



CYBERCRIMES.IT

Computer Forensics & Crimine Informatico

Maurizio Anconelli

OS X come piattaforma di analisi : Diskarbitrationd

Con la crescente diffusione sui desktop di giovani e professionisti, il sistema operativo della mela morsicata inizia a rivestire un ruolo di primaria importanza in Computer Forensics, sia come sistema indagato che come piattaforma d'analisi. In quest'ultimo caso, una delle prime caratteristiche da padroneggiare è il sistema di gestione dei dischi, in particolare come sia possibile disabilitare il mount automatico di supporti esterni per evitare modifiche sui media da analizzare.

Il sistema illustrato non sostituisce l'utilizzo di un write-blocker, in quanto rappresenta pur sempre una sicurezza software. In ogni caso, come vedremo, dai test effettuati non emerge alcuna modifica dei supporti di prova, ed otterremo una sicurezza operativa paragonabile a quella delle maggiori distribuzioni forensi di base "Linuxiana".

Disk Arbitration Framework

Le operazioni da eseguire per disabilitare il mount automatico dei dischi si riassumono in due comandi da eseguire con i privilegi di root. Come da precetto in Digital Forensics, è tuttavia importante avere ben chiari i meccanismi che sottendono le procedure stesse.

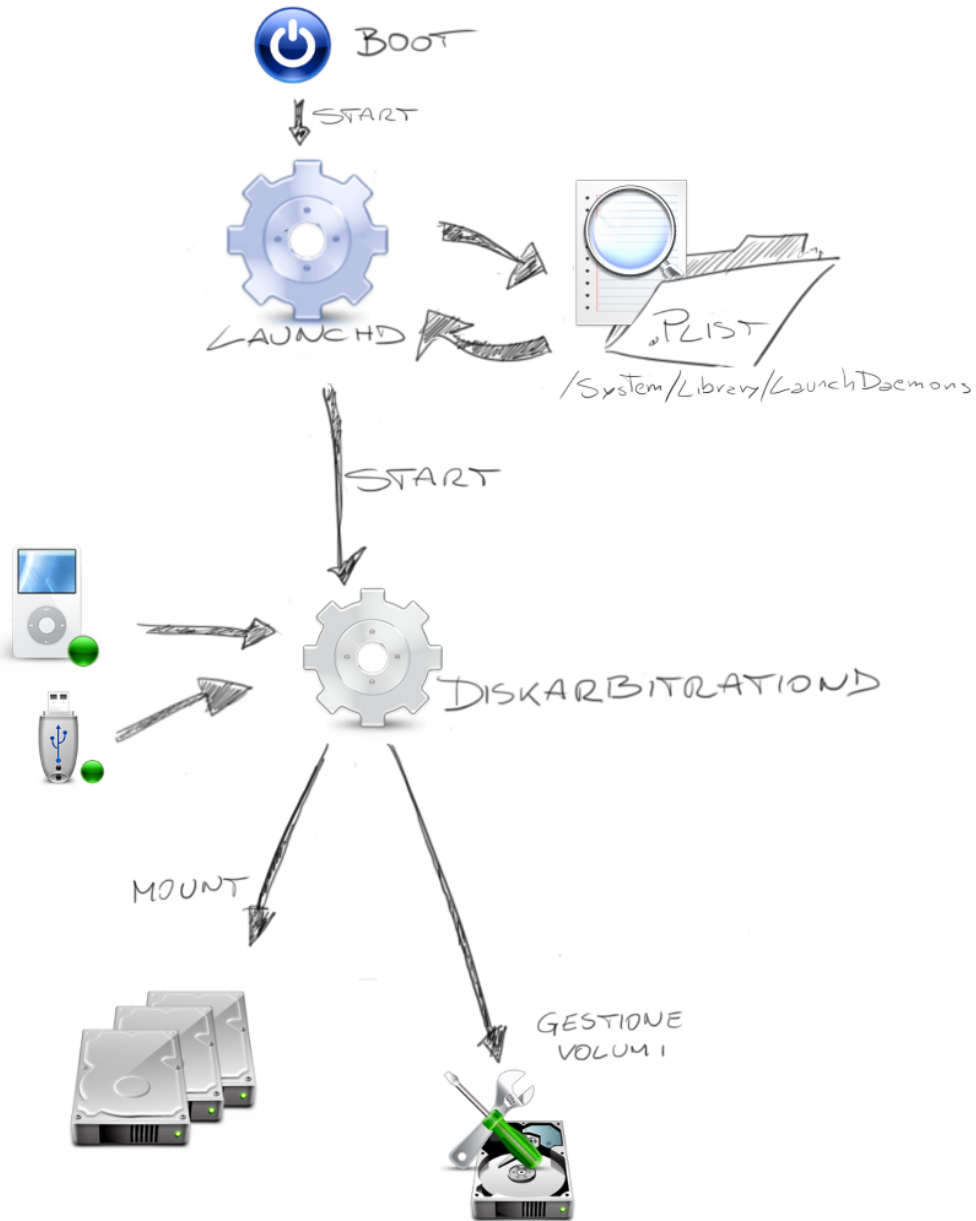
In OS X, la gestione dei dischi e dei filesystem connessi al sistema avviene attraverso un framework dedicato, **Disk Arbitration Framework**, e un demone che si occupa di gestire le notifiche ed il mount delle eventuali periferiche collegate: **diskarbitrationd**.

Nell'installazione normale di OS X, diskarbitrationd è attivo di default, richiamato all'avvio da **launchd**, il demone preposto alla gestione dei processi utente e di sistema.

Launchd, richiamato come primo processo in avvio dal kernel, esegue il parsing dei file presenti nella directory **/System/Library/LaunchDaemons**, all'interno della quale risiede il file **com.apple.diskarbitrationd.plist**, contenente la configurazione di avvio del demone in questione.

All'inserimento di un disco esterno il kernel, attraverso il Disk Arbitration Framework, notifica l'avvenuta connessione al demone diskarbitrationd. Quest'ultimo interroga il file **/etc/fstab** per eventuali mount-point e filesystem predefiniti, interroga il sistema sui filesystem supportati e se tra questi sono presenti quelli relativi al media collegato, monta il disco e gli eventuali volumi riconosciuti.

Lo schema seguente riassume in maniera semplificata il flusso appena descritto:



Alla luce di quanto descritto, abbiamo due possibilità per inibire il mount automatico dei dischi.

Disabilitare temporaneamente il mount automatico

Il comando **launchctl** fornisce un'interfaccia interattiva per la gestione del demone launchd. Attraverso una serie di sotto-comandi possiamo gestire a basso livello il demone launchd e gli agenti ed i demoni da esso controllati:

```

owl:~ mauri$ launchctl
launchd% help
usage: launchctl <subcommand>
load    Load configuration files and/or directories
unload  Unload configuration files and/or directories
start   Start specified job
stop    Stop specified job
submit  Submit a job from the command line
remove  Remove specified job
bootstrap Bootstrap launchd
list    List jobs and information about jobs
setenv  Set an environmental variable in launchd
unsetenv Unset an environmental variable in launchd
getenv  Get an environmental variable from launchd
export  Export shell settings from launchd
debug   Set the WaitForDebugger flag for the target job to true.
limit   View and adjust launchd resource limits
stdout  Redirect launchd's standard out to the given path
stderr  Redirect launchd's standard error to the given path
shutdown Prepare for system shutdown
singleuser Switch to single-user mode
getusage Get resource usage statistics from launchd
log     Adjust the logging level or mask of launchd
umask   Change launchd's umask
bsexec  Execute a process within a different Mach bootstrap
subset
bslist  List Mach bootstrap services and optional servers
bstree  Show the entire Mach bootstrap tree. Requires root
privileges.
managerpid Print the PID of the launchd managing this Mach
bootstrap.
manageruid Print the UID of the launchd managing this Mach
bootstrap.
managername Print the name of this Mach bootstrap.
exit    Exit the interactive invocation of launchctl
quit    Quit the interactive invocation of launchctl
help    This help output

```

I comandi interessanti sono:

- **list** - elenca e fornisce informazioni sui job attivi
- **load** - carica un file di configurazione (.plist)
- **unload** - arresta i job specificati da un file di configurazione (.plist)

```

owl:~ mauri$ sudo ps aux | grep -i diskarb
root    33  0,0  0,0  2446808  1512  ??  Ss   9:22am  0:00.12 /usr/sbin/diskarbitrationd

```

```

owl:~ mauri$ sudo launchctl list | grep diskarb
33 - com.apple.diskarbitrationd

```

Possiamo osservare come il processo diskarbitrationd sia attivo con ID 33. A questo punto possiamo arrestare il processo utilizzando il comando stop seguito dal file di configurazione relativo a diskarbitrationd:

```
owl:~ mauri$ sudo launchctl unload /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist
```

Controlliamo che il processo non sia attivo attraverso i precedenti comandi:

```
owl:~ mauri$ sudo launchctl list | grep -i disk  
- 0 com.apple.diskmanagementd
```

```
owl:~ mauri$ ps aux | grep -i disk  
mauri 370 0,0 0,0 2435032 528 s000 S+ 6:04pm 0:00.00 grep -i disk
```

Con questo sistema abbiamo disabilitato il demone diskarbitrationd e possiamo connettere una periferica da analizzare o duplicare senza che venga montata in automatico.

Per riabilitare il demone possiamo eseguire il reboot del mac o utilizzare il sotto-comando load:

```
owl:~ mauri$ sudo launchctl load /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist
```

Disabilitare il mount automatico in maniera permanente

Eliminando o spostando il file com.apple.diskarbitrationd.plist dalla cartella /System/Library/LaunchDaemons, eviteremo che launchd carichi il demone diskarbitrationd in fase di avvio.

Creiamo una cartella di destinazione per contenere temporaneamente o definitivamente il file plist e spostiamoci quest'ultimo:

```
owl:dev mauri$ sudo mkdir /Users/mauri/morgue/plist.files
```

```
owl:dev mauri$ sudo mv /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist  
/Users/mauri/morgue/plist.files/
```

controlliamo che il file non sia più presente nella dir LaunchDaemons:

```
owl:dev mauri$ ls -la /System/Library/LaunchDaemons/com.apple.d*  
-rw-r--r-- 1 root wheel 458 31 Lug 09:18  
/System/Library/LaunchDaemons/com.apple.dashboard.advisory.fetch.plist  
-rw-r--r-- 1 root wheel 528 1 Ago 08:02 /System/Library/LaunchDaemons/com.apple.diskmanagementd.plist  
-rw-r--r-- 1 root wheel 581 15 Lug 07:26 /System/Library/LaunchDaemons/com.apple.distnoted.plist  
-rw-r--r-- 1 root wheel 425 25 Lug 07:34 /System/Library/LaunchDaemons/com.apple.dnsextd.plist  
-rw-r--r-- 1 root wheel 512 19 Mag 2009 /System/Library/LaunchDaemons/com.apple.docsetinstalld.plist  
-rw-r--r-- 1 root wheel 494 29 Mag 2009  
/System/Library/LaunchDaemons/com.apple.dvdplayback.setregion.plist  
-rw-r--r-- 1 root wheel 226 24 Giu 03:40 /System/Library/LaunchDaemons/com.apple.dynamic_pager.plist
```

Al riavvio diskarbitrationd non sarà più avviato da launchd.

Per ripristinare l'avvio automatico del demone basterà ripristinare il file nella cartella originale:

```
owl:dev mauri$ sudo mv /Users/mauri/morgue/plist.files/com.apple.diskarbitrationd.plist  
/System/Library/LaunchDaemons/
```

```
owl:dev mauri$ ls -la /System/Library/LaunchDaemons/com.apple.d*
```

```

-rw-r--r-- 1 root wheel 458 31 Lug 09:18
/System/Library/LaunchDaemons/com.apple.dashboard.advisory.fetch.plist
-rw-r--r-- 1 root wheel 624 7 Dic 10:59 /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist
-rw-r--r-- 1 root wheel 528 1 Ago 08:02 /System/Library/LaunchDaemons/com.apple.diskmanagementd.plist
-rw-r--r-- 1 root wheel 581 15 Lug 07:26 /System/Library/LaunchDaemons/com.apple.distnoted.plist
-rw-r--r-- 1 root wheel 425 25 Lug 07:34 /System/Library/LaunchDaemons/com.apple.dnsextd.plist
-rw-r--r-- 1 root wheel 512 19 Mag 2009 /System/Library/LaunchDaemons/com.apple.docsetinstalld.plist
-rw-r--r-- 1 root wheel 494 29 Mag 2009
/System/Library/LaunchDaemons/com.apple.dvdplayback.setregion.plist
-rw-r--r-- 1 root wheel 226 24 Giu 03:40 /System/Library/LaunchDaemons/com.apple.dynamic_pager.plist

```

Disabilitare il mount automatico in Tiger

La struttura del Disk Arbitration Framework in MAC OS X 10.4 è leggermente differente e non ci consente di abilitare o disabilitare diskarbitrationd in maniera interattiva.

Utilizzeremo quindi un procedimento analogo a quello impiegato in OS X 10.5 per disabilitare diskarbitrationd in modo permanente, varia solo la posizione del file da spostare:

```
mauri$ sudo mv /etc/mach_init.d/diskarbitrationd.plist /Users/mauri/morgue/plist.files
```

Riavviare il sistema.

Come nella procedura relativa a Leopard, per ripristinare il tutto basterà riportare il file nella posizione iniziale:

```
mauri$ sudo mv /Users/mauri/morgue/plist.files/diskarbitrationd.plist /etc/mach_init.d/
```

Attivare il logging di diskarbitrationd

Quando si tratta di indagini delicate i log non sono mai abbastanza. Abilitare i log approfonditi di questo demone che si occupa di montare e smontare dischi e filesystem non può che rivelarsi utile e salutare.

Avremo una traccia di quanto avviene sul sistema a livello di dischi e potremo allegare alla reportistica gli eventi legati ad un malaugurato mount accidentale.

Per abilitare il logging è necessario modificare il file di configurazione com.apple.diskarbitrationd.plist come segue:

```

:::::::::: ORIGINALE ::::::::::::
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnableTransactions</key>
  <true/>
  <key>HopefullyExitsLast</key>
  <true/>
  <key>KeepAlive</key>
  <true/>
  <key>Label</key>
  <string>com.apple.diskarbitrationd</string>
  <key>MachServices</key>
  <dict>
    <key>com.apple.DiskArbitration.diskarbitrationd</key>
    <true/>
  </dict>
  <key>ProgramArguments</key>
  <array>

```



```

13:14:59
13:14:59 diskarbitrationd [532] -> diskarbitrationd [532]
13:14:59 probed disk, id = /dev/disk0s2, with hfs, ongoing.
13:14:59 probed disk, id = /dev/disk0s2, with hfs, success.
13:15:05
13:15:05 iokit [0] -> diskarbitrationd [532]
13:15:05 created disk, id = /dev/disk1.
13:15:05 created disk, id = /dev/disk1s1.
13:15:05
13:15:05 diskarbitrationd [532] -> diskarbitrationd [532]
13:15:05 probed disk, id = /dev/disk1s1, with msdos, ongoing.
13:15:05 probed disk, id = /dev/disk1s1, with msdos, success.
13:15:05 mounted disk, id = /dev/disk1s1, ongoing.

```

Come si può notare le informazioni riportate sono molto utili se rilevate nell'ottica dell'analista forense.

Verifica inibizione disco USB inserito

Eseguiamo l'esperimento su una chiavetta USB da 512 MB.

Per verificare che il device inserito sia duplicabile ma non modificabile, utilizzeremo i tools che sono compresi in una distribuzione base di OS X 10.5 : dd e shasum.

Osserviamo la situazione base dei dischi:

```

owl:~ root# ls /dev/*disk*
/dev/disk0 /dev/disk0s1 /dev/disk0s2 /dev/rdisk0 /dev/rdisk0s1 /dev/rdisk0s2

```

Colleghiamo quindi la chiavetta e verifichiamo la presenza di un nuovo device:

```

owl:~ root# ls /dev/*disk*
/dev/disk0 /dev/disk0s1 /dev/disk0s2 /dev/disk1 /dev/disk1s1 /dev/rdisk0 /dev/rdisk0s1 /dev/rdisk0s2
/dev/rdisk1 /dev/rdisk1s1

```

Una controllata alla tipologia del device inserito:

```

owl:dev mauri$ hdiutil partition /dev/rdisk1
scheme: fdisk
block size: 512
_ ## Type _____ Name _____ Start__ Size ___
+ MBR          Master Boot Record      0    1
+ Apple_Free           1    31
  1 DOS_FAT_16          32 1023456
+ Apple_Free           1023488 512
+ synthesized

```

Calcoliamo l'hash del device:

```

owl:dev mauri$ sudo -i
owl:~ root# shasum /dev/rdisk1
d3a924c67bc0002c35e95bb2c261fae0bf802d4a /dev/rdisk1

```

Duplichiamo infine la chiavetta USB:

```
owl:~ root# dd if=/dev/rdisk1 conv=sync,noerror of=/Users/mauri/morgue/usbpen.dd
1024000+0 records in
1024000+0 records out
524288000 bytes transferred in 577.644631 secs (907631 bytes/sec)
```

Calcoliamo l'hash dell'immagine per controllare che corrisponda a quella del device:

```
owl:~ root# shasum /Users/mauri/morgue/usbpen.dd
d3a924c67bc0002c35e95bb2c261fae0bf802d4a /Users/mauri/morgue/usbpen.dd
```

Ora scollegiamo e ri-collegiamo la chiavetta un paio di volte, non si sa mai.
Calcoliamo nuovamente l'hash:

```
owl:~ root# shasum /dev/rdisk1
d3a924c67bc0002c35e95bb2c261fae0bf802d4a /dev/rdisk1
```

A questo punto possiamo affermare che la USB pen non è stata modificata in alcun modo e possiamo divertirci a duplicare il file dd per effettuare le dovute indagini, invasive o meno.

Per concludere, la raccomandazione di rito prima di accingersi a duplicare un device connettendolo ad un mac è quella di aver compreso a fondo il sistema e, possibilmente, aver eseguito un paio di test con dispositivi differenti. Un write-blocker eviterà poi sollecitazioni supplementari alle coronarie già abbastanza shakerate di un analista forense.

