



Security Enhanced Linux (SELinux)

a cura di:
Michelangelo Magliari
Loredana Luzzi
Andrea Fiore




25/10/2002 Security Enhanced Linux 1



Cos'è SELinux?

- **Prototipo di Sistema Operativo**
- **Realizzato dalla National Security Agency**
- **Implementa un Mandatory Access Control flessibile all'interno del kernel 2.4.2 di Linux.**
- **Prima release pubblica 22 dicembre 2000, kernel 2.2.12 su Red Hat 6.1**
- **<http://www.nsa.gov/selinux>**

25/10/2002 Security Enhanced Linux 2



Perché SELinux

- Nasce per far fronte alle limitazioni dei moderni Sistemi Operativi
- Nei Sistemi Operativi moderni le decisioni d'accesso sono basate sul Discretionary Access Control (DAC)


25/10/2002 Security Enhanced Linux 3



Che cos'è il DAC

- Meccanismo software
- permette al proprietario di un file o di una directory di concedere o negare l'accesso agli altri utenti.


25/10/2002 Security Enhanced Linux 4



Esempio

- `int creat(const char *pathname, mode_t mode)`
- *mode* specifica i permessi di accesso al file:
 - ✓ lettura, scrittura, esecuzione al proprietario
 - ✓ lettura, scrittura, esecuzione al gruppo
 - ✓ lettura, scrittura, esecuzione agli altri utenti

25/10/2002 Security Enhanced Linux 5



Esempio

- In un sistema con DAC i permessi del file possono essere cambiati da un processo in esecuzione con lo stesso user ID del proprietario del file, o user ID 0 (root)

25/10/2002 Security Enhanced Linux 6



Perché il DAC è inadeguato

- Decisioni di accesso basate solo sull'identità dell'utente e su ciò che possiede nel sistema
- L' amministratore del sistema ne ha il completo controllo

25/10/2002

Security Enhanced Linux

7



Che cos'è il MAC

- Mandatory Access Control (MAC)
- Mezzo per limitare l'accesso agli oggetti
- Basato sulla sensibilità delle informazioni contenute negli oggetti
- e sulle autorizzazioni dei soggetti per accedere a tali informazioni sensibili

25/10/2002

Security Enhanced Linux

8



Esempio

- `int creat(const char *pathname, mode_t mode)`
- `mode` specifica i permessi di accesso al file:
 - ✓ lettura, scrittura, esecuzione al proprietario
 - ✓ lettura, scrittura, esecuzione al gruppo
 - ✓ lettura, scrittura, esecuzione agli altri utenti

25/10/2002

Security Enhanced Linux

9



Esempio

- In un sistema con MAC ad un file viene assegnata un'etichetta di sicurezza basata sull'importanza delle informazioni contenute nel file
- Ogni etichetta ha un certo livello di sicurezza

25/10/2002

Security Enhanced Linux

10



Esempio

- Il MAC non si cura di chi sia il proprietario del file, o se l'ID del processo, che ha effettuato una richiesta, è 0 (root)
- Un processo che ha etichetta di sicurezza di livello inferiore a quella del file non può leggere il file
- Un processo che ha etichetta di sicurezza diversa da quella del file non può scrivere il file

25/10/2002

Security Enhanced Linux

11



Che cosa può offrire il MAC

- Decisioni di accesso basate su etichette di sicurezza
- L'amministratore non ha più il controllo assoluto (il concetto di root così come lo conosciamo non esiste più)

25/10/2002

Security Enhanced Linux

12

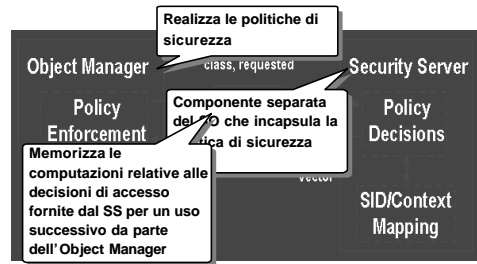


MAC flessibile

- Realizzato dalla National Security Agency (NSA) in collaborazione con la Secure Computing Corporation (SCC)
- Ottenuto sulla base del lavoro fatto sul Type Enforcement (TE)
- Basato sull'architettura FLASK



Architettura Flask



Architettura Flask

- Definisce due tipi di dati indipendenti per le etichette di sicurezza:
 - Contesto di sicurezza: rappresentazione dell'etichetta tramite una stringa di lunghezza variabile;
 - Identificatore di sicurezza (SID): intero mappato dal SS in un contesto di sicurezza.



Architettura Flask: Object Manager

- L'Object Manager gestisce l'associazione delle etichette di sicurezza agli oggetti
- L'Object Manager richiede un'etichetta per un nuovo oggetto al SS
- L'Object Manager consulta l'AVC per ottenere le decisioni di accesso dal SS



Architettura Flask: decisioni di labeling

Il SID per il nuovo oggetto è restituito come un parametro di output.

```
int security_transition_sid(
    security_id_t ssid,
    security_id_t tsid,
    security_class_t tclass,
    security_id_t *out_sid);
```

```
ret = security_transition_sid(
    current->sid,
    dir->i_sid,
    SECLASS_FILE,
    &sid);
```

I parametri di input sono il soggetto SID, il SID di un oggetto inerente e la classe di un nuovo oggetto.

Interfaccia ed esempio di chiamata per ottenere una security label.



Architettura Flask: decisioni di accesso

I permessi concessi sono ritornati come parametri di output.

```
int security_compute_av(
    security_id_t ssid,
    security_id_t tsid,
    security_class_t tclass,
    access_vector_t requested,
    access_vector_t *allowed,
    access_vector_t *decided,
    __u32 *seqno);
```

I parametri di input sono una coppia di SID, la classe dell'oggetto, e l'insieme dei permessi richiesti.

Interfaccia per ottenere le decisioni di accesso dal security server.

Secure Enhanced Linux (SELinux)

- **Prototipo realizzato dalla National Security Agency implementando l'architettura Flask all'interno del kernel 2.4.2 di Linux**

25/10/2002 Security Enhanced Linux 19

SELinux: controlli dei processi

PERMISSION(S)	DESCRIPTION
execute	Execute
transition	Change label
entrypoint	Enter via program
sigkill	Signal
sigalt	
sigstop	
sigchld	
sigalrm	
fork	Fork
rtexec	Trace
getsechid	Get schedule info
setsechid	Set schedule info
getsession	Get session
getppid	Get process group
setppid	Set process group
getcap	Get capabilities
setcap	Set capabilities

Questo permesso è usato per controllare la capacità di un processo di transire da un SID ad un altro
 Questo permesso è usato per controllare la capacità di un processo di tener traccia di un altro processo

Permessi per la classe oggetto processo.

Oltre a questi permessi SELinux fornisce un permesso equivalente per ognuno dei permessi già esistenti in Linux

25/10/2002 Security Enhanced Linux 20

SELinux: controlli dei files

PERMISSION(S)	DESCRIPTION
create	Create
getattr	Get attributes
setattr	Set attributes
inherit	Inherit across execve
receive	Receive via IPC

Permessi per la classe oggetto open file description.

25/10/2002 Security Enhanced Linux 21

SELinux: controlli dei files

- SELinux etichetta e controlla gli open file descriptions poiché essi possono essere ereditati attraverso execve o trasferiti tramite socket UNIX IPC
- Un open file description è etichettato con il SID del suo processo creatore

25/10/2002 Security Enhanced Linux 22

SELinux: controlli dei files

PERMISSION(S)	DESCRIPTION
mount	Mount
remount	Change options
unmount	Unmount
getattr	Get attributes
relabelfrom	Relabel
relabelto	
transition	
associate	Associate file

Permessi per la classe oggetto file system.

25/10/2002 Security Enhanced Linux 23

SELinux: controlli dei files

- SELinux lega etichette di sicurezza ai files e alle directories e controlla gli accessi ad essi
- Per ogni filesystem, SELinux memorizza una tavola di labeling persistente che specifica l'etichetta di sicurezza per ciascun file e directory in quel filesystem
- SELinux assegna un valore intero, detto persistent SID (PSID), a ciascuna etichetta di sicurezza usata da un oggetto in un filesystem

25/10/2002 Security Enhanced Linux 24



SELinux: controlli dei files

PERMISSION(S)	DESCRIPTION
read	Read
write	Write or append
append	Append
poll	Poll/select
lock	IO control
create	Create
execute	Execute
access	Check accessibility
getattr	Get attributes
setattr	Set attributes
unlink	Remove hard link
link	Create hard link
rename	Rename hard link
lock	Lock or unlock
relabelfrom	Relabel
transition	

Permessi per le classi oggetto pipe e file.



SELinux: controlli dei files

PERMISSION(S)	DESCRIPTION
add_name	Add a name
remove_name	Remove a name
reparent	Change parent directory
search	Search
rmdir	Remove
mounton	Use as mount point
mountassociate	

Permessi aggiuntivi per la classe oggetto directory



SELinux: controlli dei socket

- Al socket layer, SELinux controlla la capacità dei processi di eseguire operazioni sui socket
- Al transport layer, SELinux controlla la capacità di spedire e ricevere messaggi sulle interfacce di rete
- SELinux controlla anche la capacità dei processi di configurare le interfacce di rete e di manipolare la tavola di routing del kernel



SELinux: controlli dei socket

PERMISSION(S)	DESCRIPTION
bind	Bind name
name_bind	Use port or file
connect	Initiate connection
getopt	Get socket options
setopt	Set socket options
shutdown	Shut down connection
recvfrom	Receive from socket
sendto	Send to socket
recv_msg	Receive message
send_msg	Send message

Permessi aggiuntivi per la classe oggetto socket



SELinux: controlli dei socket

- Il servizio connection-oriented fornito dallo stream socket richiede ulteriori permessi

PERMISSION(S)	DESCRIPTION
listen	Listen for connections
accept	Accept a connection
newconn	Create new socket for connection
connectto	Connect to server socket
acceptfrom	Accept connection from client socket

Permessi aggiuntivi per le classi oggetto TCP e Unix stream socket.



SELinux: controlli dei socket

PERMISSION(S)	DESCRIPTION
getattr	Get attributes
setattr	Set attributes
tcp_recv	Receive TCP packet
tcp_send	Send TCP packet
udp_recv	Receive UDP packet
udp_send	Send UDP packet
rawip_recv	Receive Raw IP packet
rawip_send	Send Raw IP packet

Permessi per le classi oggetto interfaccia di rete e nodo.



SELinux: controlli dei socket

- I socket servono come proxies di comunicazione nel modello di controllo di SELinux
- I socket sono etichettati di default con l'etichetta del processo creatore
- In SELinux i messaggi sono associati sia con l'etichetta del socket trasmittente, sia con l'etichetta distinta del messaggio
- Per default l'etichetta del messaggio è la stessa del socket trasmittente

25/10/2002

Security Enhanced Linux

31



SELinux: API

- Chiamate API esistenti in Linux non modificate
- Nuove chiamate API che estendono quelle esistenti con parametri aggiuntivi per i SID

25/10/2002

Security Enhanced Linux

32



Esempio di Security Server in SELinux

- Il Security Server di esempio è ottenuto tramite una combinazione di:
 - ✓ Identity Based Access Control (IBAC)
 - ✓ Role Based Access Control (RBAC)
 - ✓ Type Enforcement (TE)
- Decisioni di etichettamento e di accesso sono definite tramite files di configurazione
- SELinux non dipende da questo modello, sostituibile con qualsiasi altro modello

25/10/2002

Security Enhanced Linux

33



Concetti Type Enforcement

- Domini per i processi e tipi per gli oggetti
- Specifica gli accessi consentiti dai domini ai tipi
- Specifica le interazioni ammissibili tra domini
- Specifica le transizioni permesse e automatiche tra domini
- Specifica le restrizioni di endpoint e di esecuzione di codice per i domini

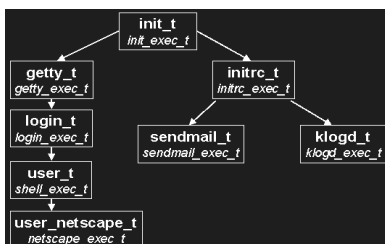
25/10/2002

Security Enhanced Linux

34



Type Enforcement: domini



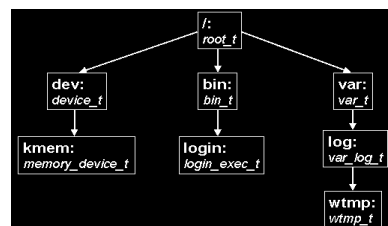
25/10/2002

Security Enhanced Linux

35



Type Enforcement: tipi



25/10/2002

Security Enhanced Linux

36



Type Enforcement: istruzioni

- Istruzione *allow* permette di specificare permessi tra coppie di tipi per ogni classe di oggetti, la sintassi è:

```
allow type_1 type_2: class { perm_1 ... Perm_n};
```

- Istruzione *type_transition* permette transizioni automatiche tra tipi, la sintassi è:

```
type_transition type_1 type_2: file default_file_type
type_transition type_1 type_2: process default_file_type
```

25/10/2002

Security Enhanced Linux

37



Concetti Role Based Access Control

- Ruoli per i processi
- Specifica domini che possono essere accessi da ciascun ruolo
- Specifica ruoli che sono legittimi per ciascun utente
- Dominio iniziale associato con ciascun ruolo utente
- Le transizioni tra ruoli sono esplicite, per esempio login o newrole

25/10/2002

Security Enhanced Linux

38



Role Based Access Control: ruoli

system_r	user_r	sysadm_r
<i>init_t</i>	<i>user_t</i>	<i>sysadm_t</i>
<i>getty_t</i>	<i>user_netscape_t</i>	<i>insmod_t</i>
<i>klogd_t</i>	<i>passwd_t</i>	<i>fsadm_t</i>
<i>sendmail_t</i>		

25/10/2002

Security Enhanced Linux

39



Obiettivi di sicurezza

- Proteggere l'integrità del kernel, includendo i files di boot, i moduli e le variabili sysctl
- Proteggere l'integrità del software di sistema, i files di configurazione e i logs
- Proteggere il ruolo e il dominio dell'amministratore
- Confinare i processi di sistema e i programmi privilegiati
- Proteggere dall'esecuzione di software malizioso

25/10/2002

Security Enhanced Linux

40



Limitazione degli accessi diretti ai dati

- Viene definito il dominio *fsadm_t* per controllare *fsck* e utilities simili in quanto accedono al disco in modo diretto
- Viene, inoltre, assegnato il tipo *fsadm_exec_t* ai files dei programmi di queste utilities
- Porzione della configurazione per il dominio *fsadm_t*:

```
allow fsadm_t fsadm_exec_t: process {entrypoint execute};
allow fsadm_t fixed_disk_device_t: blk_file {read write};
allow initrc_t fsadm_t: process transition;
allow sysadm_t fsadm_t: process transition;
```

25/10/2002

Security Enhanced Linux

41



Limitazione degli accessi diretti ai dati

- Viene definito il dominio *klogd_t* per permettere a *klogd* di accedere alle devices di memoria del kernel
- Viene assegnato il tipo *klogd_exec_t* ai files di programma
- Porzione della configurazione per il dominio *klogd_t*:

```
allow klogd_t klogd_exec_t: process {entrypoint execute};
allow klogd_t memory_device_t: chr_file read;
allow initrc_t klogd_t: process transition;
```

25/10/2002

Security Enhanced Linux

42



Protezione dell'integrità del kernel

- Protezione dei files contenuti in */boot*
- Porzione della configurazione per i files in */boot*:

```
allow initrc_t boot_t: dir {read search add_name remove_name};
allow initrc_t boot_runtime_t: file {create write unlink};
type_transition initrc_t boot_t: file boot_runtime_t;
```

25/10/2002

Security Enhanced Linux

43



Protezione dell'integrità del kernel

- Il programma *insmod* viene etichettato con il tipo *insmod_exec_t* e gira nel dominio *insmod_t*
- Porzione della configurazione per il dominio *insmod_t*:

```
allow insmod_t insmod_exec_t: file x_file_perms;
allow sysadm_t insmod_t: process transition;
allow insmod_t insmod_exec_t: process {entrypoint execute};
allow insmod_t sysadm_t: fd inherit_fd_perms;
allow insmod_t self: capability sys_module;
allow insmod_t sysadm_t: process sigchld;
```

25/10/2002

Security Enhanced Linux

44



Protezione dell'integrità dei files di sistema

- Tipi separati per programmi di sistema
 - ✓ Esempio: *bin_t*, *sbin_t*
- Tipi separati per i files di configurazione di sistema
 - ✓ Esempio: *etc_t*
- Tipi separati per le librerie condivise
- Tipi separati per i logs di sistema
- Tipi separati per linker dinamico

25/10/2002

Security Enhanced Linux

45



Restrizioni dei processi privilegiati

- La configurazione limita lo sfruttamento di "difetti" nei processi privilegiati definendo per loro domini separati
- Restringe i loro accessi ai minimi privilegi

25/10/2002

Security Enhanced Linux

46



Separazione dei processi

- Le interazioni tra processi in diversi domini sono limitate
- Controllo dell'accesso ai files temporanei

25/10/2002

Security Enhanced Linux

47



Protezione del dominio dell'amministratore

- Controllo dell'accesso al dominio *sysadm_t*
- Limitazione del dominio ad eseguire solo tipi approvati
- Separazione dagli altri domini

25/10/2002

Security Enhanced Linux

48



Esempio: protezione da codice malizioso

- Confinamento di netscape nel suo dominio:

```
type_transition user_t netscape_exec_t: process user_netscape_t;
allow user_t netscape_exec_t: process {entrypoint execute};
allow user_netscape_t user_netscape_rw_t: file
    {read write create unlink};
```



Prestazioni

- Tests Microbenchmark su kernel base Linux 2.4.2, e su kernel SELinux 2.4.2

Microbenchmark	Base	SELinux	Overhead
file copy 4KB	49.5	48.6	2%
file copy 1KB	40.4	38.6	5%
file copy 256B	23.0	21.0	10%
pipe	6.17	7.17	16%
pipe switching	12.7	15.0	18%
process creation	485	494	2%
exec1	2480	2610	5%
shell scripts (8)	659	684	4%



Prestazioni

Microbenchmark	Base	SELinux	Overhead
mail TO	1.45	1.93	23%
stat	8.06	10.3	28%
open/close	11.0	14.0	27%
0KB create	22.0	26.0	18%
0KB delete	1.72	1.90	10%
fork	499	505	1%
execve	2730	2820	3%
sh	10K	11K	10%
pipe	12.5	14.0	12%
AFLINUX	20.6	24.6	19%
UDP	310	356	15%
RPC/UDP	441	519	18%
TCP	389	425	9%
RPC/TCP	667	726	9%
TCP connect	675	738	9%



Prestazioni

	Base	SELinux	Overhead
elapsed	11:14	11:15	0%
system	00:49	00:51	4%
latency	0.56	0.56	0%
throughput	8.29	8.28	0%



SECURITY ENHANCED LINUX

Università degli Studi di Salerno
 Facoltà di Scienze MM FF NN
 Corso di Laurea in Informatica
 Dipartimento di Informatica ed Applicazioni
 Progetto di:
 Sistemi di Elaborazione dell'Informazione
 (Sicurezza su Reti)
 Prof. Alfredo De Santis
 Anno accademico 2001/2002
 Autori del progetto:
 Michelangelo Magliari matr. 56/00012
 Loredana Luzzi matr. 56/100486
 Andrea Fiore matr. 56/100738