000, una seconda partizione primaria è formattata in FAT32 ed una terza *mini-partizione* primaria formattata in NTFS alla fine del disco. Dall'output di *mmls*, in particolare, è possibile osservare che oltre al *Master Boot Record (Primary*

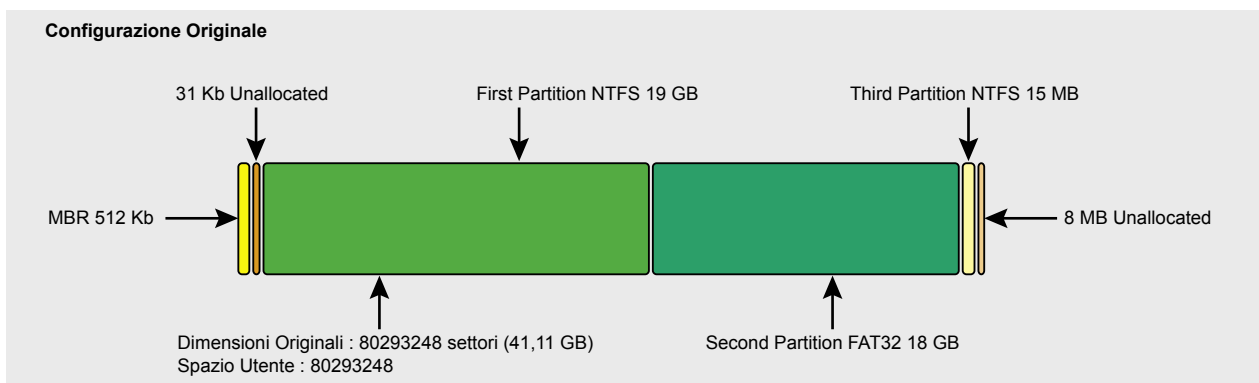


Fig. 1: Configurazione base del disco

Table) e ad un piccolo spazio non allocato dal settore 1 al settore 62, sono presenti 8 MB di spazio alla fine del disco (sett. 80276805-80293247) lasciati dall'installazione di Windows 2000 per un'eventuale successiva conversione del disco in dinamico. In Figura 1 è visibile uno schema di configurazione del disco.

Progetto

Il nostro scopo è quello di accorciare la parte utente (settori del disco utilizzabili da impostazioni di fabbrica) del disco con qualche colpo di bacchetta magica ed uno o due comandi ad hoc. Creeremo prima una *Device Configuration Overlay* in modo da far scomparire l'ultima partizione e lo spazio non allocato al termine del disco, setteremo in seguito una *Host Protected Area* che dimezzerà in pratica la capacità del disco.

Tali modifiche non devono risultare distruttive per i dati contenuti nelle rispettive partizioni che devono essere recuperabili attraverso le successive fasi di annullamento delle operazioni o attraverso un accesso a basso livello ai relativi settori del disco.

Leggere il Master Boot Record

Le strade per nascondere una parte di hard disk attraverso modifiche ATA sono fondamentalmente due. La prima, quella facile, consiste nell'eseguire

le modifiche su un supporto vergine, per poi installare comodamente il sistema operativo sulla parte utente visibile. La seconda, quella più complicata, avviene a disco già configurato, modificando il *Master Boot Record* ed i record di avvio di ogni partizione in modo che riflettano le modifiche eseguite a basso livello, evitando così problemi al sistema operativo. Ovviamente sceglieremo la strada più complicata, lasciando come unica concessione alla semplicità l'accortezza di prendere ad esempio un disco contenente solo partizioni primarie.

Struttura

Il *Master Boot Record* risiede nei primi 512 byte del disco e consiste in una parte di codice, richiamata dopo il POST, preposta al caricamento del sistema operativo ed in una serie di record che specificano le partizioni del disco ed i relativi attributi, come dimensioni, tipo ed indirizzi di inizio e fine settore.

L'implementazione del codice presente nel *Master Boot Record* varia a seconda del sistema operativo utilizzato, non esistendo uno standard univoco per la codifica di questo settore. Nonostante ciò i produttori di OS sembrano seguire linee guida comuni per quanto riguarda la mappatura delle partizioni e i parametri relativi ed in particolare per la firma 55AA presente alla fine del settore che lo identifica inequivocabilmente come settore di boot.

Nella Figura 2 è presente il dump del *Master Boot Record* in questione, creato da un'installazione standard di Windows 2000. Possiamo visualizzare il settore di boot direttamente dalla shell:

```
# dd if=/dev/hda bs=512 count=1 | xxd
```

La prima parte (dal byte 0 al byte 445) contiene il codice che si occupa di caricare la tabella delle partizioni e lanciare il sistema operativo.

Dal byte 446 inizia la tabella delle partizioni, che analizzeremo in dettaglio e modificheremo manualmente per rispecchiare le modifiche del ATA che apporteremo al disco.

Gli ultimi due byte contengono la signature 55AA che identifica il settore di boot.

Tabella delle partizioni

Le caratteristiche delle partizioni sono inserite in quattro gruppi di 16 byte a partire dal byte 446. Analizziamo la prima partizione che inizia nel byte 0x1BE e termina in 0x1CD:

- il byte 0 (0x1BE) contiene il flag di boot, settato a 0x80 in quanto partizione di boot,
- i byte 1-3 (0x1BF-0x1C1) il settore di inizio in formato CHS,
- nel byte 4 (0x1C2) il tipo di partizione (07 = NTFS),
- dal byte 5 al 7 (0x1C3-0x1C5) l'indirizzo di fine partizione in CHS,
- i byte dall'8 all'11 (0x1C6 – 0x1C9) l'indirizzo di inizio partizione LBA,
- dal 12 al 15 (0x1CA – 0x1CD) la grandezza della partizione in settori.

La figura 3 mostra in dettaglio la parte del *Master Boot Record* contenente le specifiche delle partizioni.

Entrando nel sistema Windows 2000 contenuto nella prima partizione, il disk manager mostra le partizioni visibili in Figura 3a.

Salvataggio del Master Boot Record

Prima di effettuare operazioni potenzialmente distruttive è sempre buona norma salvare i 512 byte iniziali in

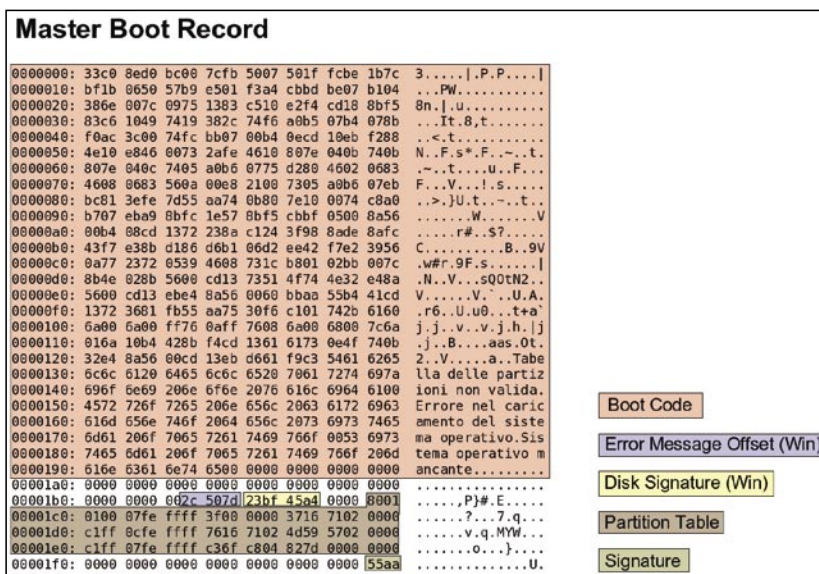


Fig. 2: Master Boot Record

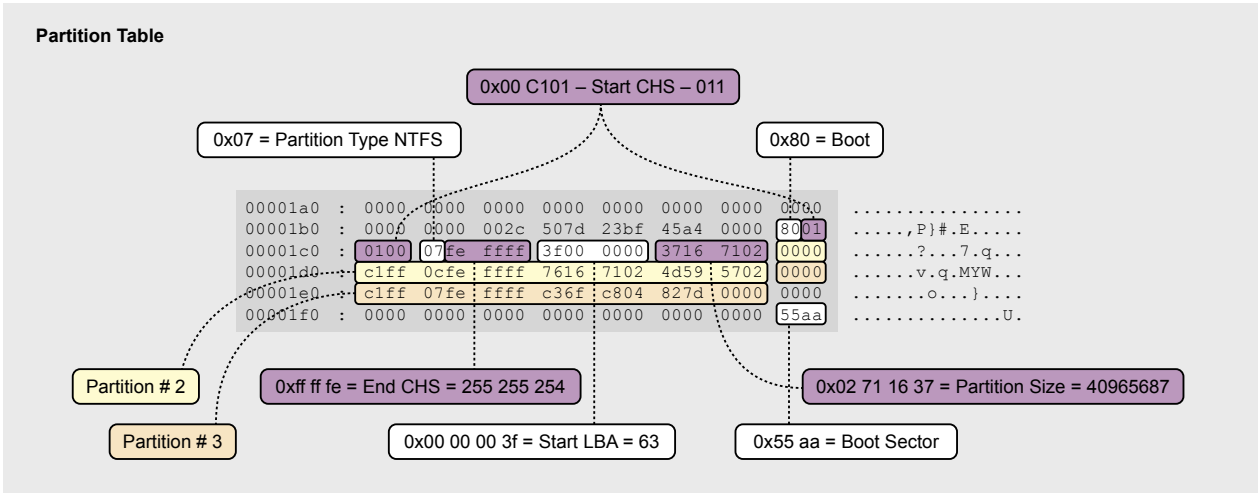


Fig. 3: Tabella delle Partizioni

modo da poter ripristinare il MBR in qualsiasi momento.

```
# dd if=/dev/hda bs=512 count=1 of=mbr_hda_orig.dd
```

L'immagine del MBR originale verrà utilizzata dopo aver rimosso le aree inaccessibili, per riportare il disco allo stato originale.

Utilizzare lo spazio non allocato

Il sistema più semplice e meno efficace di nascondere dati all'esterno del filesystem è utilizzare lo spazio non allocato del disco. Oltre ad essere automaticamente incluso nelle immagini ottenute con le corrette procedure forensi, lo spazio non

allocato può essere in ogni caso raggiunto dai programmi che accedono direttamente ai settori del disco.

È possibile, ad esempio, utilizzare lo spazio non allocato di 31 Kb dal settore 1 al settore 62 per nascondere un file di testo, uno script o un piccolo programma.

Così come possiamo nascondere un file nello spazio non allocato dal sistema operativo:

```
# dd if=/etc/passwd bs=512 seek=1 of=/dev/hda
```

È però possibile allo stesso modo scoprire le informazioni in esso contenute

```
# dd if=/dev/hda count=4 skip=1 | less
```

L'efficacia di questo metodo per nascondere informazioni importanti è quindi molto limitata.

I comandi ATA

Prima di procedere alla modifica del disco a basso livello è bene riassumere alcuni comandi e caratteristiche ATA riguardanti le dimensioni del disco.

La capacità massima reale dell'hard disk (*Native Max Address*), ovvero il massimo settore indirizzabile, si ottiene attraverso i comandi `READ NATIVE MAX ADDRESS` e `READ NATIVE MAX ADDRESS EXT`.

I comandi che terminano in `EXT` si riferiscono ai dischi che supportano l'indirizzamento LBA a 48 bit (capacità fino a 128 Petabytes), per semplicità prenderemo in considerazione unicamente i comandi con indirizzamento LBA a 28 bit (dischi fino a 128 GB).

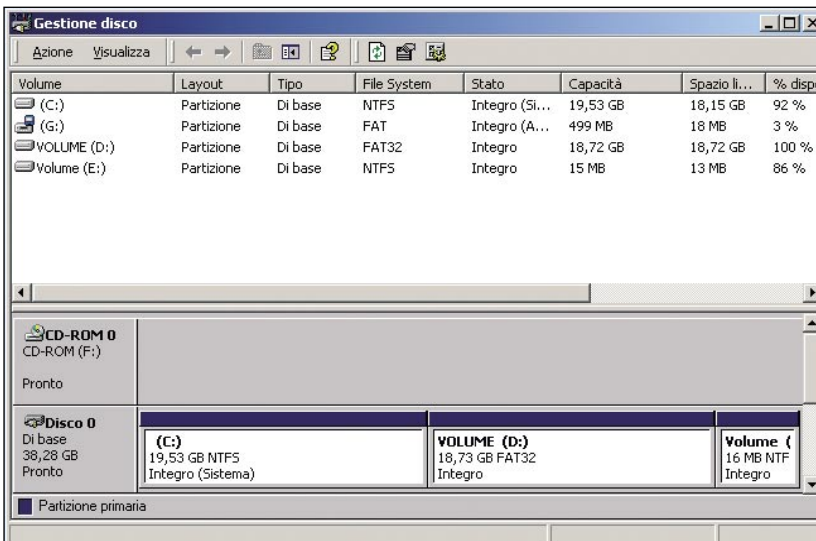


Fig. 3a: Disk Manager – Situazione Originale



Fig. 4: Etichetta Hard Disk

Il comando `IDENTIFY DEVICE` riporta una serie di informazioni sul disco, tra le quali le varie caratteristiche supportate ed i settori indirizzabili. La differenza sostanziale è che mentre `READ NATIVE MAX ADDRESS` riporta i settori reali del disco, `IDENTIFY DEVICE` riporta un parametro modificabile attraverso appositi comandi che vedremo nella configurazione della *Host Protected Area*.

Non è finita. Il comando `DEVICE CONFIGURATION SET` (DCO) può attivare o disattivare alcune caratteristiche del disco e modificare la massima dimensione LBA dello stesso, facendo in modo che anche con il comando `READ NATIVE MAX ADDRESS`, sia riportato un numero di settori raggiungibili limitato rispetto all'originale.

Lasciando da parte i comandi ATA, due metodi pratici ed efficaci per ricavare la reale capacità del disco consistono nel prendere nota di marca e modello dell'hard disk per controllarne le caratteristiche sul sito del produttore e nell'estrarre fisicamente il drive osservando

sull'etichetta il numero max LBA e le dimensioni in GB riportate correttamente sulla maggioranza degli hard disk. In Figura 4 è riportata l'etichetta del disco con evidenziato l'indirizzo max LBA di fabbrica.

Tornando all'acquisizione della reale capacità del disco attraverso gli standard ATA, si presenta quindi una sorta di gerarchia di comandi, tra i quali `DEVICE CONFIGURATION SET` ricopre la posizione dominante.

Ecco perché la prima modifica che andremo a fare sarà la creazione dell'area DCO. Riassumendo, ecco come varieranno i settori utilizzabili a seconda dei comandi impartiti:

- `SETTORI REALI`: 80293248 settori (41,10 GB),
- `DEVICE CONFIGURATION SET`: (DCO): 80244674 settori (41,08 GB),
- `SET MAX ADDRESS`: (HPA): 40965751 settori (20,97 GB).

La Figura 5 mostra i passaggi della configurazione:

Device Configuration Overlay

Introdotta con le specifiche ATA/ATAPI 6, il DCO interagisce con una serie di comandi e caratteristiche modificando così i parametri del disco riportati:

- `DEVICE CONFIGURATION IDENTIFY` – Mostra in dettaglio le caratteristiche selezionabili e la reale grandezza del disco,
- `DEVICE CONFIGURATION SET` – Permette di modificare i comandi e le modalità del disco, compresa la grandezza,
- `DEVICE CONFIGURATION RESTORE` – Rimuove ogni modifica e ritorna ai parametri originali,
- `DEVICE CONFIGURATION FREEZE LOCK` – Blocca le modifiche alla configurazione DCO.

La nostra prima operazione consiste nel nascondere la partizione NTFS da 16 MB e lo spazio non allocato al termine dello stesso (8MB) attraverso `DEVICE CONFIGURATION SET`.

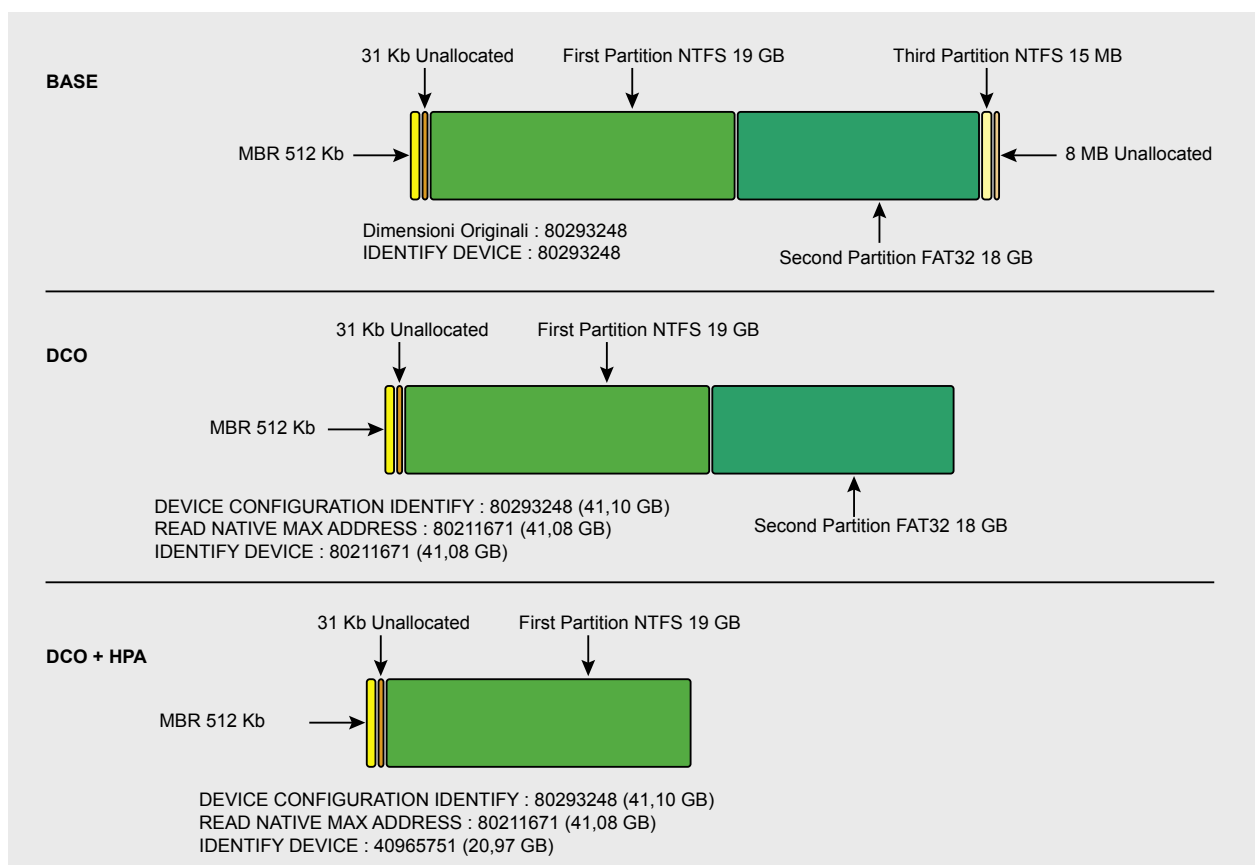


Fig. 5: Fasi di Configurazione

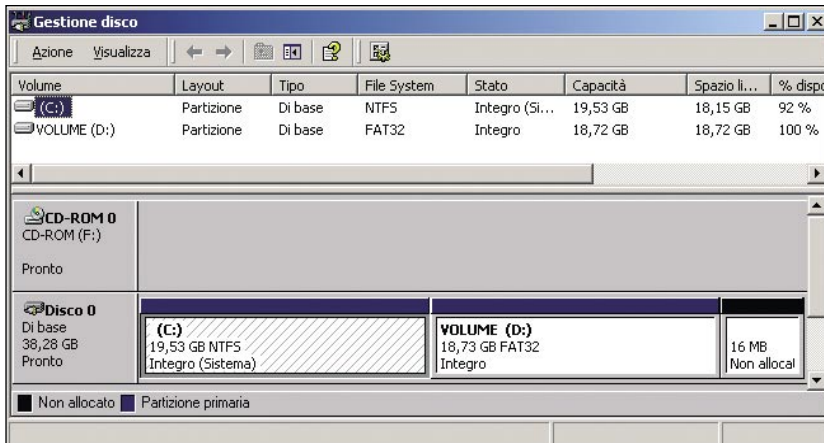


Fig. 6: Disk Manager dopo modifica DCO

Come abbiamo visto dalla precedente analisi il terzo record ha inizio nel byte 0x01DE e termina nel byte 0x1ED. Salviamolo con `dd` settando ad un settore la dimensione dei blocchi (`bs=1`), a 478 il blocco di partenza (`skip=478`) ed ovviamente a 16 il numero dei byte da copiare (`count=16`):

```
dd if=/dev/hda bs=1 skip=478
count=16 of=/mnt/floppy/part3.img
```

Ora possiamo eliminare la partizione portando a zero i byte relativi. Il parametro di offset in output di `dd` è `seek` (rispetto a `skip` in `input`):

```
dd if=/dev/zero bs=1 count=16
seek=478 of=/dev/hda
```

Eseguendo il riavvio della macchina ed entrando in Windows possiamo constatare come il disk manager veda come libero lo spazio al termine del disco (Figura 6). Eseguiamo un ulteriore riavvio dopo aver inserito il floppy di `hdat2`.

Entrati nell'ambiente del programma, la prima schermata (*Device List*) consiste in una lista dei device da selezionare. Selezionato l'hard disk in causa, entriamo nel menu *Device Configuration Overlay Menu*.

Selezionando *Modify* entriamo direttamente nell'ambiente di modifica DCO. Per ridurre la dimensione del disco dobbiamo editare il parametro *Maximum LBA sectors*, impostando come ultimo settore quello finale della seconda partizione. Dall'output di `fdisk` e `mmls` abbiamo visto che la se-

conda partizione termina nel settore 80276804, inseriamolo quindi come nuovo valore (*INS*) al posto del massimo settore originale. Premiamo `S` per rendere effettive le modifiche e confermiamo.

Al riavvio troveremo un hard disk sottodimensionato. Già a livello del BIOS è possibile osservare che la grandezza del disco è variata. Avendo modificato la tabella delle partizioni nel *Master Boot Record*, Windows 2000 vedrà due partizioni sane, senza alcuno spazio libero (Figura 7).

È possibile ottenere la conferma finale eseguendo il *boot* da *hakin9 live*, eseguendo `fdisk`, `hdparm` e `mmls` (nel Listato 2 l'output di `fdisk` e `mmls` dopo la modifica DCO).

Programmi dedicati all'analisi forense quali *EnCase* e *X-Ways*

Forensics rilevano in ogni caso la discrepanza tra la dimensione effettiva e quella utilizzabile (in Figura 8 *X-Ways* mostra il messaggio relativo).

SleuthKit non supporta al momento il rilevamento di modifiche DCO ma la caratteristica è in fase di implementazione da parte dell'autore (Brian Carrier).

Una modifica di questo tipo, essendo contenuta in termini di spazio sottratto a quello effettivo, passa solitamente inosservata.

Host Protected Area

La modifica successiva, consistente nella creazione di una *Host Protected Area*, è più drastica e facilmente rilevabile, visto che verrà completamente nascosta la seconda partizione e la dimensione del disco sarà praticamente dimezzata.

Le *Host Protected Area* sono introdotte dagli standard ATA / ATAPI 4 e consistono in particolari zone dell'hard disk nascoste al sistema operativo ed al filesystem. L'utilizzo tradizionale di queste aree è lo storage di dati o applicazioni di ripristino del sistema, immagini ridotte di sistemi operativi richiamati dalle utility di boot di produttori ecc... È però possibile utilizzare la stessa caratteristica per creare aree inaccessibili attraverso il normale utilizzo del sistema, al fine di nascondere dati più o meno legali.

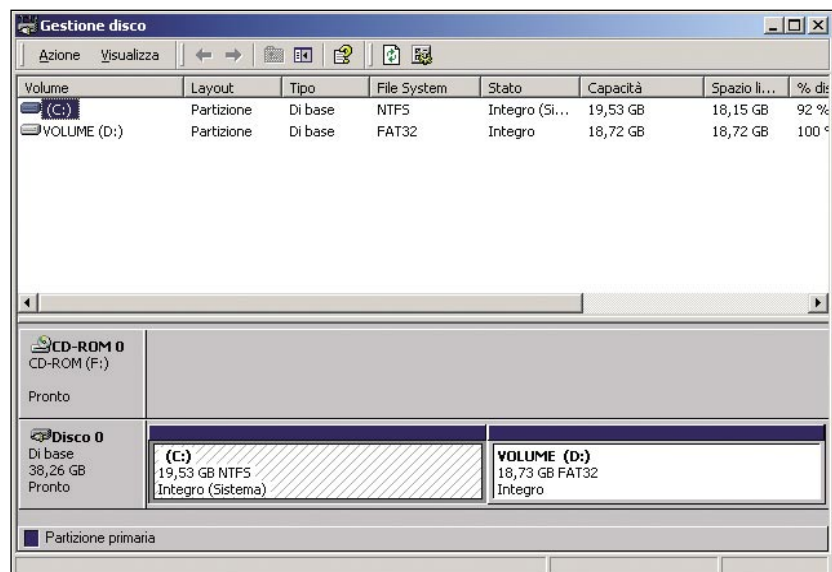


Fig. 7: Disk Manager dopo DCO

Come per il DCO, andiamo prima ad analizzare i comandi ATA riguardanti le *Host Protected Area*.

- `READ NATIVE MAX ADDRESS` – Riporta la quantità massima originale di settori utilizzabili (tranne in caso di DCO),
- `SET MAX ADDRESS` – Permette di modificare il numero dei settori accessibili a livello utente,
- `SET MAX SET PASSWORD` – Imposta una password non resistente al riavvio,
- `SET MAX LOCK` – Disabilita i comandi `SET MAX` eccetto `SET MAX UNLOCK`,
- `SET MAX UNLOCK` – Riabilita i comandi `SET MAX`,
- `SET MAX FREEZE LOCK` – Disabilita tutti i comandi `SET MAX` compreso `UNLOCK`.

Utilizzeremo il comando `SET MAX ADDRESS` per reimpostare il numero massimo di settori utilizzabili, come riportato da `DEVICE IDENTIFY` e nascondere quindi completamente anche la partizione FAT da 18 GB.

Il primo passo consiste, come in precedenza, nella modifica della tabella delle partizioni in modo da non lasciare traccia della seconda partizione che, osservando i precedenti output di `fdisk` e `mmls`, si estende dal settore 0040965750 al settore 0080244674.

Creiamo un backup del record della seconda partizione che, come possiamo osservare va dal byte 462 al byte 478:

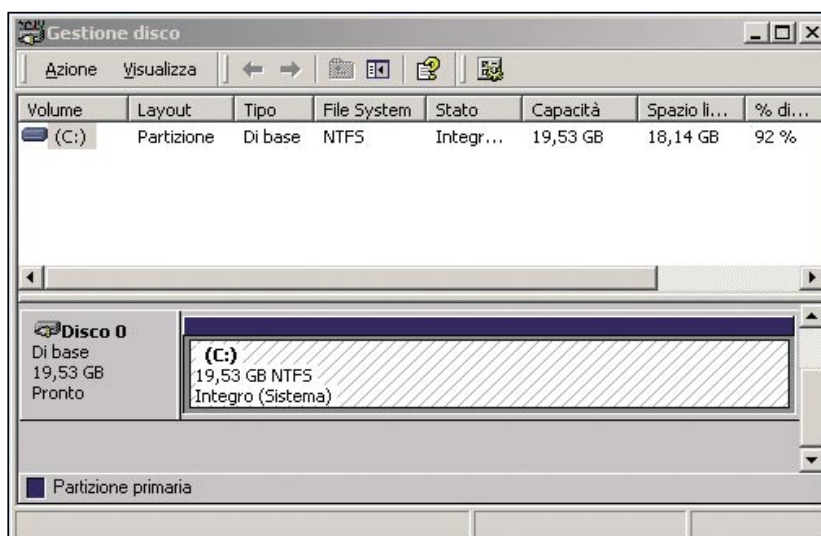


Fig. 9: Disk Manager dopo modifica HPA

```
dd if=/dev/hda bs=1 count=16
skip=462 of=/mnt/floppy/part3.img
```

quindi un bel colpo di spugna:

```
dd if=/dev/zero bs=1 count=16
seek=462 of=/dev/hda
```

Impostazione della Host Protected Area

Eseguiamo l'avvio del sistema con `hdatt2` e selezioniamo il menu `Set Max Address` all'interno della sezione `SET MAX (HPA)`.

Osservando i parametri `NATIVE MAX ADDRESS (max. address of sectors)`:

```
Native area: 80244675 sectors=41.08 GB
User area: 80244675 sectors=41.08 GB
```

È evidente come il comando `READ NATIVE MAX ADDRESS` non veda la porzione di disco nascosta in DCO.

Impostiamo quindi come nuovo valore (INS) il limite della prima partizione, 40965749, modificando la grandezza del disco per l'utente in 20,97 GB. Inseriamo il dato con `Invio` e confermiamo con `S`.

Al riavvio i disk manager di Windows 2000 mostrerà un disco da 20,97 GB (Figura 9). E in Linux?

Kernel 2.6.x e HPA

Eseguito il riavvio con `hakin9 live` `fdisk` mostra sempre un disco di 41 Gb e la partizione NTFS, `mmls` mostra lo spazio non allocato dove una volta era la partizione FAT. Il mistero è svelato attraverso l'esecuzione di `dmesg | grep hda`:

```
hda: Host Protected Area detected
hda: Host Protected Area disabled
hda: 80244675 sectors (41085 MB).....
```

Il kernel 2.6 rileva e disabilita in automatico le *Host Protected Area*, a meno che il BIOS impedisca l'accesso ai comandi `SET MAX`. Questo comportamento ha generato problemi con alcuni controller RAID e con le funzioni di sospensione dei laptop e sembra che nelle future versioni la disabilitazione delle HPA verrà resa facoltativa.

Montare la partizione in HPA

Visto che il kernel ha gentilmente disabilitato la *Host Protected Area*, approfittiamone per montarla e vedere cosa rimane all'interno.

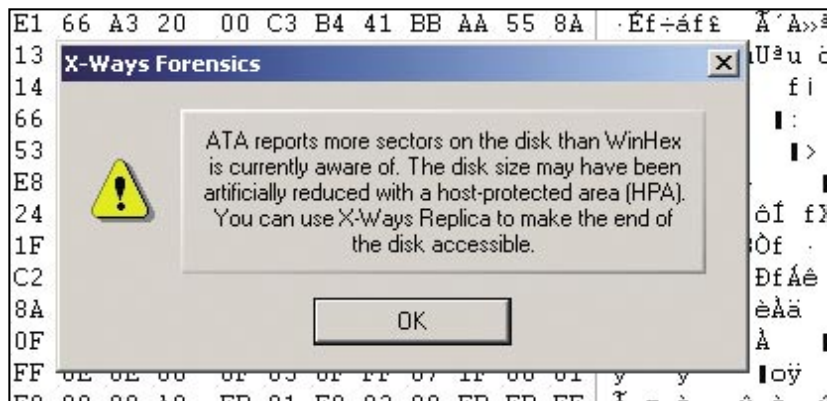


Fig. 8: Avviso di X-Way Foresics

Conosciamo il settore di partenza della seconda partizione: 40965750. Moltiplichiamolo per 512 (byte per settore) ed otterremo l'offset in byte da passare a mount e losetup:

```
# FAT2=$((40965750*512))
# mount -ro loop,offset=$FAT2 /dev/hda
/mnt/hda2
```

Un giretto nella cartella appena montata mostrerà i file intatti presenti prima della modifica della *Host Protected Area*.

Rilevamento HPA

Come per il DCO, la maggior parte dei programmi di *Computer Forensics* è in grado di rilevare e duplicare una *Host Protected Area*.

Anche *SleuthKit*, attraverso il tool `disk_stat`, è in grado di rilevare una HPA:

```
# disk_stat /dev/hda
Maximum Disk Sector: 80244674
Maximum User Sector: 40965749
** HPA Detected (Sectors 40965750
- 80244674) **
```

Ovviamente resta valido il sistema di confrontare i settori accessibili con le caratteristiche riportate sull'etichetta del disco o sul sito del produttore.

Ripristino del disco

Procedendo al contrario è possibile riportare il disco alle condizioni iniziali.

Il primo passo consiste nel ripristinare il *Master Boot Record* attraverso l'immagine salvata in precedenza:

```
dd if=mbr_hda_orig bs=512 count=1
of=/dev/hda
```

Quindi con `hdparm` eliminiamo prima la *Host Protected Area* riportando al valore massimo (80244675) il parametro *New Native Max Address* all'interno della sezione *Set Max Address* e reimpostiamo il DCO in modo che il *Maximum LBA Sectors* all'interno di *DCO Modify* rispecchi la reale dimensione del disco (80293247).

P-List e G-List

Il seguente è un sistema più complicato, sia per quanto riguarda l'occultamento sia per quanto riguarda il recupero dei dati, legato profondamente all'hardware ed alle caratteristiche implementate dai produttori. Nonostante ciò è una possibilità che soprattutto a livello forense dev'essere tenuta in debita considerazione.

Il sistema si basa sulle liste di settori difettosi e sulle aree di settori di riserva che i produttori di *Hard Disk* utilizzano per correggere eventuali errori, sia in fase di produzione che di utilizzo del disco.

In pratica al rilevamento di un settore danneggiato nella parte utilizzabile del disco, lo stesso settore viene rimpiazzato da uno spare contenuto in un apposito spazio esterno e la transazione viene registrata in apposite liste di errore.

Esistono due liste preposte a scopi differenti: la *P-List* o *Primary Defect List* viene utilizzata per i settori danneggiati eventualmente emersi dalle operazioni di test prima della messa

in vendita del supporto e la *G-List* o *Grown Defect List* che, al contrario viene utilizzata durante il ciclo di vita dell'apparato.

A differenza delle modalità di marcatura a livello di filesystem, dove i settori danneggiati vengono segnalati come inutilizzabili dal sistema operativo rimanendo comunque nella catena dei settori, le Defect List lavorano ad un livello più basso e permettono la sostituzione del settore prima del passaggio al controller del disco.

Il settore danneggiato durante il normale utilizzo del disco viene aggiunto alla tabella *G-List* e viene assegnato il relativo indirizzo LBA ad un settore di riserva. In pratica il settore corrotto fisicamente rimane nella stessa locazione ma, ad una richiesta del controller indirizzata al settore danneggiato, la testina viene indirizzata fisicamente sul settore spare che ne ha ereditato l'indirizzo LBA. La Figura 10 mostra uno schema logico del sistema:

A seconda del sistema utilizzato dal produttore i settori danneggiati vengono aggiunti automaticamente durante fasi di *self-test* o manualmente attraverso appositi comandi, nella maggior parte dei casi senza modificarne il contenuto. In teoria la *P-List* dovrebbe essere immutabile durante il normale utilizzo. Nella realtà è comunque possibile modificare la lista con alcuni programmi dedicati.

Ovviamente lo spazio riservato alle *Defect List* è limitato, variabile a seconda di produttore, modello e tecnologia utilizzata, non consen-

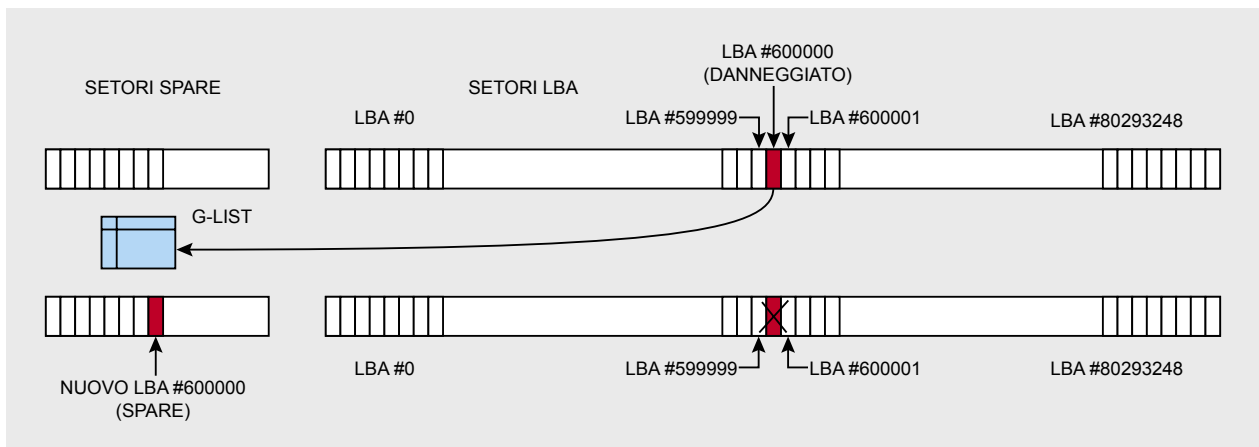


Fig. 10: Schema logico di sostituzione del settore danneggiato

Tabella 1. Struttura di Sicurezza ATA

Utente	Livello	Operazioni
Master	Max	NON può sbloccare il disco (SECURITY UNLOCK) Può unicamente cambiare la password utente dopo la formattazione sicura del drive e la reimpostazione della sicurezza del disco
Master	High	Può sbloccare il disco (SECURITY UNLOCK) Può disabilitare la password (SECURITY DISABLE PASSWORD)
User	Max	Può disabilitare la password (SECURITY DISABLE PASSWORD) Può sbloccare il disco (SECURITY UNLOCK)
User	High	Può disabilitare la password (SECURITY DISABLE PASSWORD) Può sbloccare il disco (SECURITY UNLOCK)

tendo quindi l'utilizzo delle stesse per nascondere una grande quantità di dati.

Marcatura di un settore

Per fare un esempio di sostituzione di settore danneggiato utilizzeremo un programma simile ad hdat2 ed altrettanto gratuito quale MHDD prelevabile da <http://files.hddguru.com/index.html>.

Per prima cosa selezioniamo tre settori contigui non utilizzati dal sistema operativo. Per questo esempio utilizzeremo i tre settori 599999, 600000 e 600001, li contrassegneremo con una stringa per renderli identificabili, quindi marcheremo il settore 600000 come danneggiato. Per prima cosa azzeriamo il contenuto dei settori:

```
# dd if=/dev/zero bs=512 count=3
seek=599999 of=/dev/hdc
```

Quindi inseriamo nei settori dei patterns identificativi:

```
perl -e "print '#SETTORE 599999#' x 32"
| dd of=/dev/hdc seek=599999
count=1
perl -e "print 'IL MIO TESSSSORO' x 32"
| dd of=/dev/hdc seek=600000
count=1
perl -e "print '#SETTORE 600001#' x 32"
| dd of=/dev/hdc seek=600001
count=1
```

La Figura 11 mostra il contenuto dei tre settori contrassegnati estratti attraverso il comando:

```
# dd if=/dev/hda bs=512 count=3
skip=599999 | xxd
```

Avviamo il sistema con il floppy di MHDD inserito ed utilizziamo il comando MAKEBAD, che permette di marcare uno o più settori come difettosi.

Digitiamo y alla prima richiesta di conferma quindi impostiamo lo stesso settore come inizio e fine (600000):

```
Continue? (y/N)
1 block= 1 sector (slow mode)
Type start sector to write
(from 0) [0]:
Type end sector [0]:
Start   : 600000
End     : 600000
Continue? (y/N)
```

Confermiamo un'ultima volta usciamo dal programma, estraiamo il floppy ed eseguiamo il *reboot* della stazione con il cd di *hakin9 live* inserito. L'output del comando:

```
# dd if=/dev/hda bs=512 count=3
skip=599999 | xxd
```

Mostra come il settore 600000 sia completamente vuoto, mentre i set-

```
Before MAKEBAD

0000000: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000010: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000020: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000030: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
.....
0000200: 494c 204d 494f 2054 4553 5353 534f 524f IL MIO TESSSSORO
0000210: 494c 204d 494f 2054 4553 5353 534f 524f IL MIO TESSSSORO
0000220: 494c 204d 494f 2054 4553 5353 534f 524f IL MIO TESSSSORO
0000230: 494c 204d 494f 2054 4553 5353 534f 524f IL MIO TESSSSORO
.....
0000400: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000410: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000420: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000430: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
.....
```

Fig. 11: Settori prima di MAKEBAD

```
After MAKEBAD

0000000: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000010: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000020: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
0000030: 2353 4554 544f 5245 2035 3939 3939 3923 #SETTORE 599999#
.....
0000200: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000210: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000220: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000230: 0000 0000 0000 0000 0000 0000 0000 0000 .....
.....
0000400: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000410: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000420: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
0000430: 2353 4554 544f 5245 2036 3030 3030 3123 #SETTORE 600001#
.....
```

Fig. 12: Settori dopo MAKEBAD



tori contigui hanno mantenuto il proprio contrassegno (Figura 12).

Il settore 600000, apparentemente cancellato, è stato in realtà marcato come UNC (*Uncorrectable Data Error*) e presumibilmente aggiunto alla *G-List*, forzando una rimappatura dell'indirizzo LBA relativo con un settore spare.

Il *presumibilmente* è dovuto al comportamento non standardizzato degli hard disk, variabile a seconda del produttore, della tecnologia e dello stesso modello del disco. La rimappatura e l'inserimento nella defect list può avvenire infatti in automatico, a seguito di determinate istruzioni o dopo una formattazione a basso livello ed un remap completo dello spazio utente.

Per riutilizzare il settore finito nella *G-List* senza cancellarne il contenuto è necessaria una notevole esperienza nella struttura fisica degli hard disk ed un programma apposito che utilizzi le caratteristiche proprie del disco in questione, ovvero conosca i comandi specifici del produttore ed il layout delle aree precluse dell'hard disk. Uno dei pochi programmi che sembrano poter effettuare simili operazioni in combinazione con apposite schede hardware è pc3000 <http://www.pc3000.com/>.

Altra area nella quale è virtualmente possibile inserire dati è la (*System Area*), parte di hard disk all'interno della quale sono inseriti moduli contenenti firmware, parti di codice, informazioni e caratteristiche del supporto e spesso le stesse *Defect Lists*. Solitamente in queste aree esiste spazio non allocato utilizzato per aggiornamenti del firmware o la modifica di caratteristiche, virtualmente occupabile, con debite conoscenze e strumenti, da informazioni aggiuntive.

Hard Disk Password

Gli standard ATA 3 introducono specifiche di sicurezza con le quali è possibile limitare l'accesso ai dati presenti in area utente. Il sistema si basa sull'impostazione di due password che unitamente ad altrettante modalità di sicurezza consentono l'accesso al disco alle sole persone autorizzate, e su una procedura di cancellazione sicura del drive.

I comandi da prendere in considerazione sono in questo caso i seguenti:

- SECURITY SET PASSWORD – Imposta le password *Master* e *User*, imposta il Livello di sicurezza del disco,
- SECURITY DISBLE PASSWORD – Disabilita lo stato di *Lock* del disco ma non modifica la *Master Password*,
- SECURITY UNLOCK – Sblocca la password (fino al prossimo riavvio),
- SECURITY FREEZE LOCK – Imposta il device in modalità *Frozen*, impedendo ogni altro comando *Lock* fino al successivo riavvio,
- SECURITY ERASE PREPARE – Preparazione al comando *Erase*. Deve essere obbligatoriamente passato prima del comando SECURITY ERASE,
- SECURITY ERASE UNIT – Sovrascrive con degli zero o dei pattern specificati (in *Enhanced Mode*) i settori dell'area utente.

Struttura di Sicurezza ATA

Le specifiche di sicurezza prevedono due livelli di password, *Master* e *User* abbinati a due livelli di sicurezza: *Maximum* e *High*.

Una volta impostata, la password *User* abilita la sicurezza del disco e viene richiesta dal BIOS ad ogni avvio del computer. Al contrario l'inserimento della password *Master* non abilita di default la sicurezza del disco ed alcuni produttori impostano una password *Master* di default. L'utilizzo principale della password *Master* è quello di fornire una modalità di riserva per l'accesso ai dati in caso di smarrimento della password utente.

Con il livello *Maximum* impostato la password *Master* non può sbloccare il disco e per disabilitare la funzionalità di sicurezza deve obbligatoriamente utilizzare il comando SECURITY ERASE UNIT, cancellando quindi tutto il contenuto del disco. Questa procedura permette all'utente di conservare la segretezza dei dati anche nei confronti del detentore della *Master password*.

La Tabella 1 riassume le combinazioni password/livello di sicurezza.

Una volta bloccato il disco permette l'esecuzione esclusiva di un subset di comandi ata, quali IDENTIFY DEVICE ed il comando SECURITY UNLOCK DEVICE.

Sia *HDAT2* che *MHDD* permettono le impostazioni di sicurezza. In *HDAT2* la configurazione avviene all'interno del *Security Menu*, attraverso l'opzione *Set Password*. Con il parametro *Identifier* si imposta l'utente mentre con *Password Level* il livello di sicurezza.

Analisi Forense

Nonostante le funzionalità di sicurezza ATA non prevedano la crittografia dei dati, l'analisi di un disco protetto da una password ATA può rivelarsi problematica. Le caratteristiche da tenere in considerazione sono le seguenti:

- La tecnica del *brute-force* è praticamente inapplicabile a meno di possedere un set minimo di alternative sicure da provare (es. eventuali password di sistema o applicative appartenute al-

Terminologia

- DCO – *Device Configuration Overlay*, caratteristiche modificanti i parametri originali del disco,
- HPA – *Host Protected Area*, aree esterne allo spazio utente comunemente non visibili dal sistema operativo,
- P-List – *Primary Defect List*, lista dei settori danneggiati rilevati dal produttore durante le fasi di test,
- G-List – *Grown Defect List*, lista dei settori danneggiati durante il normale utilizzo del disco,
- SA (*System Area*) – Parte dell'Hard Disk esterna all'area utente contenente firmware, parti di codice e caratteristiche del disco.

In Rete

- <http://www.hdat2.com> – Tool HDAT2 – Documentazione interessante su HD Recovery.
- <http://files.hddguru.com/index.html> – Tool MHDD.
- <http://sleuthkit.org> – Sleuthkit e Autopsy, Open Source Computer Forensics Software.
- <http://www.pc3000.com> – Software ed Hardware per interventi a basso livello su HD.
- <http://t13.org/> – Comitato di sviluppo specifiche T13 AT Attachment.
- <http://www.ata-atapi.com/index.htm> – Information for *Developers of Products Using ATA (PATA, IDE/EIDE), Serial ATA (SATA), ATAPI, CF, CE-ATA and Other ATA Related Interfaces.*

l'utente del disco). Le implementazioni standard infatti prevedono un contatore che limita a cinque i possibili tentativi, terminati i quali diventa obbligatorio il riavvio del sistema.

- Alcuni produttori impostano una *Master password* di default, utile unicamente quando il livello di sicurezza è settato ad *High*. Si può capire se la password Master è stata modificata dal *Master Password Revision Code* (se supportato dal device), un codice di riferimento che come valore di fabbrica ha il valore 0xFFFE. Tale valore viene diminuito ad ogni cambio di password.
- Ad oggi sembra non esistano tool che permettano di bypassare in maniera semplice la password, alcune aziende vendono tuttavia dei servizi (da verificare) di rimozione delle password ATA dal disco.
- L'accesso al disco attraverso un tool di inserimento della password su un sistema con BIOS differente da quello di origine, può comportare l'impossibilità di sbloccare il disco pur conoscendo la password corretta del-

l'utente, a causa degli algoritmi di hashing che alcuni BIOS applicano alla password prima di passarla al disco.

- Non esistendo crittografia è possibile risalire ai dati fisicamente, aprendo il disco in camera sterile ed effettuando il recupero con le apposite apparecchiature.
- In alcuni casi la sostituzione del firmware del disco può causare la cancellazione dei parametri permettendo l'accesso ai dati.

Riassunto

L'articolo ha mostrato alcune metodologie di occultamento dei dati utilizzando le caratteristiche a basso livello degli hard disk. Tali sistemi rappresentano possibili fonti di problemi per l'analista forense che non ne comprenda appieno le caratteristiche. Ulteriori difficoltà sono poi dovute al fatto che la difformità delle caratteristiche di implementazione degli standard ATA utilizzate dai produttori rende difficile la codifica di metodologie univoche e lo sviluppo di software appropriati. Come se ciò non bastasse, la mediazione di BIOS dal comportamento differente complica ulteriormente l'ambiente di analisi.

Il testo di base per chi sia interessato ad approfondire gli argomenti è il libro di Brian Carrier (lo sviluppatore di *SleuthKit*) *File System Forensics Analysis* che, oltre a fornire una visione approfondita dei principali filesystem, comprende una parte estesa sulla struttura di Hard Disk, partizioni e volumi. ●

Cenni sull'autore

L'autore si occupa di computer forensics ed investigazione digitale, gestisce il sito dedicato all'argomento www.Cybercrimes.it e lavora come amministratore di rete e sicurezza presso una compagnia multiservizi del Nord Italia.



Sul nostro sito troverete:
materiali per gli
articoli - i listing,
documentazione aggiuntiva,
strumenti utili,
gli articoli più
interessanti da scaricare,
le attualità,
informazioni sui numeri
in arrivo