Hardening Macchine Linux

# HARDENING

## Sicurezza Fisica

Configurare il BIOS in modo da disabilitare il booting da CD/DVD, Device Esterni, Floppy.

Impostare una password al BIOS

Proteggere il GRUB con una password in modo da ridurre accessi fisici al sistema

## Partizioni dei dischi

Fare in modo che il disco non sia costituito da un’unica partizione. Avere più partizioni consente una maggiore sicurezza nel caso succeda un qualunque disastro. Creando diverse partizioni, i dati possono essere separati e raggruppati. Quando avviene un incidente, solo i dati della partizione sono compromessi, mentre i dati delle altre partizioni rimangono inalterati.

Inoltre alcune delle directories tipiche dei sistemi linux sono esposte ad attacchi di tipo noto, e questo a causa del ruolo che svolgono all’interno del sistema operativo, e quindi è plausibile che siano montate su filesystem separati dal resto del sistema.

In particolare le directories:

/var è la directory che contiene i logs del sistema, ed una generazione fuori controllo può in poco tempo esaurire tutto lo spazio della partizione e quindi del disco in una soluzione con un’unica partizione, e questo impedisce al sistema operativo di funzionare correttamente.

/tmp è la directory adibita a contenere file temporanei; gode spesso di privilegi alti e quindi rootkit ed exploit vengono spesso lanciati proprio da questa directory, proprio perché quella più vulnerabile.

Analizzando il file /etc/fstab è possibile verificare quali siano le partizioni presenti ed i relativi punti di mount al boot, ed è quindi necessario far si che /var e /tmp siano montati su due filesystem separati. Conviene creare partizioni separate anche per le directories usr ed home, ed inoltre abilitare le quote in modo da evitare un riempimento totale delle partizioni in caso di malfunzionamento.

## Minimizzare pacchetti per minimizzare le vulnerabilità

Evitare l’installazione di pacchetti inutili per evitare le vulnerabilità indotte dagli stessi. Questo porta a minimizzare il rischio che la compromissione di un servizio possa portare alla compromissione di altri servizi.

E’ buona norma trovare e disabilitare e/o eliminare servizi non desiderati dalla macchina. Per far ciò si utilizza il comando

**/sbin/chkconfig --list |grep '3:on'**

che permette di ritrovare servizi che sono in esecuzione sul runlevel 3, che rappresenta la configurazione di sistema in cui tutti i servizi sono attivi.p

## Kernel e pacchetti aggiornati

È di fondamentale importanza **aggiornare i pacchetti alle ultime versioni disponibili**, in modo che bug delle versioni precedenti, e corretti con le ultime, non possano essere un veicolo di attacco. Stesso discorso vale per il kernel.

Per questa operazione, occorre affidarsi **al package manager del proprio sistema**; generalmente tramite interfaccia grafica è possibile vedere in modo più semplice le versioni di riferimento. È inoltre utile tenere sott’occhio i forum, e le newsletter che notificano gli aggiornamenti di sicurezza del kernel per avere una panoramica più completa.

## Confinamento Cronjobs

Cron specifica chi può e non può lanciare job. Questi controlli sono effettuati tramite in file

*/etc/cron.allow* che contiene la lista degli utenti che possono lanciare cron

*/etc/cron.deny* che contiene, invece, la lista degli utenti che non possono usare cron

Per evitare che gli utenti possano utilizzare cron basta aggiungere ALL al file cron.deny

**echo ALL >> /etc/cron.deny**

## Disabilitare il rilevamento delle USB stick

Per disabilitare le usb stick bisogna invece creare innanzitutto il file */etc/modprobe.d/no-usb*

e scrivere all’interno

**install usb-storage /bin/true**

## Proteggere le comunicazioni

L’utilizzo di servizi come **telnet**, **ftp**, **rsh**, **comporta che username e password siano trasmessi in chiaro**, il che espone questi dati alla possibile cattura di un packet sniffer in ascolto sulla rete. Onde evitare problemi legati alla possibilità di intercettare username e password che vengono trasmessi in chiaro sulla rete è necessario utilizzare qualche forma di cifratura dei dati, per esempio con l’utilizzo di **OpenSSH**, **scp**, **SFTP**, che forniscono servizi di protezione dei dati trasmessi sulla rete.

In particolare ssh, utilizzato per l’accesso remoto al sistema operativo, viene installato con un configurazione di default che va modificata (file **/etc/ssh/sshd\_config**), **in particolare l’accesso con permessi di root non deve essere consentito ad alcun utente**, ed allo stesso tempo deve essere disabilitata l’**autenticazione senza password**; normalmente ssh è un servizio in ascolto sulla porta 22, e come tale è esposto ad attacchi di forza bruta che in automatico operano su quella stessa porta, per questo motivo è conveniente **modificare la porta di questo servizio, facendo attenzione a non andare in conflitto con la porta di qualche altro demone**.

Nel file **/etc/ssh/sshd\_config**, le modifiche da fare sono:

**PermitRootLogin no**

**PermitEmptyPasswords no**

**Port 41567**

**Protocol 2**

## Chiudere le porte aperte

Tutti i servizi che non sono di stretta necessità per il sistema e che sono in ascolto di connessioni di ingresso, **possono rappresentare dei possibili punti di attacco, per tanto occorre chiudere o disabilitare questi servizi**. Per individuare quelli che sono in ascolto su delle porte, è sufficiente utilizzare netstat, che con la seguente sintassi consente di avere l’elenco delle porte aperte, lo stato della connessione ed il pid del processo che controlla quella connessione.

**netstat -tulpn**

In questo modo è possibile identificare le connessioni attive e fare in modo di chiudere le  porte aperte (stoppando i servizi, o in casi estremi uccidendo i processi).

## Controllo d’accesso

Implementare una politica di tipo Mandatorio (MAC). Eseguire programmi con il kernel MAC, garantisce protezione contro software malevoli che possono danneggiare il sistema, o fornire porte di accesso non autorizzate.

# APPARMOR

## Introduzione

AppArmor è un'implementazione del «Linux Security Module» per il controllo degli accessi vincolante basato sul nome. AppArmor racchiude individualmente i programmi in un insieme di file e capacità posix 1003.1e draft.

AppArmor è installato e caricato in modo predefinito e utilizza i profili di un'applicazione per determinare quali file e permessi siano necessari all'applicazione. Alcuni pacchetti installano i propri profili e ulteriori profili possono essere trovati nel pacchetto *apparmor-profiles*.

sudo apt-get install apparmor-profiles

I profili di AppArmor dispongono di due modalità di esecuzione:

1. Apprendimento (complaining/learning): le violazioni del profilo sono consentite e vengono registrate. Utile per verificare e sviluppare nuovi profili.
2. Esecutiva (enforced/confined): obbliga a rispettare la politica del profilo e registra le violazioni.

## Profili

I profili di apparmor sono dei file di testo situati nella cartella /etc/apparmor.d/ .

Il nome dei file è rappresentativo del path dell’applicativo di cui rappresentano il profilo. Avremo cioè che, volendo profilare il comando ping, situato in /bin/ping il nome del file profilo sarà bin.ping.

Di seguito è rappresentata la struttura di un profilo di apparmor :

In un profilo, i commenti sono sempre preceduti dal simbolo # .

# profile to configure foo

#include<tunables/global>

@{HOME} = /home/\*/ /root/ # variabile

/usr/bin/foo {

#include <abstractions/base>

network inet tcp,

capability setgid,

/bin/mount ux,

/dev/{,u}random r,

/etc/ld.so.cache r,

/etc/foo/\* r,

/lib/ld-\*.so\* mr,

/lib/lib\*.so\* mr,

/proc/[0-9]\*\* r,

/usr/lib/\*\* mr,

/tmp/ r,

/tmp/foo.pid wr,

/tmp/foo.\* lrw,

/@{HOME}/.foo\_file rw,

/@{HOME}/.foo\_lock kw,

link /etc/sysconfig/foo -> /etc/foo.conf,

deny /etc/shadow w,

owner /home/\*/\*\* rw,

/usr/bin/foobar cx,

/bin/\*\* px -> bin\_generic

# comment on foo's local profile, foobar.

foobar {

/bin/bash rmix,

/bin/cat rmix,

/bin/more rmix,

/var/log/foobar\* rwl,

/etc/foobar r,

}

}

## #Include<tunables/global>

Attraverso l’istruzione #include, si può dare al programma l’accesso a una serie di cartelle o file che sono richiesti anche da altri programmi. Usando gli include è possibile ridurre la dimensione di un profilo. Esistono tre tipi di inclusioni:

- **Abstractions**, che rappresentano un insieme di task base per le applicazioni (accesso ai meccanismi di autenticazione, system accounting, accesso alle routine di name service,…)

- **Program chunks** contengono i controlli d’accesso a programmi specifici che l’amministratore vuole regolare in base alle politiche del sistema. Ogni chunk è usato da un singolo programma.

- **Tunables** sono regole di inclusione da altri file. Consente di usare un file comune con regole di inclusione per molteplici applicazioni.

## Variabili locali

Le variabili locali sono definite all’inizio del profilo. Vengono spesso utilizzate per create shortcuts, come nell’esempio:

@{CHROOT\_BASE}=/tmp/foo

/sbin/syslog-ng {

...

# chrooted applications

@{CHROOT\_BASE}/var/lib/\*/dev/log w,

@{CHROOT\_BASE}/var/log/\*\* w,

...

}

## Alias

possono essere usati al posto delle variabili locali per riscrivere percorsi

alias /home/ -> /mnt/users/

## Rules

Nella restante parte del profilo sono invece contenute delle regole di vario genere, in relazione all’aspetto da controllare. Vi sono:

1. **Path Entries**: contengono informazioni circa i file a cui l’applicazione può accedere.
2. **Capability Entries**: determinano i privilegi di cui gode il processo. Queste capability si costruiscono anteponendo la parola “capability” alle capacità descritte nella POSIX.1e
3. **Network Entries**: determinano il tipo di connessione (tcp, datagram, raw) .

Per quello che concerne le Path Entries, le opzioni utilizzabili sono:

| Options | File | Descrizione |
| --- | --- | --- |
| read | r | permesso di lettura |
| write | w | permesso di scrittura |
| link | l | regola la creazione di link verso altri file |
| file locking | k | permesso di bloccare un file |
| file append | a | possibilità di aggiungere dati ad un file |
| deny | deny | nega l’accesso a file o cartelle |
| owner | owner | espressione condizionale per indicare la proprietà di un file (euid e fsuid uguali per file e processo) |
| memory map | m | permesso di mappare in memoria una file eseguibile |

per file eseguibili invece si hanno le seguenti opzioni:

| Option | File | Description |
| --- | --- | --- |
| inherit | ix | deve stare nella stessa cartella del genitore |
| profile | px | richiede che esista un profilo per il programma eseguito |
| Local Profile | cx | richiede che esista un profilo locale per il programma eseguito |
| Unconstrained | ux | esegue il programma senza profile. Modalità da evitare per ragioni di sicurezza |

e infine vi sono anche delle espressioni utilizzate per effettuare l’aggregazione dei percorsi

| Glob | Description |
| --- | --- |
| \* | sostituibile da qualsiasi lettera o numero, eccetto /  (matching con tutti file di una directory) |
| \*\* | sostituibile da qualsiasi sequenza di lettere o numeri, anche / (matching con tutti i file e directory presenti in una cartella ) |
| ? | sostituibile da una lettera o numero, eccetto / |
| [ abc ] | sostituibile con il singolo carattere a, b o c |
| [ a-c ] | sostituibile con il singolo carattere a, b o c |
| {ab, cd} | alternativa: divisibile in due regole, una che effettua un matching con ab e l’altra con cd |
| [ ^a ] | sostituibile con qualunque carattere eccetto che con a |

## Creazione Profilo

I test sono stati effettuati su una macchina in cui è installata la distribuzione Linux OpenSuse 13.01.

Lanciando come root user il comando

apparmor\_status

visualizziamo lo stato dell’applicazione e quindi tutti i profili attivi sulla macchina, sia in modalità complain che in modalità enforce.

Per creare correttamente il profilo di un’applicazione, dobbiamo utilizzare il programma mentre Apparmor si trova in complain mode. Questo procedimento prevede l’esecuzione di varie operazioni sull’applicazione da profilare, quali:

1. l’avvio
2. l’arresto
3. il riavvio
4. utilizzo di eventuali comandi da riga di comando
5. l’utilizzo di tutte le parti di cui si compone l’applicazione

Per eseguire tutta l’analisi e l’apprendimento del comportamento dell’applicazione, bisogna utilizzare innanzitutto il comando:

aa-genprof /percorso/al/binario/da/profilare

che andrà a leggere le attività svolte dall’applicazione sul log.

Quindi una volta lanciato il comando, inizierà una sessione di utilizzo dell’applicazione, che dovrà includere tutte le operazioni di base e quindi che cercherà di delineare il comportamento dell’applicazione stessa.

Ultimata la sessione si ritornerà sul terminale per aggiornare la politica, tramite le direttive di genprof, il quale:

1. (S)can system logs : scanning del log per individuare i record dell’applicazione e quindi andrà a creare sotto il controllo dell’operatore il profilo
2. (F)inish : chisura di genprof

Per un profilo ancora più fine e accurato, bisogna mettere il profilo appena generato in complain mode

aa-complain /etc/apparmor.d/profilo

e quindi mediante il comando

aa-logprof

si andrà ad analizzare il log per verificare se ancora sono presenti delle azioni dell’applicazione ancora non contemplate all’interno del profilo corrente.

Dopo aver effettuato tutti i test, si può infine mettere il profilo in modalità enforce

aa-enforce /etc/apparmor.d/profilo