

SELinux 入門初探

Kenduest Lee (小州)

<kenduest@info.sayya.org>

內容大綱

- 傳統 Linux 權限管理概念
- 傳統 Linux 權限管理問題
- SELinux 架構概觀
- SELinux 安全本文
- SELinux 管理與診斷

傳統 Linux 權限概觀

2008 年 10 月

傳統 Linux 權限控管

- Linux 提供一般權限控管方式
 - 核心層級控管 (Kernel control)
 - 行程層級控管 (Process control)
 - 依據使用者執行中的程式判斷存取身份
 - 資源控管 (Resource control)
 - 檔案系統的檔案目錄讀寫
 - 網路 Socket 等存取

傳統 Linux 權限控管

- 決定實際的存取權限方式
 - Process
 - 執行中 process 帶有 Real ID 與 Effective ID
 - 依據 Real 與 Effective ID 決定相關身份
 - Resource
 - 對於檔案、目錄等項目實際應對的 r,w,x 等權限

依據 Process + Resource 決定是否有存取權限

傳統 Linux 權限控管

- 程式執行時候具備身份
 - Real ID 與 Effective ID
 - Real ID，為程式的執行者
 - Effective ID，實際對於系統的存取權限
 - 一般與 Real ID 相同
 - 若是程式本身具備 `setuid`, `setgid` 等權限則會異動

傳統 Linux 權限控管

- 決定程式的身份權限

-rwxr-xr-x	1	root	root	23132	May	24	A
-rwsr-xr-x	1	root	root	24120	May	24	B

- 以 peter 身份執行該程式情況
 - A 程式執行期具備 peter 身份權限
 - B 程式執行期具備 root 身份權限

傳統 Linux 權限控管

- 檔案存取權限是基於身份來判斷
 - 只要行程身份於特定檔案權限部份有 rw 權限，該使用者執行的程式對該檔案便具備 rw 權限

```
-rw-r--r-- 1 peter users 5230 2008-08-31 file.txt
```

owner 權限為 rw-
group 權限為 r--
other 權限為 r--

owner 為 peter
group 為 users

傳統 Linux 權限控管

- 基於使用者身份來判斷權限的問題
 - 使用者執行的 firefox 程式是否應該允許可以存取自己的 ssh private key 檔案？

執行中的程式：

```
kendlee 4608 ? R+ 02:46 0:00 firefox-bin
```

ssh 的 private key 檔案：

```
-rw----- 1 kendlee users 744 2006-05-2 id_dsa
```

傳統 Linux 權限問題

2008 年 10 月

傳統 Linux 權限控管

- 只有 root 與 user 兩種權限分級
 - 以 root 身份執行的程式表示有最大權限，不受到權限規範的範圍
 - 需要用 root 身份啟動與運作的程式可能有意外的風險

傳統 Linux 權限控管

- 行程是允許修改檔案目錄權限
 - 只要使用者為檔案的 owner 即可變更該檔案權限
 - 使用者執行的程式，程式具備任意修改 owner 為自己的檔案目錄權限
- 舉例..
 - mail 與 mutt 有權限可以變更 mailbox 權限

```
-rw----- 1 kendlee users 744 2006-05-2 /var/mail/kendlee
```

傳統 Linux 權限控管

- 行程允許進行系統大部分呼叫無限制
 - 可以透過 signal 架構發送信號給行程
 - 可以透過 socket 有網路存取功能
 - 可以使用 ioctl 進行比較低階系統溝通
 - 可以使用 fork, exec 呼叫產生與建立行程
 - ... (大部分都可以呼叫執行任意使用)

傳統 Linux 權限控管

- 傳統 Linux 權限控管的問題
 - 基於 使用者身份 決定權限，非 個別行程 來決定
 - 無法針對 行程 本身提供細部更別的權限規範機制
 - 以 root 身份運作的身份有最大權限而無相關限制

權限控管規範機制

2008年10月

15

權限控管機制

- 權限控管機制項目
 - DAC (Discretionary Access Control)
 - 自主存取控制
 - 依據當時候運作的身份決定存取權限
 - MAC (Mandatory Access Control)
 - 強制存取控制
 - 依據指定條件決定是否具備可存取權限

權限控管機制

- DAC (Discretionary Access Control)
 - 自主存取控制，為現有作業系統的權限存取控管方式
 - 檔案系統面，提供依據 owner,group,other 所屬身份與 r,w,x 權限來規範
 - root 執行的程式具備最高權限無限制
 - 行程的權限基於使用者身份，無法針對行程本身個別進行額外條件的限制規範

權限控管機制

- **MAC (Mandatory Access Control)**
 - 強制存取限制，為現有作業系統的權限存取控管方式
 - 可以規範個別系統項目進行限制存取，其中包含檔案系統面的存取，行程管理面與網路面等項目
 - 實際應用，行程需要先符合 DAC 規範，並且通過 MAC 的限制規範後才可以存取所需要項目

SELinux 介紹

2008 年 10 月

SELinux 簡介

- 何謂 SELinux ?
 - 全名為 Security-Enhanced Linux
 - 為美國國家安全局 (National Security Agency , NSA) 所開發的安全功能
 - SELinux 提供別於傳統 UNIX 系統以使用者權限控管的作法，可提供更詳盡逐一的授權控管

SELinux 簡介

- 哪些版本可以用 SELinux ?
 - SELinux 的開發主要為 NSA 與 RedHat，最早納入到 Fedora 與 RHEL 版本
 - 目前 Ubuntu 與 SuSE Linux 都宣佈要支援納入 SELinux 功能到系統內

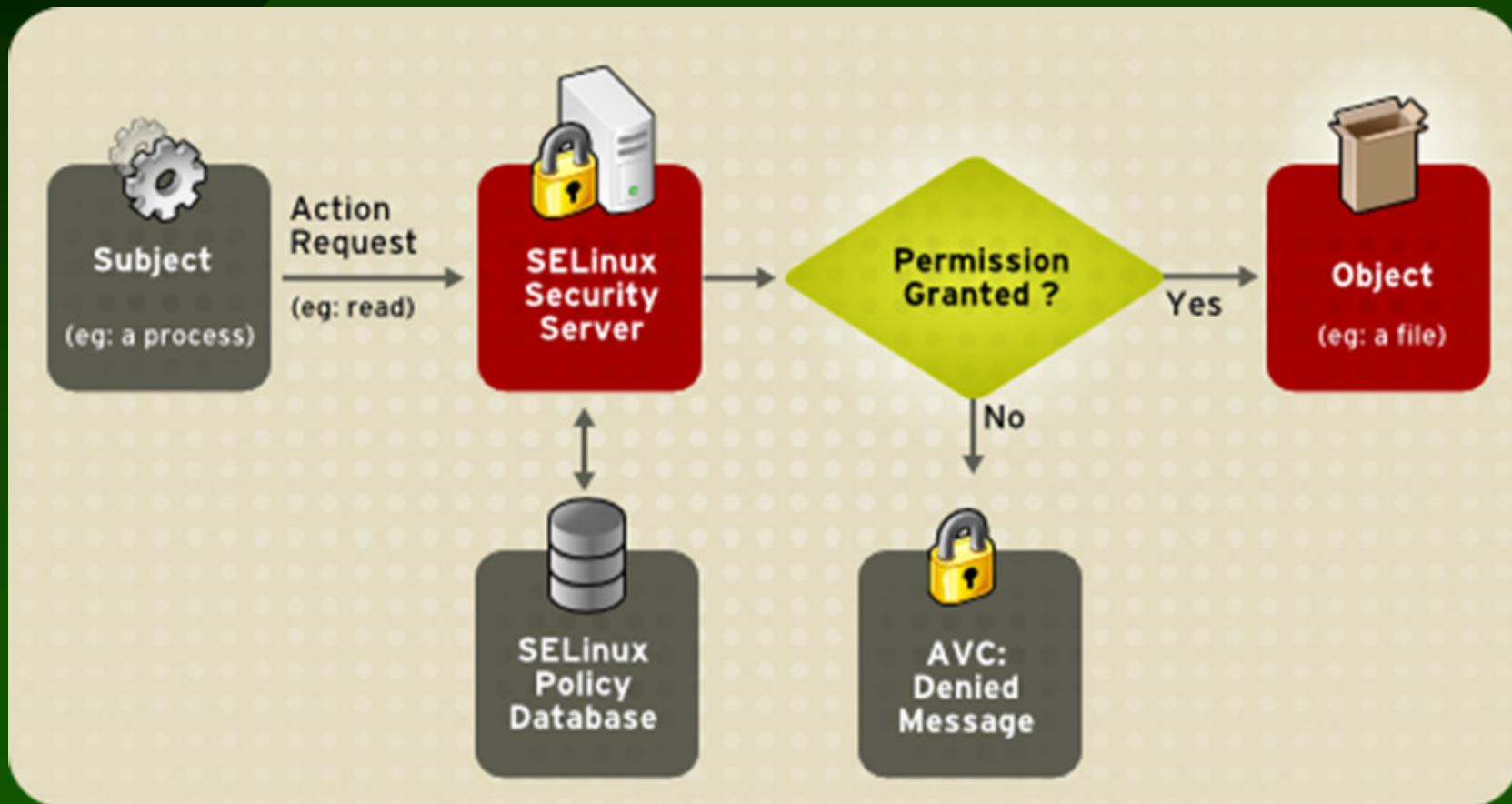
SELinux 簡介

- 許多系統管理者對 SELinux 的觀感？
 - 由於早期 Fedora/RHEL 版本訂立的政策不多，所以能夠保護的服務很少
 - 另外因為政策的編寫規則考量環境太少，導致導致許多網路服務程式無法正常工作，所以不部份的系統都選擇關閉不啟用避免問題
 - SELinux 於 Fedora Linux 5 與 RHEL 5 版本開始已經逐漸穩健，可以預設啟用讓系統正常運作

SELinux 規範基準為何？

- SELinux 提供 Subject 與 Object 來規範
 - Subject
 - 表示要進行存取操作的行程項目
 - 比方提供服務的 ftpd 程式本身
 - Object
 - 表示要存取的項目
 - 包含檔案系統面存取，網路存取與行程溝通等項目

SELinux 運作示意圖



SELinux 架構

SELinux 規範組成

- **規範特性項目**
 - Mandatory Access Control (MAC)
 - Role Based Access Control (RBAC)
 - Type Enforcement (TE)
 - Domain Translation

SELinux 規範組成

- **Mandatory Access Control (MAC)**
 - 明確完整的徹底規範機制
 - Object Classes and Permissions

- 檔案系統面：read, write, unlink, rename, mount... 等
- 網路通訊面：listen, accept, bind... 等
- 行程管理面：fork, signal... 等

SELinux 規範組成

- Role Based Access Control (RBAC)
 - 基於角色的方式來規範有不同權限
 - 可以規範不同使用者有不同的角色項目
 - 切換到特定角色才賦予特定存取權限

SELinux 規範組成

- Type Enforcement (TE)
 - 只有給行程最小的所需運作環境
 - 制定許多 Type 項目，套用於指定檔案與網路項目
 - 規範行程以特定 Domain 運作
 - 最後限制特定的 Domain 能夠存取特定的 Type

SELinux 規範組成

- Domain Translation
 - 規範不同行程的 Domain 轉移方式
 - A Process 規範使用 A-Domain 環境執行
 - B Process 規範使用 B-Domain 環境執行
 - 於 A-Domain 環境執行 B 時，是否要沿用還是轉移

SELinux 啟用

2008 年 10 月

31

SELinux 啟用配置

- GRUB 程式核心傳入參數項目
 - 決定預設 kernel 是否啟用 selinux 支援
 - 配置參數：`selinux=[0|1]`
 - grub 的 `/boot/grub/menu.lst` 配置
 - `kernel /boot/vmlinuz root=/dev/sda1 selinux=1`

SELinux 啟用配置

- **設定檔案**
 - /et/sysconfig/selinux 或是 /etc/selinux/config
 - 內容組成
 - SELINUX=[enforcing | permissive | disabled]
 - SELINUXTYPE=[targeted | strict | mls]

SELinux 啟用配置

- 項目組成說明
 - SELINUX=[enforcing | permissive | disabled]
 - enforcing) 強制模式，並提供限制存取機制
 - permissive) 寬容模式，以警告代替強制規範
 - disabled) 關閉

指定預設 SELinux 啟用的模式環境

SELinux 啟用配置

- 項目組成說明
 - SELINUXTYPE=[targeted | strict | mls]
 - targeted) 僅保護特定有策略規則的網路程式
 - strict) 完整保護受限模式
 - mls) mls 保護受限模式

指定要載入使用的政策資料庫項目

SELinux Relabel

- 檔案系統重新標上安全本文資訊
 - SELinux 將檔案目錄的“security context”紀錄於檔案系統上，變更 policy 項目需要重新套用。
 - 作法
 - **touch /.autorelabel ; reboot**

SELinux 狀態管理

2008 年 10 月

37

SELinux 狀態查詢

- 檢視目前 SELinux 配置狀態

- 使用 `sestatus` 命令

- 狀態輸出

- SELinux status: enabled
- SELinuxfs mount: /selinux
- Current mode: enforcing
- Mode from config file: permissive
- Policy version: 21
- Policy from config file: targeted

SELinux 狀態查詢

- 切換目前的運作模式
 - 使用 `setenforce` 命令
 - 使用方式
 - `setenforce [1 | 0 | Enforcing | Permissive]`

SELinux 安全本文

何謂 SELinux 安全本文？

- Security Context
 - SELinux 處理的基礎項目為 Subject 與 Object
 - Subject：特定身份運作執行的程式項目
 - Object：被存取的檔案目錄、網路 socket
 - 每個 Subject 與 Object 都帶有 security context，後續SELinux 實際依據該項目規範某個行為是否可以存取

SELinux 檢視與組成

- 檢視 Security Context
 - 程式項目
 - id -Z
 - ls -Z , ls --scontext
 - ps Z , ps -Z , ps auxZ
 - netstat -Z

SELinux 檢視與組成

- 檢視 Security Context
 - id -Z
 - root:system_r:unconfined_t
 - user_u:system_r:unconfined_t

SELinux 檢視與組成

- 檢視 Security Context
 - ls -Z

```
drwxr-xr-x root root system_u:object_r:bin_t bin
```

SELinux 檢視與組成

- 檢視 Security Context
 - ps -Z

LABEL	PID	...	STAT	TIME	CMD
system_u:system_r:init_t	1	...	Ss	0:00	init [5]

SELinux 檢視與組成

- Security Context 組成
 - 主要格式
 - identity:role:type

SELinux 檢視與組成

- Security Context 組成
 - 其他擴增項目
 - MLS (Multi Level System)
 - user:role:type:sensitivity:compartments
 - MCS (Multi Category System)
 - user:role:type:sensitivity:category
 - 其中 sensitivity 固定為 s0

SELinux 檢視與組成

- **identity**
 - 近於於系統使用者帳號，提供給 **selinux** 識別
 - 帳號登入主機後，不管程式切換到那個身份帳號，該 **identity** 都固定相同，一般搭配 **role** 使用
 - **root** 表示 **root** 帳號身份
 - **system_u** 表示系統行程
 - **user_u** 表示一般使用者帳號

SELinux 檢視與組成

- **role**
 - 角色身份
 - 可以依據不同的角色賦予不同權限

SELinux 檢視與組成

- **role**
 - 檔案目錄，一般為 `object_r`
 - 行程資訊，一般為 `system_r`
 - 使用者
 - targeted policy 下為 `system_r`
 - strict policy 下細分 `sysadm_r`, `staff_r`, `user_r`

SELinux 檢視與組成

- **role**
 - 類似於群組，不同角色可以具備不同身份權限
 - 可以具備多個，但是同一時間內只能夠使用一個 role
 - 在 targeted 的 selinux 環境下，該項目沒實質功能

SELinux 檢視與組成

- **type**
 - SELinux 內的 type 又可以稱呼為 domain
 - 檔案目錄, 行程皆有一個 type 標示規範項目
 - 一般檔案目錄主要稱呼為 type
 - 執行中的行程主要稱呼為 domain
 - 可以依據 type 相關組合來限制可以存取資源項目

SELinux 檢視與組成

- 檢視 policy 內 security context 設定
 - 使用 sestatus 程式
 - sestatus -a
 - sestatus -a -s ftpd_t

SELinux Boolean Value

SELinux Boolean Value

- **Boolean Value**
 - SELinux 歸範了許多規則項目，可以透過調整 boolean value 達成開啓與關閉功能
 - 實際 boolean value 檔案於 /selinux/bootleans 目錄
 - 可以透過 **getsebool** 與 **setsebool** 管理

SELinux Boolean Value

- **getsebool**
 - 說明：列出所有 selinux bool 值清單列表與內容
 - 使用方式： `getsebool [-a]`
 - 使用範例
 - `getsebool ftpd_disable_trans`
 - `getsebool -a`

SELinux Boolean Value

- **setsebool**
 - 說明：修改指定的 selinux bool 內容
 - 使用方式：`setsebool [-P] name=value`
 - 使用範例
 - `setsebool -P ftpd_disable_trans=on`

SELinux Boolean Value

- **實驗測試**
 - 啓用 ftpd 服務後，使用一般帳號登入主機
 - 預設拒絕存取使用者家目錄
 - 調整 ftp_home_dir 該 boolean value 數值解決

SELinux 檔案安全本文修改

SELinux 檔案安全本文

- 變更檔案 security context
 - 程式項目
 - chcon
 - restorecon
 - fixfiles

SELinux 檔案安全本文

- **chcon**
 - 變更檔案目錄的 security context
 - **chcon -t var_t /etc/vsftpd/vsftpd.conf**
 - **chcon --reference=/var/www/html index.html**

SELinux 檔案安全本文

- **restorecon**
 - 回存檔案目錄的 security context
 - 依據 /etc/selinux/<POLICY>/contexts/files/ 內 file_contexts 與 file_contexts.local 恢復
 - **restorecon filename**

SELinux 檔案安全本文

- **fixfiles**
 - 回存檔案目錄的 security context
 - 為 shell script，支援更多功能
 - 使用方式
 - `fixfiles { check | restore[-F] relabel } [[dir] ...]`
 - `fixfiles -R rpmpackage[,rpmpackage...] { check | restore }`

SELinux 檔案安全本文

- **實驗測試**
 - ftp 服務匿名登入後，無法上傳檔案於開放目錄
 - 調整該目錄為 `public_content_rw_t` 即可允許

SELinux Domain 轉移

2008 年 10 月

65

SELinux Domain 轉移

- 服務啟動是否要規範於 selinux 環境
 - 依據 selinux 安排，執行位於 /etc/init.d/ 內 script 檔案都帶有 initrc_exec_t 該 type，所以後續該 script 執行的程式只要程式檔案就會轉移進入該 domain 環境加以規範
 - 服務若是單純手動執行 /usr/sbin/xxx 則不受到 selinux 使用 targeted policy 的政策規範限制

SELinux 問題診斷工具

2008 年 10 月

67

SELinux 問題診斷

- 問題檢視方式
 - 啟用 auditd 與 setroubleshoot 服務
 - 可以提供 X Window 環境檢視問題點
 - 使用 `sealert -l <msg-id>` 檢視問題點
 - 使用 `audit2way < /var/log/audit/audit.log` 檢視問題點

SELinux 問題診斷

- **處理問題方式**
 - 是否可以透過 boolean value 調整變更處理
 - 是否修正檔案目錄的 security context 處理
 - 是否可以透過 semanage 修正解決問題
 - 是否關閉該服務程式的 selinux 限制保護

SELinux 問題診斷

- 其他常見問題

- 更換相關 daemon port 導致程式無法啟用
- 可以透過 semanage 程式調整變更
- 常見 web server 設定項目
 - `semanage port -a -t http_port_t -p tcp 81`

其他相關專案計畫

2008年10月

其他相關計畫

- **相關計畫站台**
 - **smack**
 - <http://www.schaufler-ca.com/>
 - **apparmor**
 - <http://www.novell.com/linux/security/armor>
 - **TOMOYO Linux**
 - <http://tomoyo.sourceforge.jp/>