

# OpenBSD



# filesystem

Cosa troviamo nella radice (/) ?

Per rispondere a questa domanda ci viene incontro la pagina del manuale di *hier*

( **man hier** o <http://www.openbsd.org/cgi-bin/man.cgi?query=hier&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=html> )

Ora che sappiamo cosa ci troviamo (e quindi dovremmo anche sapere che stiamo facendo), possiamo iniziare a pensare a come utilizzare al meglio il filesystem per implementare la sicurezza

Una macchina insicura localmente lo sarà ancor di più da remoto

# filesystem - mount

mount, come nella maggior parte delle piattaforme unix based permette di gestire i filesystem.

Le opzioni più interessanti

*noexec*

*nosuid*

*rdonly*

*softdep*

*Altre opzioni interessanti*

*noatime*

Possiamo sempre aggiornare le opzioni di mount con **-u -f**

# La configurazione del sistema

Come abbiamo visto dalla pagina di *hier*, in */etc* troviamo la configurazione del sistema.

Andremo ora a vedere alcuni file e come vanno gestiti.

*/etc/myname*

*/etc/mygate*

*/etc/hostname.if*

*/etc/boot.conf*

Un altro file molto importante è */etc/rc.conf*. Da notare che tale file va considerato solo come un modello, ma che ogni modifica va apportata al file di over-ride */etc/rc.conf.local*.

# La configurazione del sistema

Per modificare molti dei comportamenti del kernel OpenBSD ci mette a disposizione un comando:

***sysctl***

Questo comando ci permette di accedere a moltissime variabili e parametri del sistema.

*ddb, fs, hw, kern, machdep, net, user, vfs, vm*

Alcuni sono read-only, mentre altri possono esser modificati.

L'utilizzo di sysctl permette solo una modifica **temporanea** di queste impostazioni.

In tal caso ci viene in aiuto il file */etc/sysctl.conf*, che modificato in modo corretto ed opportuno, rende questi cambi permanenti (ad ogni riavvio di sistema questo file verrà letto ed applicate le variabili in esso).

# La configurazione del sistema

Per rendere ancora più sicuro OpenBSD si è ricorsi al *securelevel*.

Questo modo è comune alle piattaforme BSD.

OpenBSD offre 4 livelli: -1, 0, 1, 2

( diamo un'occhiata a cosa ci dice la pagina del manuale – man securelevel )

# La configurazione del sistema

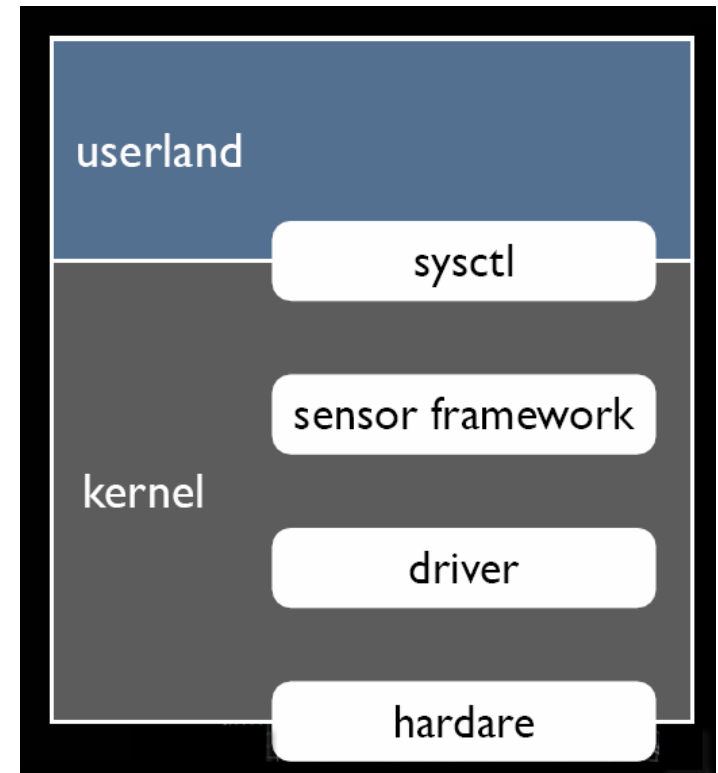
OpenBSD, come ogni sistema operativo, per esser affidabile si basa anche sull'hardware che abbiamo a disposizione.

Il progetto ha portato avanti essenzialmente 2 tool integrati nel sistema di base:

Uno è inerente hardware specializzato e l'altro fa riferimento allo stato delle interfacce di rete.

I sensori sono un qualsiasi oggetto (integrato o aggiunto) che possa fornire informazioni sullo stato del sistema. Insomma si tratta di qualcosa che sia in grado di dirci qualcosa inerente lo stato del computer (i.e. temperature a voltaggio della cpu, temperatura ambiente, stato di eventuali alimentatori ridondati).

Per l'hardware specializzato si ricorre al demone ***sensorsd*** che ha come file di configurazione */etc/sensorsd.conf*



# La configurazione del sistema

Oltre ai sensori, per hardware specializzato abbiamo anche **bio**.

Questo tool è gestito da **bioctl**.

**bio** nasce per gestire specialmente hardware raid. Molti produttori non rilasciano tool free. Alcuni rifiutano anche di dare le specifiche. Il team di OpenBSD, per mezzo di reverse engineering e con alcuni vendor collaborativi, ha potuto sviluppare questo tool per gestire e verificare questi device di storage.

**bio** agisce come uno pseudo-device: permette di gestire ed accedere alle periferiche che non abbiamo una relativa entry all'interno di `/dev`

Per quel che riguarda il monitoring delle interfacce di rete abbiamo a disposizione ***ifstated*** ed al relativo file di configurazione `/etc/ifstated.conf`

Questo tool si dimostra assai utile nel caso si debba gestire il failover.



# Aggiornamento del sistema

Alla pagina <http://www.OpenBSD.org/errata.html> troviamo tutti i link alla nostra specifica versione.

Qui possiamo accedere alla pagina errataXX.html che contiene, in successione, tutti gli aggiornamenti, miglioramenti ed eventuali security bug della versione.

È importante consultare questa pagina con una certa frequenza.

Una volta, che abbiamo identificato, un aggiornamento che sia o necessario o che ci interessi dovremmo ricompilare.

Sarà quindi necessario aver installato il compilatore C (gcc) ed avere una copia dei sorgenti.

Questi possono essere scaricati dal sito oppure dai cd di installazione (cd #3).

Se dobbiamo ricompilare 1 solo binario, possiamo scaricare la patch ad esso inerente e seguire la procedura inclusa con la patch.

# Aggiornamento del sistema

Nel caso in cui si voglia creare un sistema aggiornato (per esempio vogliamo installare la versione *stable*, dovremmo ricompilare tutto il sistema e preparare i pacchetti d'installazione.

In tal caso le FAQ ci vengono in aiuto le F.A.Q. (faq #5)

Oltre a tali motivi si può esser un altro motivo per cui noi si debba metter mano al sistema:

Abilitare e disabilitare specifico hardware.

# Aggiornamento del sistema

Abilitare e disabilitare specifico hardware.

Per modificare un hardware già presente nel kernel GENERIC possiamo ricorrere a 2 tool:

config

config

(non si tratta di un errore :). config permette sia di editare i parametri di un kernel esistente (modificare parametri e disabilitare driver) sia di verificare la correttezza (formale) di un file di configurazione del kernel)

Nel primo caso il comando sarà `config -e [-o nuovo kernel ] /bsd|/bsd.rd`

Qui accederemmo all'interfaccia di riconfigurazione del kernel

È facilmente riconoscibile in quanto il prompt muterà in *ukc>*

# Aggiornamento del sistema

Se invece vogliamo aggiungere qualche driver che non è presente nel kernel GENERIC, dovremmo ricorrere ai seguenti step:

*Ottenere i sorgenti del sistema*

*Editare il nuovo file di configurazione (possiamo usare GENERIC come skeleton)*

*Eeguire il comando **config** nei confronti del nuovo kernel*

Questo è altresì l'unico caso in cui il team di OpenBSD potrà darvi supporto se avete problemi.

Il team, non a torto, sconsiglia di crearsi un kernel custom.

Per poter creare un kernel, esente da difetti, e veramente funzionante si deve studiare ogni singolo driver anche nelle sue dipendenze.

# Credits

Tutto il materiale presentato è stato tratto da sito di OpenBSD.org e dal manuale on line di OpenBSD