Linux e TCSEC

Daniele Diodati

6 maggio 2010

1 TCSEC

Trusted Computer System Evaluation Criteria (TCSEC) è uno standard emanato dall'United States Government Department of Defense che definisce i requisiti basilari per la valutazione della sicurezza di un sistema informatico.
TCSEC, chiamato anche Orange Book, è la colonna centrale della serie di pubblicazioni Rainbow. Sviluppato inizialmente nel 1983 dal National Computer Security Center (NCSC), un ramo della National Security Agency, e aggiornato nel 1985, è stato sostituito dallo standard internazionale Common Criteria pubblicato nel 2005.
I criteri sono stati sviluppati con tre obiettivi principali:

- Fornire agli utenti un metro di valutazione sulla sicurezza del trattamento di informazioni sensibili su sistemi informatici;
- Fornire informazioni ai fabbricanti di prodotti commerciali, utili al fine di soddisfare i requisiti di fiducia per le applicazioni sensibili;
- Fornire una base per specificare i requisiti di sicurezza nelle specifiche di acquisizione.

1.1 Requisiti

1.1.1 Policy

Ci deve essere un'esplicita e ben definita politica di sicurezza imposta dal sistema, individuati i soggetti e gli oggetti, ci devono essere un insieme di regole che vengono utilizzate dal sistema per determinare se un determinato soggetto può accedere a un oggetto specifico.

1.1.2 Marking

Etichette per il controllo degli accessi devono essere associate agli oggetti. Al fine di controllare l'accesso alle informazioni memorizzate in un computer, secondo le regole di una politica di sicurezza stabilita, deve essere possibile etichettare ogni oggetto identificando in maniera affidabile il suo livello di sensibilità, e/o la modalità di accesso accordato ai soggetti che vi possono accedere.

1.1.3 Identification

I singoli soggetti devono essere identificati. Ogni accesso all'informazione deve essere permesso o impedito in base ai diritti di chi vi accede. L'identificazione e l'autorizzazione deve essere gestita dal sistema informatico ADP (Automatic Data Processing) e deve mediare ogni elemento attivo che svolge una qualsiasi azione *security-relevant* (¡SR) all'interno del sistema.

1.1.4 Accountability

Informazioni di controllo devono essere conservate in modo selettivo e protette in modo che ogni azione SR può essere ricondotta al responsabile (Audit Trail). Capacità di selezionare eventi SR è fondamentale per ridurre al minimo le spese di auditing e per consentire un'analisi efficiente. I dati Audit devono essere protetti da modifiche non autorizzate permettendo l'individuazione e la conseguente indagine sulla violazione della Policy.

1.1.5 Assurance

Il sistema informatico deve contenere meccanismi HW/SW valutabili in maniera indipendente in modo che vengano fornite sufficienti garanzie di Policy, Marking, Identification e Accountability. Per questo l'hardware e il software che svolge tali funzioni deve essere ben identificato. La base del funzionamento di tali meccanismi di sicurezza deve essere chiaramente documentata in modo tale che sia possibile valutare indipendentemente la loro adeguatezza.

1.1.6 Continuous Protection

I meccanismi di sucurezza o Trusted Computing Base (TCB) devono essere costantemente protetti contro manomissioni e/o modifiche non autorizzate. Nessun sistema informatico può essere considerato veramente sicuro se l'hardware di base e il software che controlla la politica di sicurezza sono a loro volta soggetti a modifiche non autorizzate. L'obbligo di una continua protezione ha implicazioni dirette per tutta la vita del sistema.

1.2 Divisioni

1.2.1 Divisione D

Questa divisione contiene una sola classe riservata a quei sistemi che sono stati valutati, ma che non soddisfano i requisiti di una classe superiore di valutazione.

1.2.2 Divisione C Classe 1

TCB soddisfa i requisiti minimi fornendo la separazione di utenti e dati. Esso incorpora una qualche forma di controllo credibile, in grado di far rispettare le limitazioni di accesso su base individuale, vale a dire, apparentemente adeguato per consentire agli utenti di essere in grado di proteggere le informazioni sensibili:

- Discrectionary Access Control;
- Identificazione e Autenticazione (es. username e password);

- Dominio TCB testato e protetto;
- Sistema per test futuri sulla funzionalità;
- Documentazione amministratore, utente, test e design.

1.2.3 Divisione C Classe 2

I sistemi di classe C2 richiedono tutte le proprietà della classe C1, inoltre è richiesto un controllo di accesso a grana più fine, per rendere gli utenti individualmente responsabili delle loro azioni. Questo è svolto attraverso:

- Discrectionary Access Control elevato (single subject - single object);
- Identificazione univoca e Autenticazione obbligatoria;
- Dati audit protetti;F
- Object Reuse: Nessuna informazione sensibile, legata ad azioni precedenti di un soggetto svolte con l'ausilio di un oggetto, deve essere reperibile da ciascun soggetto che ottiene l'accesso allo stesso oggetto, reso nuovamente disponibile dal sistema.

1.2.4 Divisione B Classe 1

I sistemi di classe B1 richiedono tutte le proprietà della classe C2. Inoltre:

- Etichettamento soggetti e oggetti sotto il controllo del TCB;
- Mandatory Access Control;
- Canali di comunicazione I/O singolo livello o multilivello;
- Policy *No Read Up No Write Down* (BLP);
- Isolamento spazio singolo processo;
- Dominio TCB testato (eliminazione delle vulnerabilità) e protetto.

1.2.5 Divisione B Classe 2

I sistemi di classe B2 richiedono tutte le proprietà della classe B1. Inoltre:

- Etichettamento esteso a tutti i soggetti e oggetti del sistema ADP (Automatic Data Processing);
- Canale di comunicazione sicuro tra TCB e utenti per login e autenticazione;
- Stima sulla massima banda di un Convert Channel (canale che permette ad un processo di trasferire informazioni in modo non autorizzato);
- TCB suddivisa in moduli ben definiti e isolati;
- DTLS (Distribuited Top Level Specification): descrizione completa TCB con eccezioni, messaggi di errore e conseguenze;
- CMS (Configuration Management System): mapping consistente sulla documentazione e il codice associato al TCB;
- Sistema relativamente resistente ad attacchi.

1.2.6 Divisione B Classe 3

I sistemi di classe B3 richiedono tutte le proprietà della classe B2. Inoltre:

- Canale sicuro tra TCB e utenti per qualsiasi comunicazione;
- TCB capace di evidenziare un eccesso di violazioni, notificandolo all'amministratore e in caso estremo di effettuare una procedura (la meno distruttiva) per far cessare le violazioni;
- TCB strutturata con meccanismi di protezione semplici, escludendo tutte le componenti non essenziali alla gestione della sicurezza del sistema;
- Trusted recovery: procedura di failback senza compromissione della protezione;
- Sistema fortemente resistente ad attacchi.

1.2.7 Divisione A Classe 1

I sistemi di classe A richiedono tutte le proprietà della classe B2. Inoltre:

- Un modello formale della policy deve essere chiaramente identificato e documentato, compresa una dimostrazione della sua consistenza;
- FTLS (Formal Top-Level Specification): definizoni astratte delle funzioni svolte dal TCB e dei meccanismi HW e/o FW che sono usati per supprtare domini di esecuzione distinti;
- FTLS deve essere consistente con la policy, questo deve essere dimostrato formalmente dove possibile e informalmente nei restanti casi;
- TCB deve essere consistente con il FTLS, questo deve essere dimostrato informalmente;
- Tecniche di analisi formale devono essere utilizzate per identificare e analizzare Covert Channels. L'esistenza di CC residui nel sistema deve essere giustificata.

2 TCSEC e Linux

Linux, come tutti i moderni sistemi operativi di più ampia diffusione è classificato a livello C2, i limiti più significativi sono:

- scarso controllo sull'utilizzo appropriato della memoria e dei meccanismi di comunicazione interprocesso, a vantaggio dell'efficienza;
- complessità del software e conseguente maggiore difficoltà di verifica della correttezza;
- inefficace limitazione dei privilegi di accesso alle risorse, secondo il modello DAC, a vantaggio della semplicità di configurazione;
- assenza di livelli di privilegio intermedi tra l'utente comune e l'amministratore.

Il primo passo per avvicinare la classe B è certamente quello di implementare il Mandatory Access Control, così facendo si avrà:

- controllo granulare su tutto il sistema;
- livelli di privilegio intermedi tra utente e amministratore;
- limitazione spazio esecutivo malware;
- policy di sicurezza del sistema.

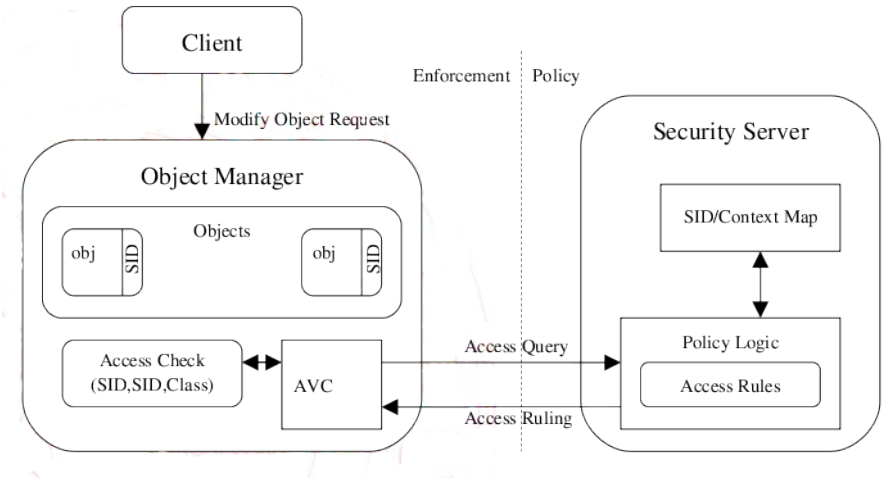
3 SELinux

3.1 Security-enhanced Linux

NSA Security-enhanced Linux è un insieme di patches al Linux kernel e utility per gestire MAC su linux. Esso fornisce un meccanismo per far rispettare la separazione delle informazioni basato su requisiti di riservatezza e integrità, limitando così minacce di manomissione, di elusione dei meccanismi di sicurezza delle applicazioni e possibili danni causati da applicazioni maligne. Esso comprende un insieme di file di configurazione che costituiscono un modello di policy per il sistema.

3.2 Architettura

Quando un soggetto cerca di accedere ad un oggetto, la parte del kernel responsabile all'applicazione della policy (Object Manager) controlla un Access Vector Cache (AVC), dove i permessi degli oggetti e dei soggetti vengono conservati. Se non è possibile prendere una decisione in base ai dati contenuti nella AVC, la richiesta viene inoltrata alla componente decisionale (Security Server), il quale, estrae i dati da una matrice (Context Map). Se l'accesso è rifiutato, viene indicato nel file /var/log/messages.



3.3 Funzionamento

Ogni soggetto/oggetto è classificato per mezzo di

- un numero intero chiamato SID (Security Identifier);
- una terna chiamata Security Context, composta da:
  1. User identity: l'account SELinux associato al soggetto/oggetto, gli account SELinux sono diversi da quelli di sistema, più account di sistema possono essere mappati su uno stesso account SELinux;
  2. Role: il ruolo correntemente utilizzato dall'utente, in ogni istante un utente può utilizzare un solo ruolo;
  3. Type: è l'attributo fondamentale utilizzato da SELinux per prendere le decisioni, si parla di type per gli oggetti e di domain per i soggetti

Le due viste sono messe in relazione 1:1 tramite la Context Map.

Il security server è chiamato a prendere due tipi di decisione:

- decisioni di accesso: determinare se un soggetto può o no svolgere una determinata operazione su di un oggetto;
- decisioni di transizione: determinare quale tipo assegnare ad un oggetto o soggetto all'atto della creazione.

SELinux assume una closed world policy, per cui ogni richiesta di autorizzazione è negata dal security server se non esiste una regola che la consente esplicitamente. Nelle decisioni d'accesso ad ogni classe di oggetti è associato un Access Vector, che contiene una regola per ogni azione definita per la classe. Il security server, interrogato sulla possibilità di effettuare una determinata azione su di un oggetto da parte di un soggetto, restituisce le regole, calcolate in base al dominio del soggetto e al tipo e classe dell'oggetto. Le decisioni di transizione riguardano l'assegnamento di un contesto di sicurezza ai nuovi soggetti ed oggetti:

- Un nuovo processo eredita il contesto del processo padre, ma è possibile una transizione ad un altro dominio:
  - decisa dal processo padre;
  - richiesta dallo stesso processo. [1]
- I nuovi file ereditano il contesto della directory che li contiene, è possibile una transizione ad un altro tipo.

3.4 Modalità

Un sistema con SELinux può essere avviato in tre modi:

- Disabled: nessun controllo degli accessi, nessun log, perdita dei SID associati ai file in caso di modifica;
- Permissive:il controllo degli accessi genera unicamente un log delle decisioni di sicurezza, ma non le mette in atto;
- Enforcing: piena funzionalità del sistema di controllo degli accessi.

[1] Caratteristica fondamentale che distingue SELinux dalle altre soluzioni è la presenza di un'API che consente di scrivere programmi in grado di interagire con il security server, anzichè solamente subire una policy.

File security class										
Append	Create	Execute	Get attribute	I/O control	Link	Lock	Read	Rename	Unlink	Write
?	?	?	?	?	?	?	?	?	?	?

	File security class										
	Append	Create	Execute	Get attribute	I/O control	Link	Lock	Read	Rename	Unlink	Write
Allow	X	X	–	–	–	–	–	–	–	–	–
Auditallow	–	–	–	–	–	–	–	–	–	–	–
Dontaudit	–	–	–	–	–	–	–	–	–	–	–

4 Installazione

SELinux è disponibile per ogni architettura Linux con kernel Vanilla o derivato da questo. Le distribuzioni più favorevoli alla sua installazione, perchè contenenti pacchetti pronti all'uso, sono

- Debian
- Ubuntu
- Fedora
- Gentoo
- Red Hat Enterprise

La scelta è ricaduta su Fedora dato che comprende l'architettura SElinux nei pacchetti d'installazione classica. Inoltre in questa distribuzione è presente un'ottima documentazione per la comprensione del sistema sia a livello utente che a livello amministratore. Queste guide sono reperibili all'indirizzo `http://docs.fedoraproject.org/`.

5 Conclusioni

Con l'utilizzo di SELinux un sistema operativo Linux può essere considerato a livello `B1` nella classificazione descritta nei `TCSEC` .

Riferimenti bibliografici

[1] http://computer-legacy.blogspot.com/.

[2] http://en.wikipedia.org/.

[3] http://web.mit.edu/rhel-doc/4/rh-docs/rhel-rg-it-4/ch-selinux.html.

[4] http://wiki.debian.org/selinux/setup#basics.

[5] http://www.boran.com/security/tcsec.html.

[6] http://www.gentoo.org/proj/it/hardened/selinux/selinux-handbook.xml.

[7] M. Prandini. Implementazione di modelli di sicurezza evoluti nel sistema operativo linux. edenti.deis.unibo.it, 2006.

[8] Ray Spencer, Stephen Smalley, Peter Loscocco, Mike Hibler, David Andersen, and Jay Lepreau. The flask security architecture: system support for diverse security policies. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 11–11, Berkeley, CA, USA, 1999. USENIX Association.