Security-Enhanced Linux (SELinux)

Hradecký Daniel, A10N0068P

Obsah přednášky

- Co je to SELinux, princip DAC a MAC mechanismů
- Troška historie a aktuální stav vývoje
- Instalace a dostupnost v distribucích
- Bezpečnostní kontext
- Type Enforcement (TE)
- Nasazení a principy konfigurace
- Výhody a nevýhody
- Pár rad na závěr

- Většina současných OS obsahuje slabiny, umožňující potenciálním útočníkům převzít nad nimi kontrolu.
- Každoročně přibývá 90% těchto pokusů

Co je to SELinux

- Bezpečnostní model OS (Linux)
- Implementace povinného řízení přístupu
- OS používají 2 základní mechanizmy řízení přístupu:
  - Volitelný (DAC - Discretionary Access Control)
  - Povinný (MAC - Mandatory Access Control)
- Oba tyto mechanizmy mohou být implementovány zároveň

DAC - Discretionary Access Control

- Princip diskrétního řízení přístupu je založen na ACL (Access Control List).
- Každý objekt má k sobě asociován jeden ACL, kde jsou definována jeho přístupová práva pro uživatele, skupinu a ostatní. Na základě ACL je přístup k objektu povolen či zamítnut.
- jednoduchost, flexibilita
- Každý spuštěný program nebo proces běží pod právy uživatele, který jej spustil. Mají tedy kontrolu nad všemi daty daného uživatele a možnost je měnit.
- Získání kontroly nad procesem SU = katastrofa

MAC - Mandatory Access Control

- Nutnost vytvoření bezpečnostní politiky
- Zavádí se kategorie subjektů (domény) a kategorie objektů (typy)
- Každý objekt a subjekt je zařazen podle jeho práv do příslušné kategorie.
- Pokud se útočníkovi podaří převzít kontrolu nad nějakým programem, může provádět pouze takové operace, které měl daný program povolené, a to i v případě programů pod právy superuživatele.

Troška historie a aktuální stav vývoje

- 90. léta 20 st. : NSA a Secure Computing System pracují na vývoji architektury s povinným řízením přístupu.
- Vytvořen prototyp po jménem Flask
- Později implementováno za spolupráci s Network Associates a MITRE do operačních systémů Linux
- 2000 : zveřejnění jako Open-Source

Troška historie a aktuální stav vývoje

- 2003 : začlenění do kernelu 2.6.0-test3.
- Květen 2004 : SELinux se stal součástí Fedora Core 2 – veliké zklamání (strict policy)
- Listopad 2004 : Fedora Core 3 – (targeted policy)

Instalace a dostupnost v distribucích

- Jak získat SELinux:
  - Nainstalování linuxové distribuce, jejíž součástí již SELinux je.
  - Nainstalováním ze zdrojových nebo binárních balíčků z domovských stránek dané linuxové distribuce.
  - Stáhnutím, zkompilováním a nainstalováním ze zdrojových kódů poskytovaných na stránkách NSA.

Linuxové distribuce se SELinuxem

- Fedora (Red Hat hlavní propagátor SELinuxu, od verze 2, dnes již verze 14)
- CentOS, Scientific Linux
- Debian (mnoho pokusů implementace do starších verzí, plnohodnotný SElinux až v Debian Lenny)
- SUSE Linux (od verze 11.1 spolu s AppArmorem)
- Ubuntu (od verze 8.04 spolu s AppArmorem)

Potřebné balíčky

- **libselinux** : knihovny SELinuxu
- **checkpolicy** : kompilace bezpečnostní politiky
- **selinux-policy-targeted** : politika targeted
- **selinux-policy-strict** : politika strict
- **selinux-policy-mls** : politika mls
- **libsemanage** : graficky management
- **policycoreutils** : zavádění politik do jádra
- **selinux-policy-devel** : komp. vlastních pravidel
- **policycoreutils-newrole** : změna rolí

Bezpečnostní kontext

- Tvar kontextu:

identita:role:typ:mls:mcs

- **identita** : Odlišná od UID, na základě identity se rozhoduje, do kterých rolí lze přejít.
- **role** : Na základě role lze blíže specifikovat přístupy k typům.
- **typ** : Nejdůležitější část bezpečnostního kontextu, na základě typů se rozhodují téměř všechny přístupy .
- Zjištění bezpečnostního kontextu : *id -Z, ls -Z, ps -Z*

Type Enforcement pravidla

- Týká se této části kontextu:

identita:role:typ

- Type Enforcement je srdcem celého SELinuxu a základem každé politiky.
- Každému subjektu i objektu je přiřazen typ.
- Se všemi subjekty (resp. objekty), které jsou ve stejné doméně (resp. mají stejný typ), je zacházeno stejným způsobem.

Type Enforcement pravidla

- Pravidla ukládána do konfiguračních souborů s příponou .te
- Struktura souboru .te:
  - deklarace atributů
  - deklarace typů
  - **pravidla:**
    - přístupová
    - přechodová

Přístupová pravidla TE

- Definuje, jaký typ nebo atribut smí přistupovat k jinému typu a které operace má povoleno provádět.

Tvar: *Příkaz typ_subjektu typ_objektu: třída {operace}*

Příklad: *allow user_t bin_t: file { read execute getattr };*

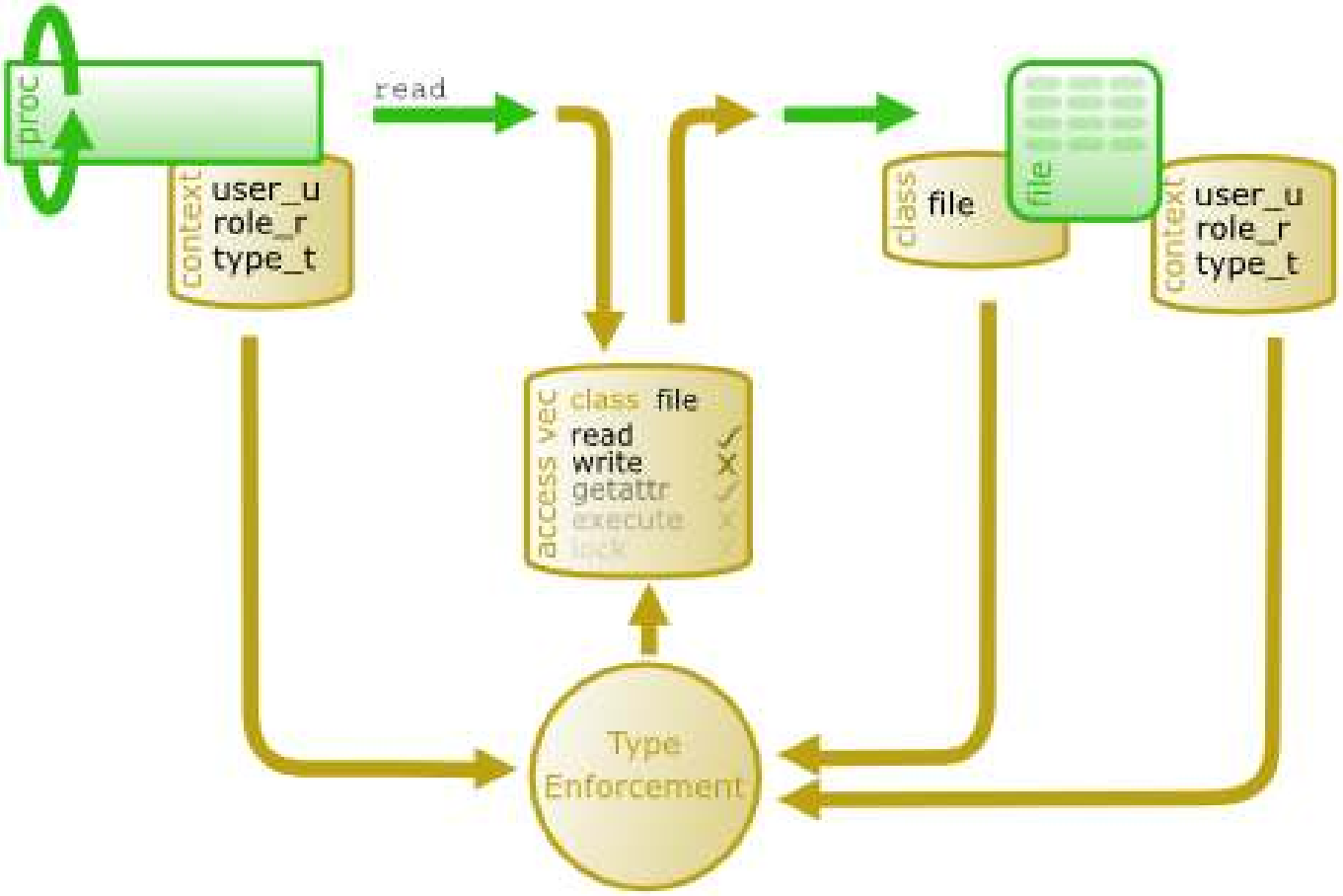
Povolí subjektům typu user_t otevřít a číst soubory typu bin_t

- Příklad dalších příkazů: auditallow, dontaudit
- Opak allow = neverallow : zakazuje dané op.

Přechodová pravidla TE

- Pravidla, umožňující přechod určitého typu (domény) do jiného.
- Postup:
  - deklarace přechodu
  - povolení změny typu (domény)
  - operace entrypoint (vstupní bod přechodu)
- Použití: změna práv daného procesu
- Přechodem do jiné domény proces ztrácí práva předchozí domény a získává práva domény nové.

Type Enforcement - schéma



Nasazení a principy konfigurace

- Vytváření bezpečnostní politiky pomocí konfiguračních souborů
- **3 typy** konfiguračních souborů
  - **.fc* : přepisovací pravidla
  - **.if* : makra
  - **.te* : TE pravidla
- Při vytváření vlastní politiky se pravidla ukládají do */usr/share/selinux/devel/*
- soubor *Makefile* v adresáři devel slouží ke kompilaci našich pravidel

Nasazení a principy konfigurace

- *make -f /usr/share/selinux/devel/Makefile*

  zkompiluje pravidla *name.fc*, *name.if*, *name.te* a vytvoří v adresáři devel modul *name.pp*

- modul *name.pp* ještě není zaveden do jádra

- *semodule -i name.pp :* zavede modul do jádra
- *semodule -r name.pp* : odstraní modul z jádra
- *semodule -l* : vypíše seznam modulů v jádře

Výhody a nevýhody

- Výhody
  - velice silný nástroj pro bezpečnost OS
  - vše je v moci bezpečnostního administrátora
  - jednoduchý princip bezpečnostního kontextu
  - dostupnost v distribucích
  - množství manuálů, publikací a návodů

Výhody a nevýhody

- Nevýhody
  - složitost (oficiální dokumentace 130 stran)
  - občasné problémy s některými aplikacemi
  - časová náročnost vytváření bezpečnostní politiky
  - stálé dolaďování
  - při neopatrných operací hrozí pád systému

Pár rad na závěr

- Podrobně se seznámit s pravidly.
- Udržovat štábní kulturu nejen v kódu, ale i v adresářové struktuře.
- Zálohovat moduly.
- U modulů je dobré uvádět verze pro lepší orientaci.
- Dávat si pozor, co konfiguruji. Neopatrnost může zapříčinit pád OS nebo jiné škody.

Děkuji za pozornost…