Bezpečnostní model MAC v projektech grsec a SELinux

Petr Mošna

MAC (Mandatory Access Control)

- 1 ze 3 nejpoužívanějších mechanismů přístupu k souborům (DAC, RBAC)
- Zavádí pro každý subjekt (proces) práva definující ke kterým objektům (soubory, adresáře, sockety, ...) může přistupovat
- Veškerá práva nastavuje administrátor řízení přístupu(NENÍ *root*), uživatelé nemají žádnou možnost jak je ovlivnit

MAC - pokračování

- Práva = určitá pravidla (politika), které poté systém vynucuje
- Process-based politika
- Nevýhoda klasického MAC – nelze určit konktrétní situace, kdy přístup bude povolen (=> RBAC)

Implementace MAC v grsec

- V prvních verzích patchsetu podpora jen MAC (RBAC později)
- MAC implementován pomocí ACL (*Access Control List*)
- Subjekty (objekty) systému mají přiřazen ACL, obsahující množinu dvojic objektů a pravidel
- Přístup subjektů k přiřazeným objektům pouze za využití jednoho z pravidel z ACL

ACL v grsec

- Lze nastavit i neexistujícím subjektům/ objektům
- Všechny ACL standardně v hlavním ACL souboru (umístěn v etc/grsec/acl)
- Možnost natažení externích souborů s dalšími ACLs (pomocí *include<pathname>*)
- Hlavní ACL soubor po spuštění systému chráněn

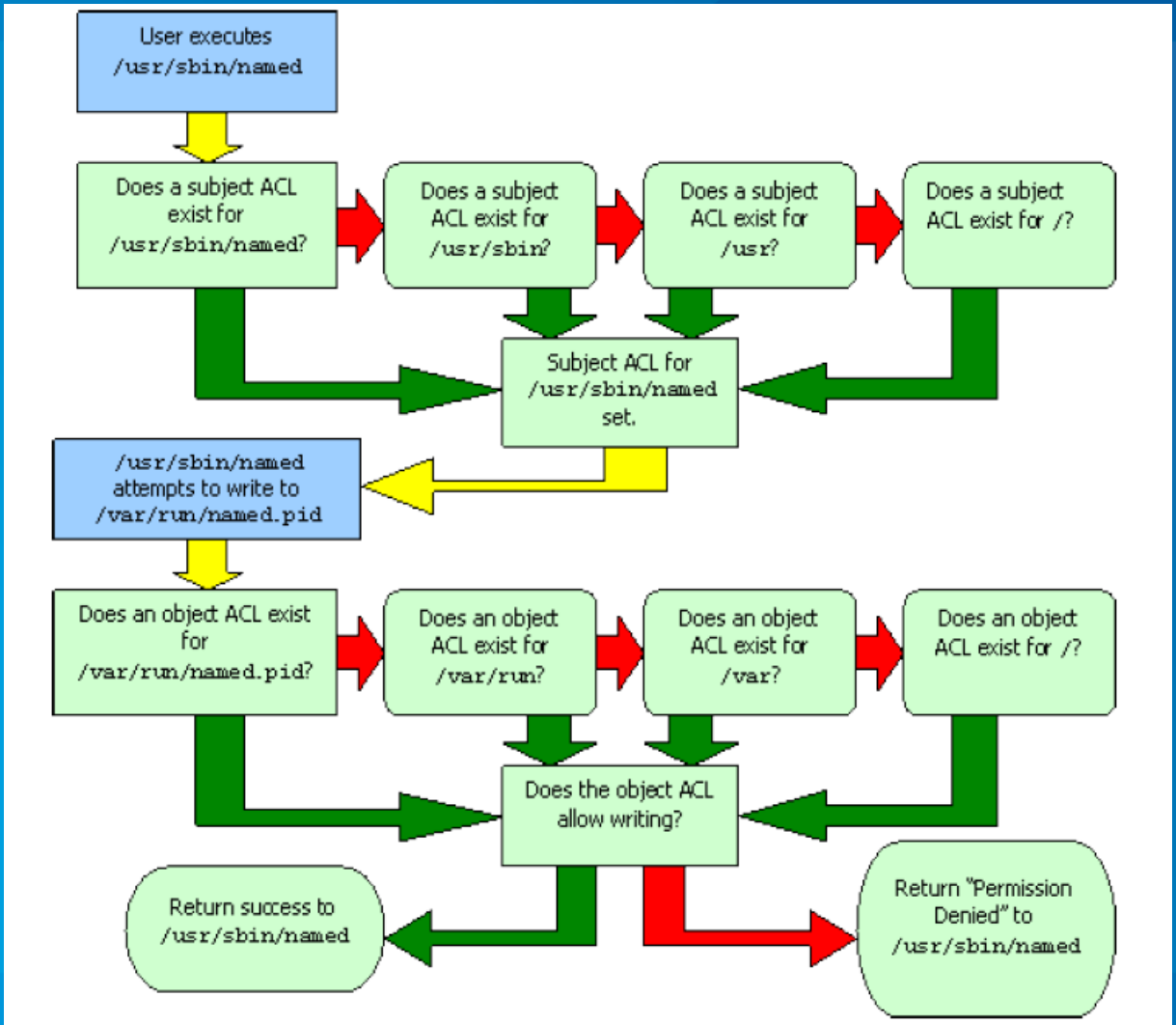
ACL struktura

```
<path of subject process> <optional subject modes> {
        <file object> <optional object modes>
        [+|-]<capability>
        <resource name> <soft limit> <hard limit>
        connect {
            <ip>/<netmask>:<low port>-<high port> <type> <proto>
        }
        bind {
            <ip>/<netmask>:<low port>-<high port> <type> <proto>
        }
}
```

ACL příklad

```
/usr/bin/mailman {
    /tmp rwx
    /proc h
    /home/*/bin x
    /tmp/error.log a
}
```

grsec – mechanismus rozhodování



grsec – další informace

- Musí existovat ACL pro /
- *nested subjects (...:subj2:subj1)*
- Dědičnost ACL subjektů
- Nástroj *gradm* a *learning mode*

grsec - shrnutí

+ Intuitivní ACL, nástroj gradm
- Dokumentace a informace všeobecně, GUI pro tvorbu politiky

Implementace MAC v SELinux

- Sofistikovanější než v grsec
- MAC pomocí implementace FLASK (Flux Advanced Security Kernel) a vlastně i TE (Type Enforcement) = FMAC

Bezpečnostní server

- Obsahuje rozhodovací logiku přístupu
- Rozhoduje podle kontextů subjektů/objektů

Bezpečnostní kontext

- Kontexty udržovány v souboru
- Objekty/Subjekty si nesou pouze (SID)
- Od verze jádra 2.6.x využívany extended attributes (Eas) mimo kernel
- Obecně *<user>:<role>:<type>(:<MLS>)*

Type Enforcement (TE)

- Pracuje s polem <type>
- Definuje k jakým objektům mohou přistupovat objekty, interakci subjektů, ...
- Pravidla jsou psána pro typy a ne přímo subjekty a objekty
- Příklady typů :

etc/shadow – shadow_t

usr/bin/passwd – passwd_t

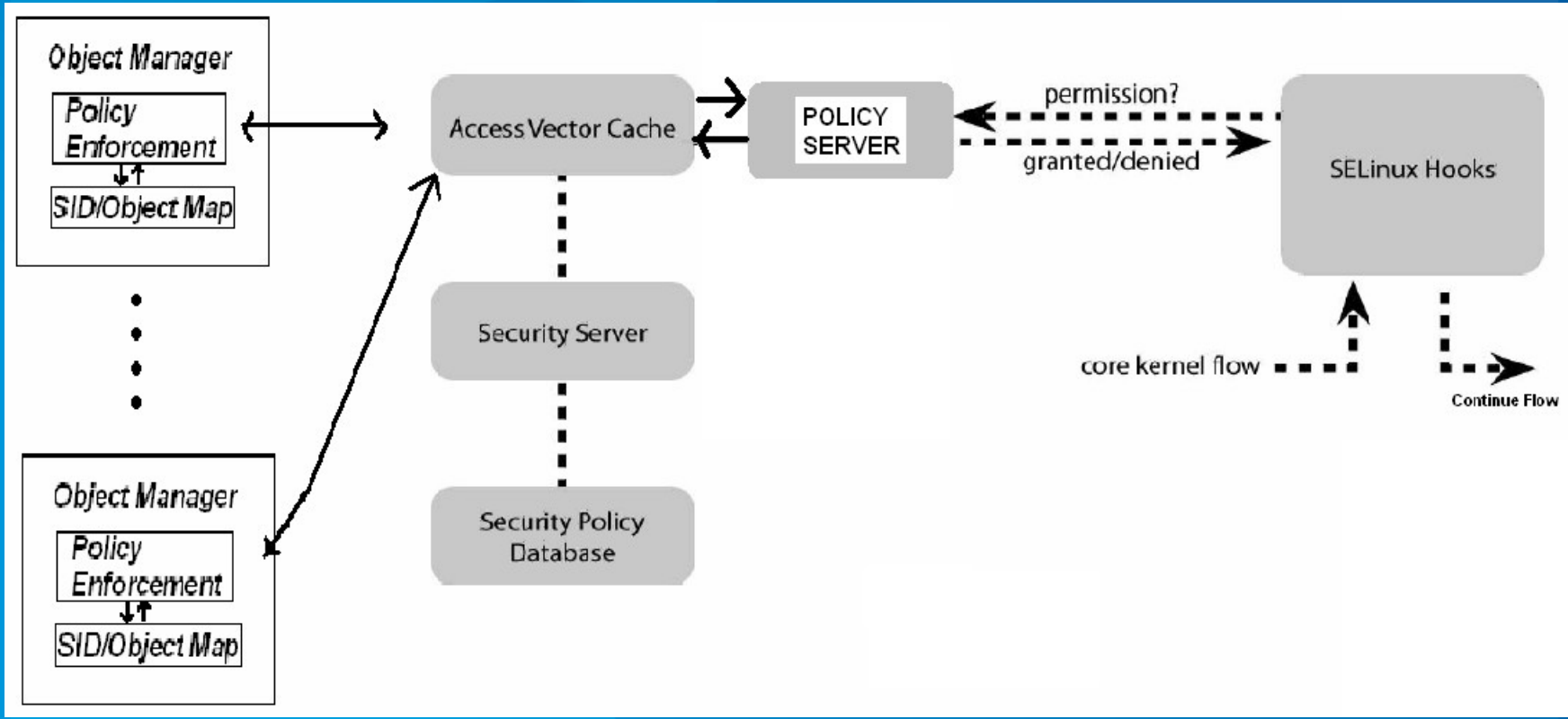
Access Vector Cache (AVC)

- Uchování rozhodnutí o přístupu z bezpečnostního serveru v mezipaměti
- Zlepšuje efektivitu přístupu

Object Manager

- Získává rozhodnutí z bezpečnostního serveru
- Spravuje množinu zdrojů dat a jejich kontexty

Rozhodovací mechanismus v SELinux



Politika v SELinux

- Uložena v *etc/selinux/default/modules/active/base.pp*
- m4, využívání maker (*/usr/share/selinux/devel/includes/**)
- Využití různých přednastavení s kombinací ručního psaní (GUI programy – př. *apol*)

Politika v SELinux - příklad

```
#DESC irssi - IRC client
#

user_application_domain(irssi)
can_network(irssi_t)

# lib access
allow irssi_t lib_t:file { getattr read ioctl };

# allowed signals
allow irssi_t irssi_t:process { signal fork sigchld };

# use of proc filesystem
allow irssi_t proc_t:dir { search };
allow irssi_t proc_t:lnk_file { read };

.....
```

SELinux - shrnutí

+ Velké možnosti nastavení, dokumentace
- Složitá politika, LSM