TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 1802R022 – Informatika a logistika

SELinux

Security-Enhanced Linux

Bakalářská práce

Autor: **Miroslav Fidranský**
Vedoucí práce: Mgr. David Kmoch

V Liberci 18. 5. 2007

Originál zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce.

Datum 18.5.2007

Podpis

Abstrakt

Tato bakalářská práce se zabývá Security-Enhanced Linuxem, což je moderní nástroj na ochranu operačního systému. Integruje v sobě několik bezpečnostních modelů, které se starají o kompletní řešení řízení přístupu. Mezi základní modely patří vynucení obsahu a RBAC, což je řízení přístupu založené na uživatelských rolích. Díky možnostem tvorby vlastních bezpečnostních nastavení můžeme systém odladit podle potřeb, které jsou na něj kladeny.

Součástí bakalářské práce je všeobecné seznámení s architekturou SELinuxu, základními vlastnostmi a nastavením bezpečnostních politik. Následuje zkouška generování vlastního zabezpečení a rozbor řešení již hotové politiky.

Řešení na bázi řízení přístupu je dnes již standardem v zabezpečení operačního systému Linux a SElinux je jednou z hlavních implementací takového zabezpečení.

Klíčová slova: SELinux, řízení přístupu, RBAC, vynucení obsahu, bezpečnostní kontext

Abstract

This bachelor's thesis is aimed to Security-Enhanced Linux, the modern tool for operating system security. This tool integrates number of security models which provides complete access control solution. The most important models are Type Enforcement and Role-Base Access Control (RBAC), this system provides access controls based on role play. Thanks to possibilities of making its own security configuration we can debug the system in accordance with the needs which are required.

Another part of this bachelor's thesis deal with general introduction to the SELinux architecture, its general features and security policy configuration. I realize also an exam of generating security policy and analysis of ready-made security policy.

The solution based on access control is in these days rated as standard for Linux operating system security policy, and SELinux is one of the most important security implementation.

Keywords: SELinux, access control, RBAC, type enforcement, security context

Obsah

Úvod

Vzhledem ke každoročně se zvyšujícímu počtu útokům na počítačové systémy je stále nutné hledat nové možnosti zabezpečení. Důležitou součástí ochrany počítačových systémů jsou pravidelné aktualizace. Pokud však záplata není vydána (protože se jedná o nově zjištěnou chybu) nebo systémový administrátor toto nebezpečí zanedbává, je takový systém snadným terčem útočníka. Pokud je útok úspěšný, záleží pak již jen na samotné ochraně systému, kam až bude útočníkovi umožněn přístup.

Základní myšlenkou ochrany systému pomocí Security-Enhanced Linuxu je omezení systémových práv a uzavření běžících služeb do omezeného prostoru nazývaného doména. V tomto prostoru funguje proces bez jakýchkoliv problému a pokud potřebuje vykonat operaci mimo stanovený prostor, musí být tato operace výslovně povolena v nastavení. Díky tomuto omezení je zbytek systému ochráněn a pokud se proces ovládaný útočníkem pokusí ovládnout celý systém, nebude úspěšný.

1 SELinux – vlastnosti

1.1 Historie

Během devadesátých let byl v americké National Security Agency (NSA) zahájen vývoj systému se začleněným povinným řízením přístupu. Na konci roku 2000 byl kód uvolněn jako open source a na vývoji se začala podílet komunita Linuxových vývojářů. Dodnes se NSA na vývoji částečně podílí.

Jako oficiální součást systému začaly SELinuxu dodávat firmy Red Hat a SUSE. Z nekomerčních distribucí to jsou Fedora Core, Debian GNU/Linux a Gentoo Linux.

1.2 Řízení přístupu

Většina dnešních operačních systémů má zabudovaný některý ze systémů řízení přístupu, který umožňuje povolit nebo zakázat přístup k datům. Nejčastěji používané jsou DAC a MAC.

1.2.1 DAC - Nepovinné řízení přístupu (Discretionary Access Control)

Každý objekt v systému má svého vlastníka, který rozhoduje o tom, jak bude s objektem zacházeno a kdo další k němu bude mít přístup. Spuštěné programy uživatele běží s jeho oprávněním a mohou proto přistupovat tam, kam má přístup uživatel.

Tato technika je jako základní ochrana u všech systémů UNIXového typu a v dnešní době již není dostačující. Velké nebezpečí může vzniknout, pokud je napadená služba spustěna s právy roota a útočník má potom přístup na celý systém.

Příklad DAC:

#ls -al /home/test.txt

-rw-r--r-- 1 franta ftp 0 dub 26 17:54 /home/test.txt

Vlastníkem souboru /home/test.txt je uživatel franta a patří skupině ftp. Do souboru může zapisovat pouze jeho vlastník a číst ho může pouze on sám a celá jeho

skupina.

1.2.2 MAC - Povinné řízení přístupu (Mandatory Access Control)

Přístupová pravidla jsou definována v systému a vlastník nemá možnost přístup ovlivnit. Program běží ve svém uzavřeném prostoru a nemůže ohrozit ostatní části systému, pokud k nim nemá vyhrazen přístup.

Příklad MAC:

 # ls -Z /home/uzivatel/test.txt

-rw-r--r-- franta ftp user_u:object_r:user_home_t /home/test.txt

K řízení přístupu DAC je přidán bezpečnostní kontext, skládající se z částí uživatel user_u, role object_r a typu user_home_t.

SELinux vylepšuje DAC v Linuxu právě o MAC a díky tomu lze celý systém lepé ochránit. Při vyhodnocování nejdříve systém prověří práva na úrovni DAC, pokud je přístup umožněn, následuje kontrola pomocí SELinuxových modůlu vynucení obsahu a RBAC. Pokud i tato kontrola v pořádku proběhne, je přístup povolen.

1.3 LSM (Linux Security Modules)

První implementace SELinuxu do Linuxového jádra byla uskutečněna ve verzi jádra 2.2, která později byla upravena pro jádra verze 2.4. S příchodem jádra 2.6 a jeho novým rozšířením o bezpečnostní moduly byl SELinux modifikován pro tento bezpečnostní framework. SELinux je témeř jediné zabezpečení Linuxu, které je implementováno právě přes LSM.

1.4 Přehled bezpečnostního modelu SELinuxu

1.4.1 Subjekty a objekty

Subjektem je v terminologii SELinuxu nazýván proces, který chce vykonat

určitou operaci na objektu, kterým může být například soubor. SELinux definuje několik tříd objektů, mezi které patří: adresář, soubor, odkaz a mnoho dalších. Každý objekt má definován seznam akcí, které může proces s objektem vykonat, bývá nazýván operace. [1] Pro objekt soubor jsou to následující operace:

- Append
- Create
- Execute
- Get attribute
- I/O control
- Link
- Lock
- Read
- Rename
- Unlink
- Write

1.4.2 Bezpečnostní kontext (Security Contexts, SC)

SELinux vytváří ke každému subjektu a objektu vlastní bezpečnostní kontext, který se skládá ze 3 částí:

- uživatelská identita
- role
- typ (doména)

Uživatelská identita

Vychází z uživatelských účtů SELinuxu, které jsou asociované se subjektem nebo objektem. V případě subjektu značí účet, pod kterým proces beží. V objektu

obsahuje identitu uživatele, který daný např. soubor vlastní.

Uživatelské účty SELinuxu jsou oddělené od účtů systému (uložených v /etc/passwd), což umožňuje spravovat jednotlivé účty separátně a díky tomu zjednoduší administraci. Zároveň je možné mít spojených několik klasických uživatelských účtů s jedním účtem SELinuxu.

V základním nastavení SELinuxu existují tři uživatelské identity: root, user_u a system_u. Uživatelská identita root je přiřazena přihlášenému uživateli s právy roota. Identita user_u je přiřazena ostatním uživatelům a systémová identita system_u je přiřazena běžícím procesům.

Role

Role definuje možnosti přístupu k danému uživatelskému účtu. Uživatelé mohou svoji roli změnit za jinou s použitím příkazu newrole, pokud je to tak umožněno v nastavení SELinuxu. Role se většinou pozná podle názvu končícího na _r.

Klasický užitavel má přednastavenou roli user_r a administrátor system_r. Role system_r umožňuje přepnutí do role sysadm_r, pod kterou je možné provádět úpravy SELinuxu.

U většiny objektů není definice role vůbec potřebná vzhledem k jejich využití, ale z důvodu zachování kontextu skládajícího se ze 3 částí je takovýmto objektům přiřazena role object_r.

Typ (doména)

Označení typ či doména mají v SELinuxu totožný význam. Označení doména se častěji používá ve spojení s procesem, naopak typ se používá, pokud se uvažuje objekt. Aby se typ odlišil od uživatele a role, je mu dána přípona _t.

Typ je základním bezpečnostním atributem SELinuxu, který je řešen v bezpečnostním mechanismu. Většinou je v systému definováno několik málo uživatelů a rolí, ale velké množství typů.

SELinux ukládá bezpečnostní kontext v jedné tabulce, kde celý řádek je označen pomocí bezpečnostního identifikátoru (SID, Security Identifier), který je uložen jako číslo (integer), a proto s ním SELinux může pracovat rychleji.

1.4.3 Přechodné a stálé objekty

Jako přechodné objekty jsou nejčastěji označovány procesy, které mají podle svého určení omezenou dobu platnosti. Pro tyto časově omezené objekty se bezpečnostní identifikátor ukládá pouze v paměti, protože je pravděpodobné, že bude častěji používán. Při ukončení systému bude proces ukončen, takže identifikátor již nebude potřebný.

Stálé objekty jsou takové, které v systému zůstávají v delším časovém období (např. soubory nebo adresáře). Jejich bezpečnostní identifikátor se proto ukládá na souborový systém, protože pouhé uložení v paměti by při restartu systému všechny tyto zaznámy zrušilo. U souborových systémů, které mohou být využity pro uchovávání bezpečnostního identifikátoru, je nutná podpora rozšířených atributů (EA). Tyto atributy standartně podporuje souborový systém Ext2, Ext3 a XFS. U systému ReiserFS je nutná úprava. [1]

1.5 SELinux módy

Pokud je v systému SELinux nainstalován a není zakázan, pak se musí nácházet v jednom ze dvou základních stavů:

- permissive
- enforcing

1.5.1 Permissive

Pokud SELinux běží v módu permissive, jeho možnosti ochrany jsou velmi omezeny. Věškeré činnosti jsou povoleny a jediné, co se v tomto módu provádí, je logování. Tento mód se většinou spouští pouze pokud chce administrátor vyladit některé politiky, a proto není určen k ostrému nasazení. Z bezpečnostních důvodů lze dokonce tento mód při kompilaci zakázat a pokud je taková kompilace provedena a SELinux nainstalován, potom běží pouze v módu enforcing.

1.5.2 Enforcing

V tomto módu jsou aktivní všechny metody ochrany systému a záleží tedy pouze

na konfiguraci politik, jak bude systém fungovat. Pokud operace není přesně povolena a její výstup není potlačen (dontaudit), dochází stejně jako u módu permissive k zápisu do logovacího souboru.

1.6 Sledování SELinuxu

Pokud je provedena nepovolená operace nebo je provedena operace s nastavenou hodnout auditallow, dojde k zapsání informací o takovéto operaci do systémového logu. Zároveň je do logovacího souboru zapisována informace o načtení nových politik při startu systému nebo rekompilaci politik.

Formát zápisu operace je vysvětlen na následujícím příkladu. Soubor info.php v domovském adresáři serveru httpd má chybný bezpečnostní kontext, jako typ je nastaven user_home_t (typ pro soubory v domácím adresáři uživatele), správně by měly mít soubory pro zpracovaní webovým serverem kontext s typem httpd_sys_content_t.

Při vyvolání souboru info.php přes webový prohlížeč je vrácena odpověď 403 Forbidden, takže httpd server nemá k souboru info.php umožněn přístup a do logovacího souboru se zapíše následující hodnota.

 # tail -n 1 /var/log/messages
Apr 30 05:25:19 selinux2 kernel: audit(1177903519.888:188): avc: denied { getattr } for pid=979 comm="httpd" name="info.php" dev=dm-0 ino=523274 scontext=root:system_r:httpd_t tcontext=user_u:object_r:user_home_t tclass=file

Vysvětlení jednotlivých částí logovacího záznamu:

Apr 30 05:25:19

Časové razítko zápisu hodnoty do logovacího souboru.

selinux2

Název počítače, na kterém běží SELinux.

kernel: audit(1177903519.888:188):

V závorce je uvedeno časové určení provedení operace. První číslo udává UNIXové časové razítko, po tečce následuje upřesnění času v milisekundách a jako poslední je uvedeno sériové číslo inkrementující se při každém zápisu do logu. [4]

avc: denied { getattr }

Výsledek AVC, v tomto případe denied, takže přístup byl odmítnut a operace nebyla povolena. Pro definici auditallow by zde bylo uvedena hodnota granted. [4]

V závorce je uvedena operace, která měla být provedena, v tomto případě getattr značí získání atributů uvedeného souboru.

pid=979

Číslo procesu, který operaci vyvolal.

comm="httpd"

Příkaz, který operaci vyvolal, v tomto příkladu se jedná o proces serveru httpd.

name="info.php"

Specifikace příkazu. V tomto případě se jedná o cílový soubor, na který se httpd pokusilo přistoupit.

dev=dm-0

Označení zařízení, na kterém se soubor nachází.

ino=523274

Číslo i-uzlu identifikující soubor nebo adresář.

scontext=root:system_r:httpd_t

Bezpečnostní kontext souboru (procesu), který o operaci požádal.

tcontext=user_u:object_r:user_home_t

Bezpečnostní kontext u cílového souboru.

tclass=file

Třída objektu cílového souboru. Podle hodnoty file se jedná o soubor.

1.7 Vynucení obsahu (TE, Type Enforcement)

Základní jednotka bezpečnostního modelu SELinuxu je vynucení obsahu. Stará se o to, aby každý proces byl spojen s doménou, pod kterou běží a každý soubor měl přiřazen typ, podle kterého se bude určovat jeho přístup. Od tradičního vynucení obsahu se SELinux liší v tom, že spojuje doménu a typ do jedné položky. Zároveň nedochází k propojení mezi SELinuxovým uživatelem a doménou s běžícím procesem, ale tuto vlastnost zajišťuje RBAC.

1.8 RBAC (Role-Base Access Control)

Druhá metoda zabezpečení v SELinuxu vychází z RBAC modelu, který lze rozdělit na 3 hlavní části.

Každý uživatel má přidělené role, ve kterých se může nacházet. Jeden uživatel může mít přířazeno více rolí, ale v jednom okamžiku může být přiřazen pouze k jedné.
Příklad nastavení:
user user_u roles { user_r sysadm_r system_r };

Pro uživatele user_r jsou definovány role user_r, sysadm_r a system_r.

Přechody mezi jednotlivými rolemi musejí být také povoleny, jinak přechod není v rámci zabezpečení SELinuxu možný.
Příklad nastavení:
allow system_r sysadm_r;

Umožňuje přechod mezi system_r (standartně přířazena uživateli root po přihlášení) do role sysadm_r.

Poslední částí je přidělování role jednotlivým typům, kdy opět může být jedna role přiřazena více typům, ale vše musí být uvedeno v nastavení SELinuxu.

Příklad nastavení:

role system_r types syslogd_t;

Doméně logovacího systému syslog nastaví roli systémového uživatele.

1.9 Politiky SELinuxu

Pro úpravu politik SELinuxu jsou nutné jejich zdrojové kódy. Při jejich kompilaci se vytváří binární soubor, který je poté možno načíst do paměti a nové nastavení politik SELinuxu bude funkční. Konfigurační soubory se dělí na dva základní typy:

- soubory s nastavením obsahu
- soubory s vynucením obsahu

1.9.1 Soubory s nastavením obsahu

Soubory s koncovkou fc slouží k definici bezpečnostního kontextu. V souborech je uvedena vždy plná cesta k souboru nebo složce (lze využít i regulárních výrazů) a k nim přiřazený bezpečnostní kontext.

Při vytváření nových souborů nebo po spuštění příkazu obnovení bezpečnostního kontextu se informace o souborech načítají ze souboru file_contexts, který vznikne během kompilace sloučením všech souboru s příponou fc.

Příklad:
Souboru /usr/libexec/mysqld bude přiřazen bezpečnostní kontext system_u:object_r:mysqld_exec_t.

/usr/libexec/mysqld -- system_u:object_r:mysqld_exec_t

1.9.2 Soubory s vynucením obsahu

Základní částí konfigurace politik jsou makra a příkazy založené na vynucení obsahu.

Makra

SELinux umožňuje vytvářet makra pomocí jazyku m4, což je UNIXový makroprocesor. Pomocí maker je tvorba politik pro SELinux výrazně zjednodušena, jedná se především o možnosti využít již hotových maker anebo také tvorba maker vlastních, které šetří čas při tvorbě a vytváří se pomocí nich přehlednější kód.

Pravidla

type

Deklarací typu lze určit jeho vlastnosti.

Příklad:

Definice typu mysqld_db_t, atribut file_type určuje, že se jedná o soubor a sysadmfile povoluje přístup k tomuto souboru pouze roli sysadm_r.

type mysqld_db_t, file_type, sysadmfile;

allow

Pravidlo allow povoluje operaci mezi subjektem a objektem. Přístup není zapsán do logu.

V zápisu pravidla se u subjektu specifikuje doména, u objektu typ, třída a operace. Pokud je potřeba uvést více operací, lze je zapsat do složené závorky.

Příklad:

Povoluje procesu pracujícím v doméně mysql_d zjistit atributy a číst soubor, který má nastaven typ usr_t.

allow mysqld_t usr_t:file { getattr read };

auditallow

Zápis je totožný s allow, ale pravidlo auditallow má jiný význam. Pokud je vyvolána operace spadající pod definici tohoto pravidla, operace není povolena, ale je pouze zaznamenána do logu. Pokud je tedy potřeba logovat operaci a zároveň ji mít povolenou, musí se zapsat obě pravidla a v logovacím souboru bude akce zaznamenána

s výsledkem granted.

dontaudit

Pravidlo dontaudit zakazuje vypisování nepovolených operaci do logovacího souboru pro vybrané operace mezi subjektem a objektem. Slouží především pro nedůležité operace, které není potřeba mít povolené a zároveň není vhodný jejich opakující se záznam v logovacím souboru. Zápis je opět shodný s pravidlem allow.

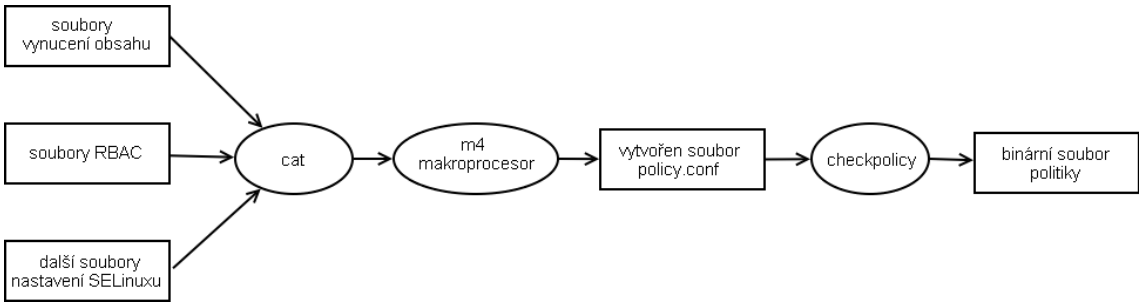
neverallow

Pravidlo neverallow se běžně nepoužívá v souborech s vynucením obsahu, ale zapisuje se do assert.te souboru. Jeho význam spočívá ve stálém zakázaní určitých operacích, takže hlídá možné bezpečnostní nedostatky v zápisu pravidel. Pokud by se dostalo pravidlo allow do sporu s neverallow, kompilace politiky nás na tento problém upozorní a je nutná oprava pravidla allow.

1.9.3 Kompilace

V adresáři se zdrojovými soubory se náchází soubor Makefile, který zabezpečuje kompilaci a načtení politiky do paměti. Příkaz make lze spustit s některým z následujících atributů: [2]

- policy - provede pouze kompilaci politik
- install - provede se kompilace a instalace politik, ale do paměti nebudou načteny
- load - příkaz vykoná kompilaci, instalaci a načtení nově zkompilovaných politik do paměti
- reload - shodné s load
- relabel - znovu načte bezpečnostní kontext ke všem souborům na disku podle aktuální politiky



Obr. 1 – Průběh kompilace politik

Proces kompilace se skládá z několika kroků, jak je naznačeno na obrázku 1, nejdřívě se spojí všechny konfigurační soubory do jednoho souboru, který je následně zpracován makroprocesorem m4 a je vygenerován soubor policy.conf. V tomto souboru jsou již všechny direktivy v základním tvaru a není doporučeno do něj zasahovat. Posledním krokem kompilace je zavolání programu checkpolicy, který vytvoří binární soubor. Tento binární soubor je následně podle volby kompilace uložen do adresáře se zrojovými kódy nebo instalován na místo původního binárního souboru politik.

Binární soubor se ukládá pod názvem policy.číslo, kdy za číslo je dosazena verze politik.

2 Příkazy SElinuxu

2.1 Vlastní příkazy SELinuxu

SELinux obsahuje několik příkazů, které slouží k nastavení a kontrole.

2.1.1 chcon

U specifikovaných souboru mění bezpečnostní kontext, lze kopírovat nastavený bezpečnostní kontext z již existujícího souboru pomocí odkazu na něj.

Příklad:

 # chcon -t etc_t /home/test.txt

Nastaví souboru text.txt typ na etc_t.

2.1.2 checkpolicy

Checkpolicy je program, který provádí kontrolu a následnou kompilaci konfiguračního souboru politik do binární formy, kterou lze následně načíst do paměti. Nespouští se však samostatně, ale je spuštěn při kompilaci pomocí příkazu Makefile.

2.1.3 fixfiles

Příkaz slouží pro obnovení bezpečnostního kontextu u souborů a složek.

Příklad:

 # fixfiles relabel /home/

/sbin/restorecon reset context /home/test.txt:user_u:object_r:etc_t ->

system_u:object_r:user_home_t

2.1.4 getenforce

Vrací současný mód, ve kterém se SELinux nachází. Možnosti jsou: enforcing, permissive nebo disabled.

Příklad:

 # getenforce

Enforcing

2.1.5 getsebool

Zobrazí stav nastavených policy booleans.

 # getsebool mysqld_disable_trans

mysqld_disable_trans --> inactive

2.1.6 newrole

Slouží k přepnutí do jiné role, než ve které se uživatel nachází. Při spuštění dojde k otevření nového shellu. Lze specifikovat i typ, do kterého se má uživatel přepnout, pokud není typ uveden, nastaví se typ, který náleží dané roli.

 # id -Z

root:system_r:unconfined_t

 # newrole -r sysadm_r -t unconfined_t

Authenticating root.

Password:

 # id -Z

root:sysadm_r:unconfined_t

2.1.7 restorecon

Vykoná obnovení bezpečnostního kontextu pro vybraný soubor z nastavení definovaného v konfiguraci politiky.

Příklad:

 # restorecon /home/test.txt

2.1.8 run_init

Spouští init skript se správným bezpečnostním kontextem.

2.1.9 sestatus

Sestatus zobrazí stav v jakém se SELinux nachází včetně podrobností o použitém módu, verzi policy a nastavení policy booleans.

Příklad:

 # sestatus

SELinux status: *enabled*
SELinuxfs mount: */selinux*
Current mode: *enforcing*
Mode from config file: *enforcing*
Policy version: *18*
Policy from config file: *targeted*
Policy booleans:
allow_syslog_to_console *inactive*

2.1.10 setenforce

U systému se spuštěným SELinuxem slouží k přepínání mezi módy enforcing a permissive.

Příklad:

 # getenforce
Enforcing
 # setenforce 0
 # getenforce
Permissive

2.1.11 setsebool

Slouží k přepínání hodnot policy booleans.

Příklad:

mysqld_disable_trans inactive
 # setsebool mysqld_disable_trans true
mysqld_disable_trans active

2.1.12 seuser

Příkaz seuser a jeho podpříkazy seuseradd, seuserdel slouží k přidávání, editaci a mazání uživatelských účtů SELinuxu.

2.1.13 togglesebool

Otočí nastavenou hodnotu boolean policy.

2.2 Upravené příkazy Linuxu

Mnoho strandartních Linuxových příkazů musí být v případě používání SELinuxu modifikováno, ve většině případů se jedná o rozšíření programu o podporu bezpečnostního kontextu.

2.2.1 cp

Příkaz vytváří kopii souboru, pokud je proveden s příznakem určujícím bezpečnostní kontext, bude mít zkopírovaný soubor toto nastavení, jinak se bezpečnostní kontext určí z cílového adresáře.

2.2.2 id

Zobrazuje informace o uživateli, upravená verze zobrazuje i údaj o bezpečnostním kontextu.

Příklad:

 # id -Z

context=root:system_r:unconfined_t

2.2.3 ls

Pokud je uveden přepínač -Z, ve výpisu se zobrazuje i bezpečnostní kontext.

2.2.4 mv

Při přesouvání souboru se nemění jeho bezpečnostní kontext, což může zapříčinit problémy s jeho pozdějším fungováním.

2.2.5 ps

Zobrazuje informace o běžících procesech, pokud je spuštěn s volbou -Z, zobrazuje k jednotlivým procesům jejich bezpečnostní kontext.

Příklad:

```
# ps -Z
LABEL                          PID   TTY     TIME         CMD
root:system_r:unconfined_t     27495 pts/0   00:00:01     bash
root:system_r:unconfined_t     28127 pts/0   00:00:00     ps
```

3 Vytváření vlastních politik

Hlavním nástrojem pro vytváření vlastních politik SELinux je program audit2allow. Jedná se o perlový skript, který se dá velmi jednoduše spustit z příkazového řádku. Pracuje na principu prohlížení logů, do kterých zapisuje SELinux odmítnuté požadavky.

Audit2allow je možné spustit s několika parametry, které ovlivňují jeho použití. Jako parametr je možné určit vstupní a výstupní soubor. Místo vstupního souboru je možné také využít výstup z programu dmesg. Jako velmi šikovné se ukázalo zvolení možnosti číst z logovacího souboru jen odmítnuté požadavky od posledního načtení bezpečnostních politik. Poslední možností je zapnutí podrobnějšího výstupu, kdy jsou do vygenerovaného souboru zapsány informace, proč který záznam byl vytvořen.
K otestování použitelnosti audit2allow jsem zvolil webový skript phpSysInfo. PhpSysInfo je php skript sloužící k zobrazení informací o systému, na kterém běží.

Protože se jedná o webovou aplikaci, je nutné využít webový server pro fungovaní aplikace. Součástí testovacího systému je webový server apache, pro který jsou již vytvořeny politiky. Protože však phpSysInfo potřebuje pro svoji správnou činnost přístup k dalším souborům, musí se pomocí skriptu audit2allow vytvořit dodatečná nastavení.

Přechod do adresáře se zdrojovými kódy politiky SELinuxu.

 # cd /etc/selinux/targeted/src/policy/

Spuštění programu audit2allow s následujícími atributy:

-d vstup načten z programu dmesg
-v podrobnější vysvětlení výstupu
-l načítat nepovolené operace pouze od posledního načtení politik
-o výstup přidat do specifikovaného souboru

 # audit2allow -d -v -l -o domains/program/phpsysinfo.te

Rekompilace politik SELinuxu a jejich načtení.

 # make reload

Pro uplné vyladění bylo nutné spustit audit2allow několikrát, protože jinak nebyla politka kompletně vytvořena a aplikace PhpSysInfo nepracovala zcela podle předpokladů. Postupně se totiž do logovacího souboru zapisovaly nepovolené operace, které byly postupně spouštěny. Správnou funkci a povolení všech potřebných operací se podařilo vyladit až po šestém spuštění audit2allow.

Výstup vygenerovaný audit2allow

1. spustění

Povolení doméně httpd_sys_script_t načíst informace z programu df pro různé diskové oddíly.

allow httpd_sys_script_t binfmt_misc_fs_t:filesystem getattr;
allow httpd_sys_script_t devpts_t:filesystem getattr;
allow httpd_sys_script_t sysfs_t:filesystem getattr;
allow httpd_sys_script_t tmpfs_t:filesystem getattr;
allow httpd_sys_script_t proc_t:filesystem getattr;

Zpracování výstupu z programu who – informace o přihlášených uživatelích.

allow httpd_sys_script_t initrc_var_run_t:file read;

Přístup k souboru s informacemi o sběrnici.

allow httpd_sys_script_t sysfs_t:dir search;

Přečtení souboru s informacemi o USB rozhraní.

allow httpd_sys_script_t usbfs_t:dir read;

Ostatní konfigurace příkazu df a nastavení přístupu doméně http_t na potřebná místa.

allow httpd_sys_script_t var_lib_nfs_t:dir search;
allow httpd_t proc_t:lnk_file getattr;
allow httpd_t sysctl_t:dir getattr;

2. spuštění

Přístup k souborovému systému typu RPC pipe přes příkaz df.
allow httpd_sys_script_t rpc_pipefs_t:filesystem getattr;

Přečtení souboru s informacemi o sběrnici (při 1. spuštění byl atribut nastaven pouze na vyhledávání a teprve nyní je možné informace načíst).
allow httpd_sys_script_t sysfs_t:dir read;

Přístup k souboru s informacemi o USB rozhraní.
allow httpd_sys_script_t usbfs_t:dir getattr;

Uzamčení pomocí lock
allow httpd_sys_script_t initrc_var_run_t:file lock;
allow httpd_t sysctl_kernel_t:dir getattr;

3. spuštění

U toho a dalších spuštění audit2allow již dochází pouze k doplňování možností přístupu k informacím o sběrnici lspci, proto jsou podrobnější komentáře vynechány.

allow httpd_sys_script_t sysfs_t:dir getattr;

4. spuštění

allow httpd_sys_script_t sysfs_t:lnk_file read;

5. spuštění

allow httpd_sys_script_t sysfs_t:file read;

6. spuštění

allow httpd_sys_script_t sysfs_t:file getattr;

Vyhodnocení

Audit2allow se používá pro rozšíření existujích politik anebo pro generování nastavení nových služeb. Jeho práce je bezproblémová vzhledem k možnostem nastavení vstupů a výstupů souborů. Jako nejvýhodnější řešení je vytváření politik pomocí rekompilace a znovunačtení politik a následné zpracování odmítnutných operací v logovacím souboru. Pro správnou funkčnost výsledné politiky je potřebné provést tento postup několikrát.

4 Popis hotového řešení pro databazi MySQL

Testování probíhalo na počítači s nainstalovaným systémem Centos ve verzi 4. Distribuce Centos je zdarma šířená verze komerční distribuce Red Hat Enterprise Linux. Součástí SELinuxu obsaženého v tomto systému byla pouze definice politik ve verzi Targeted. Nová verze systému Centos 5 již obsahuje i politiky Strict a MLS.

Politika Targeted obsahuje nastavení pro nejznámější démony a pro ostatní služby pracuje s typem nastaveným na unconfined_t, u kterého jsou všechny operace povoleny a zabezpečení tedy opět zajištuje pouze DAC.

Pro vysvětlení existujícího řešení jsem si vybral službu mysqld, což je démon známého databázového serveru MySQL. K výběru tohoto programu mne vedla jeho částečná předchozí znalost a předpoklad fungovaní SELinux politik, protože server při práci spolupracuje jak s diskem, tak zárověn používá síťová zařízení.

4.1 Nastavení portu

V souboru net_contexts je specifikován bezpečnostní kontext pro síťová rozhraní a porty.

Definice portu pro MySQL:

ifdef(`mysqld.te', `portcon tcp 3306 system_u:object_r:mysqld_port_t')

Pokud existuje soubor mysqld.te, je TCP portu 3306 přiřazen bezpečnostní kontext system_u:object_r:mysqld_port_t.

4.2 Nastavení bezpečnostních kontextů

Nastavení kontextů pro soubory se nachází v souboru: file_contexts/program/mysqld.fc.

Programy /usr/sbin/mysqld, /usr/sbin/mysqld-max a /usr/libexec/mysqld zařadí do domény mysqld_exec_t, jedná se o binární soubory, pomocí kterých se spouští démon.

/usr/sbin/mysqld(-max)? -- system_u:object_r:mysqld_exec_t
/usr/libexec/mysqld -- system_u:object_r:mysqld_exec_t

Při spuštění mysql démona se v adresáři /var/run/mysqld/ vytvoří soubor mysqld.pid s id aktuálního procesu, které náleží doméně mysqld_var_run_t.

/var/run/mysqld(/.)? system_u:object_r:mysqld_var_run_t*

Logovací soubor mysql bude označen kontextem system_u:object_r:mysqld_log_t.

/var/log/mysql. -- system_u:object_r:mysqld_log_t*

Domovský adresář mysql, zde se nachází soubory s databázemi a pokud později není specifikováno jinak, budou patřít pod typ mysqld_db_t.

/var/lib/mysql(/.)? system_u:object_r:mysqld_db_t*

Soketový soubor mysql.sock identifikovatelný podle typu "-s" je označen kontextem mysqld_var_run_t.

/var/lib/mysql/mysql\.sock -s system_u:object_r:mysqld_var_run_t

Hlavní konfigurační soubor /etc/my.cnf s kontextem mysqld_etc_t.

/etc/my\.cnf -- system_u:object_r:mysqld_etc_t

Soubory v adresáři /etc/mysql/ budou mít také doménu mysqld_etc_t.

```
/etc/mysql(/.*)?              system_u:object_r:mysqld_etc_t
```

Pouze v případě distribuce Debian

```
ifdef(`distro_debian', `
/etc/mysql/debian-start --      system_u:object_r:bin_t
')
```

4.3 Vynucení obsahu

Soubor s politikou pro MySQL démona je umístěn v domains/program/mysqld.te.

```
#DESC Mysqld - Database server
#
# Author:  Russell Coker <russell@coker.com.au>
# X-Debian-Packages: mysql-server
#
####################################
#
# Rules for the mysqld_t domain.
#
# mysqld_exec_t is the type of the mysqld executable.
#
```

Makro deamon_domain je používané u většiny démonů definovaných v politice targeted, vytváří pro démona hlavní doménu (mysql_t) a slouží k základní definici přístupů ke zdrojům, kterými mohou být soubory s proces ID (PID), lokalizační soubory a informace o zaplnění disku a další.
Z tohoto makra jsou volány další makra:

- var_run_domain - práce s PID
- read_locale - přístup k souboru /etc/localtime

daemon_domain(mysqld, `, nscd_client_domain')

Definuje typ mysqld_port_t a přiřazuje mu atribut port_type, který určuje čísla portů TCP/IP, pomocí definice v souboru net_contexts, kde je pro typ mysqld_port_t definován port 3306.

type mysqld_port_t, port_type;

Spojuje mysql_d s tcp_soketem, který pracuje pod typem mysqld_port_t.

allow mysqld_t mysqld_port_t:tcp_socket name_bind;

Povoluje doméně mysqld_t vytvářet a upravovat soketový soubor /var/lib/mysql/mysql.sock, který má typ mysqld_var_run_t.

allow mysqld_t mysqld_var_run_t:sock_file create_file_perms;

Makro etcdir_domain umožňuje doméně mysqld_t číst adresáře s nastaveným typem etc_t a zároveň je spustěno makro etc_domain, které umožní číst soubory s typem etc_t. Soubory a adresáře s typem etc_t se nacházejí ve složce /etc/, ve které se nachází většina konfiguračních souborů.

etcdir_domain(mysqld)

Definuje alias typu mysqld_etc_t, který je pojmenován etc_mysqld_t. Pravděpodobně kvůli kompatibilitě a zachování syntaxe, ale typ etc_mysqld_t není nikde použit.

typealias mysqld_etc_t alias etc_mysqld_t;

Definice typu mysqld_db_t, kterému jsou přiřazeny atributy file_type a sysadmfile. File_type určuje, že se jedná o soubor nebo adresář a sysadmfile povoluje přístup pouze uživateli s rolí administrátora.

type mysqld_db_t, file_type, sysadmfile;

Makro log_domain umožní doméně mysqld přístup k souborům s typem var_log_t.

log_domain(mysqld)

Makro tmp_domain autorizuje doménu mysql_d pro vytváření a úpravu souborů s typem tmp_t nacházející se v adresáři /tmp.

tmp_domain(mysqld)

Povoluje doméně mysqld_t číst soubor s typem usr_t.

allow mysqld_t usr_t:file { getattr read };

Definuje přístupová práva čtení a zápisu pro rouru v doméně mysqld_t.

allow mysqld_t self:fifo_file { read write };

Umožňuje vytvářet UNIXový soket v doméně mysqld_t.

allow mysqld_t self:unix_stream_socket create_stream_socket_perms;

Umožní procesu běžícím v doméně initrc_t připojení k UNIXovému soketu s typem mysqld_t.

allow initrc_t mysqld_t:unix_stream_socket connectto;

Povolí zápis doméně initrc_t do síťového soketu mající typ mysqld_var_run_t.

allow initrc_t mysqld_var_run_t:sock_file write;

Umožní operace logovacího souboru s typem mysqld_log_t doméně initrc_t.

allow initrc_t mysqld_log_t:file { write append setattr ioctl };

Umožňuje doméně mysqld_t změnu systémových práv.

allow mysqld_t self:capability { dac_override setgid setuid sys_resource net_bind_service };
allow mysqld_t self:process { setsched getsched setrlimit signal_perms rlimitinh };

Proces v doméně mysql_t může číst soubory s typem proc_t. Jedná se o soubory na souborovém systému /proc.

allow mysqld_t proc_t:file { getattr read };

Makro create_dir_file povoluje doméně mysqld_t vytvářet adresáře, soubory a linky s typem mysqld_db_t.

create_dir_file(mysqld_t, mysqld_db_t)

Pokud proces v doméně mysqld_t vytváří soubory (nebo adresáře) pod adresářem majícím typ var_lib_t, pak takto vytvořené soubory a adresáře budou mít nastavený typ mysqld_db_t. Zároveň je těmto souborům nastavena práva pro práci s doménou mysql_d. Jedná se o soubory a adresáře umístěné v /var/lib/mysql, kam si MySQL server ukládá svoje datové tabulky.

file_type_auto_trans(mysqld_t, var_lib_t, mysqld_db_t, { dir file })

Základní makro umožňující doméně mysqld_t pracovat se síťovým rozhraním a TCP/UDP protokoly.

can_network(mysqld_t)

Makro povolující přístup doméně mysqld_t do Network Information Service.

can_ypbind(mysqld_t)

Makro umožní procesu běžícím v doméně initrc_t přístup k souborům, složkám a linkům majícím typ mysqld_etc_t (jedná se především o soubor /etc/my.cnf).

r_dir_file(initrc_t, mysqld_etc_t)

Umožní doméně mysqld_t číst soubory s nastaveným typem etc_t nebo etc_runtime_t (konfigurační soubory v adresáři /etc).

allow mysqld_t { etc_t etc_runtime_t }:{ file lnk_file } { read getattr };

Povoluje procesu v doméně mysqld_t vyhledávat v adresářích s typem etc_t.

allow mysqld_t etc_t:dir search;

Makro umožňující čtení proměných sysctl.

read_sysctl(mysqld_t)

Makro pro vytvoření Unix stream soketu mezi sysadm_t a mysqld_t.

can_unix_connect(sysadm_t, mysqld_t)

Povolení čtení lokálního nastavení MySQL v domovském adresáři roota (soubor /root/my.cnf).

```
allow mysqld_t sysadm_home_dir_t:dir search;
allow mysqld_t sysadm_home_t:file { read getattr };
```

Pokud je definován soubor s konfigurací programu logrotate, bude proveden následující kód. Jedná se o definici přístupů procesu programu logrotate k souborům MySQL démona. V testované distribuci není logrotate.te definován, kompilace zadaného kódu se tedy neprovede.

```
ifdef(`logrotate.te', `
r_dir_file(logrotate_t, mysqld_etc_t)
allow logrotate_t mysqld_db_t:dir search;
allow logrotate_t mysqld_var_run_t:dir search;
allow logrotate_t mysqld_var_run_t:sock_file write;
can_unix_connect(logrotate_t, mysqld_t)
')
```

Umožní uživatelům v doméně usrdomain (user_t a sysadm_t) připojení k databázovému serveru. Kompilace této podmínky se rozvněž během kompilace neprovedla.

```
ifdef(`user_db_connect', `
allow userdomain mysqld_var_run_t:dir search;
allow userdomain mysqld_var_run_t:sock_file write;
')
```

Daemontools je kolekce nástrojů pro správu UNIXových služeb. Při kompilaci je kód vynechán.

```
ifdef(`daemontools.te', `
domain_auto_trans( svc_run_t, mysqld_exec_t, mysqld_t)
```

allow svc_start_t mysqld_t:process signal;
svc_ipc_domain(mysqld_t)
')dnl end ifdef daemontools

V souboru tunables/distro.tun je definována distribuce jako "distro_redhat", takže kód v následující podmínce bude zkompilován.

ifdef(`distro_redhat', `

Procesu běžícím v doméně initrc_t je povoleno pracovat s právy adresářů, které mají typ mysqld_db_t.

allow initrc_t mysqld_db_t:dir create_dir_perms;

Následující makro umožní přístup k soketu mysql démona, který je standartně umístěn v adresáři s type mysqld_db_t.

file_type_auto_trans(mysqld_t, mysqld_db_t, mysqld_var_run_t, sock_file)
')dnl end ifdef distro_redhat

Procesu v doméně mysqld_t umožní vytvářet soket pro komunikaci se síťovými vrstvami.

allow mysqld _t self:netlink_route_socket r_netlink_socket_perms;

4.4 Ukázky makra

4.4.1 Definice makra

V souboru domains/program/mysqld.te je voláno makro etcdir_domain.

etcdir_domain(mysqld)

Parametrem je v tomto případě 'mysqld'.

Toto makro (a mnoho dalších) se nachází v souboru macros/global_macros.te a jeho definice je následucící:

```
define(`etcdir_domain', `
etc_domain($1)
allow $1_t $1_etc_t:dir r_dir_perms;
allow $1_t $1_etc_t:lnk_file { getattr read };
')
```

Symbol $1 je při zpracovaní makroprocesorem nahrazen prvním vstupním parametrem, v tomto případě 'mysqld'. Z makra etcdir_domain jsou spuštěny další dvě makra:

- etc_domain
- r_dir_perms

4.4.2 Výsledek makra

Makro etcdir_domain po zpracování makroprocesorem m4:

```
type mysqld_etc_t, file_type, sysadmfile, usercanread;
allow mysqld_t mysqld_etc_t:file { read getattr lock ioctl };
allow mysqld_t mysqld_etc_t:dir { read getattr lock search ioctl };
allow mysqld_t mysqld_etc_t:lnk_file { getattr read };
```

První a druhý řádek je vygenerován z makra etc_domain, zbylé dva jsou z původně volaného makra etcdir_domain.

Závěr

SELinux je odladěný bezpečnostní systém, který je možné využívat pro zvýšení úrovně zabezpečení celého operačního systému. Disponuje řadou konfiguračních možností a lze jej poměrně jednoduše přizpůsobit.

V současné době jsou již vytvořeny politiky pro většinu serverů, které lze na systému Linux provozovat. S příchodem nové verze RHEL 5 se technologie SELinuxu posunula ještě dále. Mezi novinkami je například nástroj audit2why, který překládá důvody zamítnutí operace do formy, která je pro člověka snáze srozumitelná.

Použitá literatura a informační zdroje

[1] Bill McCarty: SELinux - NSA's Open Source Security Enhanced Linux

O'Reilly, 2004, ISBN 0-596-00716-7

[2] SELinux Quick Start Guide

Revize 1.1

http://www.engardelinux.org/doc/guides/selinux-quick-start-guide/selinux-quick-start-guide.pdf

[3] Faye Coker: Writing SE Linux policy HOWTO

2004

http://www.lurking-grue.org/WritingSELinuxPolicyHOWTO.pdf

[4] Red Hat SELinux Guide

Red Hat, 2005

http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/

Ostatní zdroje:

http://www.nsa.gov/selinux/

http://en.wikipedia.org/wiki/SELinux

http://en.wikipedia.org/wiki/Access_control

http://fedoraproject.org/wiki/SELinux

http://www.redhatmagazine.com/2007/05/04/whats-new-in-selinux-for-red-hat-enterprise-linux-5/

SELinux – manuálové stránky