Themen zur Computersicherheit

Autorisierung
SELinux

PD Dr. Reinhard Bündgen
buendgen@de.ibm.com

FLASK / SELinux

- Entwickelt von der Utah University / US DoD / NSA
- Flask / SELinux: http://www.nsa.gov/research/selinux/docs.shtml
- Unterstützte MAC Formen
  - Type Enforcement (TE) durch Regelwerk (policy) gesteuert
  - RBAC
  - Multi-Level Security (MLS) / Multi-Category Security (MCS) gemäß BLP

SELinux Komponenten

- Subjekte (Prozesse, Teilnehmer)
- Objekte (Prozesse, Dateien, Dateisysteme, Netzwerke, Ports, HW, …)
  - gehört zu einer *Objektklasse*, die die anwendbaren Zugriffstypen definiert (z.B. Klasse file: read, write, append, ...)
- Sicherheitskontext String der Form *user*:*role*:*type*[:*level*]
  - *user*: SELinux user (≠ Unix user), assoziiert mit einer oder mehreren SE Linux Rollen
  - *role*: SELinux Rolle, assoziiert mit einem oder mehreren SELinux Typen
    - nur für Prozesse relevant
    - alle Dateien haben die generische Rolle object_r
  - *type*: SELinux Typ
    - für Subjekt: definiert auf welche Objekte zugegriffen werden kann
    - für Objekt: definiert welche Subjekte auf Objekt zugreifen dürfen
  - *level*: eine Sicherheitsklasse, oder ein Bereich von Sicherheitsklassen (BLP)
    - optional
- Domain: Menge der Prozesse die zu einem SELinux Typ gehören

MAC-Durchsetzung im OS Kern

- Wird überprüft, wenn Zugriffsrecht gemäß Unix DAC gegeben
- Viele Systemrufe des Linux Kern sind mit „hooks“ für Linux Security Module (LSMs) instrumentiert
- SE Linux ist ein solches LSM
- Sicherheitsserver assoziiert Sicherheitskontexte aktiver Kernobjekte mit Sicherheits-Ids (SIDs)

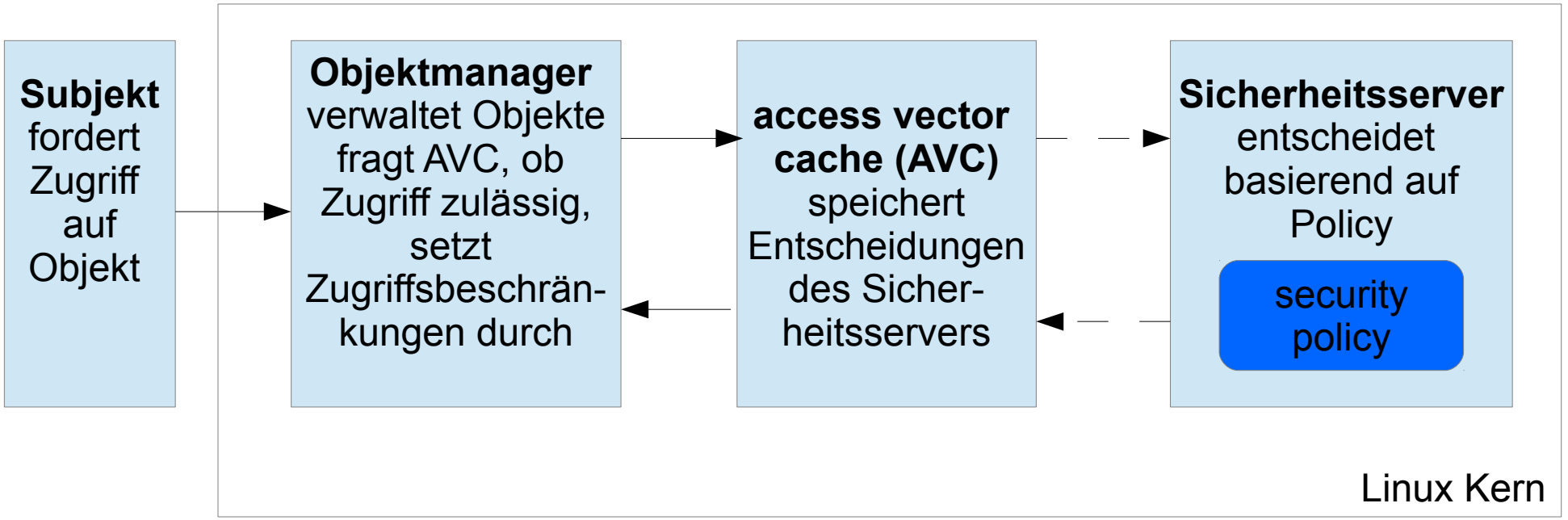


Illustration aus The SELinux Notebook

- Processing a system call: Figure 2.3 (S. 23)

Objektmarkierungen

- jedes Objekt hat eine Sicherheitsmarkierung (security label)
  - Sicherheitskontext bzw SID
- für neues Objekt erfragt Objektmanager eine Sicherheitsmarkierung vom Sicherheitsserver
- Markierungsentscheidung basiert auf
  - Markierung des erzeugenden Subjekts
  - Markierung eines verwendeten Objektes
  - von der Klasse des erzeugten Objektes
  - Beispiel
    - `fd=open(„/home/buendgen/beispiel.txt“, …);`
    - Markierung von `fd` hängt ab von der Markierung des Prozesses, der open aufruft
    - von der Markierung der Datei `/home/buendgen/beispiel.txt`
    - von der Tatsache, dass `fd` zur Objektklasse file descriptor gehört
  - Details der Entscheidung werden durch policies (oder Defaults) beschrieben
  - Sicherheitsmarkierungen von Dateiobjekten werden persistent in Dateisystem gespeichert

Zugriffsentscheidungen

- Zugriffsentscheidung
  - Paar von Sicherheitsmarkierungen
  - Pro Zugriffsart ist eine Zugriffsentscheidung nötig
  - Berechnete Zugriffsentscheidungen werden im AVC gespeichert
- Manche Operation umfassen mehrere Zugriffsarten (möglicherweise zwischen mehreren Objekten)
  - Beispiel
    - die unlink(Datei) Operation benötigt
      1. die unlink Erlaubnis für die Datei und
      2. die remove_name Erlaubnis für das Verzeichnis, in der die Datei liegt

Zugriffsrechte (Auswahl)

- Prozesse
  - transition of security label
  - entrypoint
  - execute
  - inherit open files
  - send signals
  - trace other processes
  - Prozessverwaltung: fork, wait, sched, setpgid, setpriority, …
- Netzwerk
  - Kommunication zwischen sockets
  - send/receive Nachrichten über Netzwerkschnittstellen
  - connectto / acceptfrom
  - port number association
- Dateisysteme/Dateien
  - mount
  - mounton
  - Dienste: statfs, creat, stat, link, rename, unlink, rmdir
  - append
  - write
  - add_name (Verzeichnis)
  - remove_name (Verzeichnis)
  - reparent

Ausgewählte Zugriffsrechte (gemäß FLASK Paper*)

Call	Steuerungsanforderungen			
	Klasse des Arg.	**Erlaubnis**	**Source SID**	**Target SID**
fork	process	fork	current	current
execve	dir	search	current	path
	file	execute	current	file
	process	transition	current	new
	process	entrypoint	new	file
	process	execute	new	file
	process	ptrace	parent	new
	fd	inherit	new	fd
write	fd	setattr	current	fd
	file	write	current	file
	file	append	current	file
connect	socket	connect	current	client socket
	socket	connectto	client socket	server socket
	netif	tcp_send	client socket	netif
	node	tcp_send	client socket	node
	netif	tcp_recv	server socket	netif
	node	tcp_recv	server socket	node

*) Loscocco & Smalley „Integrating Flexible Support for Security Policies into the Linux Operating System“, 2001

Policy-Sprache (Ausschnitt)

Typ Regeln

- Typübergang (type transition) für Prozesse:
  - `type_transition` T1 T2`:` `process` T3
  - Prozess vom Typ T1, der Programmdatei vom Typ T2 ausführt, erhält den Typ T3
  - z.B. `type_transition initrc_t acct_exec_t: process acct_t`
- TE-Zugriffsvektorregeln
  - `allow` T1 T2 `:` C P
  - Subjekt vom Typ T1 hat Erlaubnis P auf Objekte der Klasse C und des Typs T2
  - z.B. `allow initrc_t acct_exec_t: file execute`

Zusicherungen

- TE Zugriffsvektorzusicherungen
  - `neverallow`

Nebenbedingungen (Constraints)

- `constrain` C P E
- Subjekte der Klasse C bekommen die der Erlaubnis P nur wenn die Bedingung E erfüllt ist
- z.B, `constrain process transition (u1==u2 or t1==system_t)`

Illustration aus The SELinux Notebook

- Abschnitt 2.12.1 Domain Transition
  - Seiten 43ff
  - transition: Prozess mit Domain A startet Prozess mit Domain B
  - Figure 2.7
    - entrypoint: Erlaubnis die ausführbare Datei zu sein, die Prozess aus einer Domain startet (Verbindung zwischen Programmdatei-Typ und Prozess-Typ)

User und Rollen in SELinux

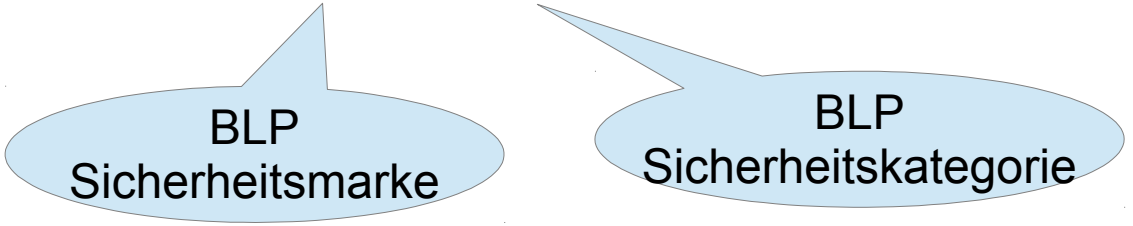
- Rollenmitgliedschaft
  - `user` U `roles` R
  - user U hat Rolle(n) R
  - z.B. `user sysadm_u roles { sysadm_r storageadm_r }`
- Typassoziation einer Rolle
  - `role` R `types {` T1 T2 ... `}`
  - Rolle R umfasst die Typen T1, T2, ...
  - z.B. `role user_r types { mail_t edit_t browse_t }`
- Rollendominanz
  - `dominance { role` R1 `{ role` R2 `}; }`
  - R2 ist Teilrolle von R1
  - z.B. `dominance { role big_r { role small_r}; }`
- Rollenübergang
  - `allow` R1 R2
  - bei Transition kann die Rolle R1 in die Rolle R2 übergehen
  - z.B `allow sysadm_r secadm_r`

Illustration aus The SELinux Notebook

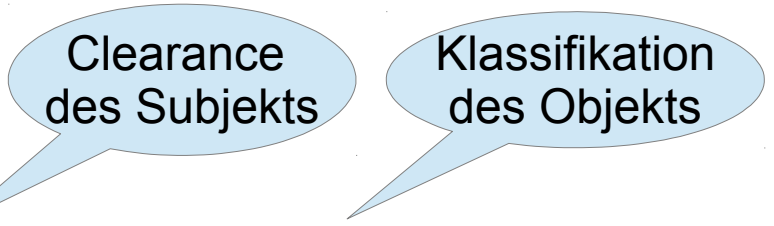
- RBAC
  - Seiten 25, Figure 2.4

Multi Level Security (MLS) / Multi Category Security (MCS)

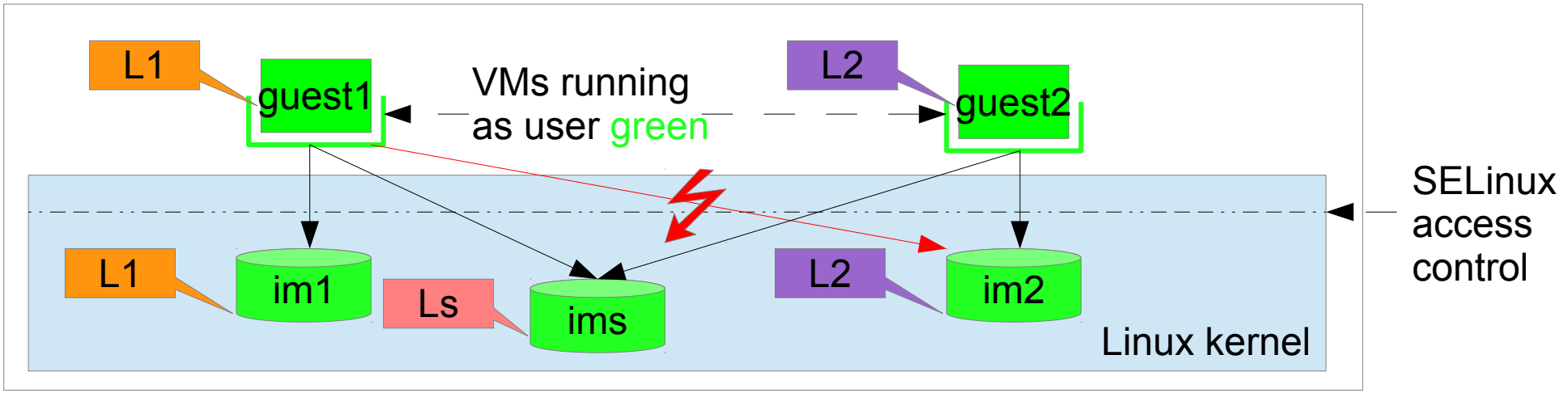
- user:role:type:sensitivity[:category,…] [- sensitivity[:category, …]]
  - sensitivity: BLP Marke
  - level: sensitivity[:category,…] BLP Sicherheitsklasse



- BLP Ordnung über Sicherheitsklassen
  - dominance constraints
    - `mlsconstrain` class access `(`L1 `dom`|`domby` L2`)`
      - `mlsconstrain file read (`l1 `dom` l2`);` # no read up
      - `mlsconstrain file write (`l1 `domby` l2`);` # no write down



Separierung von KVM Gästen



Linux/KVM Hypervisor:

- Gäste (d.h. qemu-Prozesse) laufen unter gleicher user id
- Separation der Gäste mit SELinux durch dynamische erzeugte MLS/MSC Label:
  - guest1 process: system_u,system_r,svirt_tcg_t:s0:c585,c813
  - guest1 image: system_u,system_r,svirt_image_t:s0:c585,c813
  - guest2 process: system_u,system_r,svirt_tcg_t:s0:c535,c601
  - guest2 image: system_u,system_r,svirt_image_t:s0:c535,c601
  - shared image: system_u,system_r,svirt_image_t:s0

Anzeigen der SELinux Konfiguration

SELinux Modi: enforcing, permissive, disabled

Dateien

```
# ls -Z file1

-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

Prozesse

```
# ps -eZ
system_u:system_r:dhcpc_t:s0             1869 ?  00:00:00 dhclient
system_u:system_r:sshd_t:s0-s0:c0.c1023  1882 ?  00:00:00 sshd
system_u:system_r:gpm_t:s0               1964 ?  00:00:00 gpm
```

Zuordnung von Linux Teilnehmern zu SELinux usern

```
# semanage login -l
Login Name                SELinux User              MLS/MCS Range
__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

Illustration aus The SELinux Notebook

- Überblick über SELinux Ökosystem
- Seite 20, Figure 2.2

SELinux Dokumentation

- FLASK Paper
  - Loscocco & Smalley „Integrating Flexible Support for Security Policies into the Linux Operating System“, 2001
  - https://www.nsa.gov/research/_files/publications/security_policies_linux_os.pdf
- The SELinux Notebook
  - frei verfügbares pdf
  - http://freecomputerbooks.com/books/The_SELinux_Notebook-4th_Edition.pdf
- SELinux Wiki
  - http://selinuxproject.org/page/Main_Page
- SELinux in Fedora
  - https://fedoraproject.org/wiki/SELinux

Audit

- protokollieren aller sicherheitsrelevanten Aktionen: audit (log)
- Wichtig für
  - Überprüfung der Einhaltung von Sicherheitsrichtlinien
  - Forensik nach Angriff
- Audit-Logfunktion, darf nicht unerkannt abgestellt werden können
- Audit-Logs dürfen nicht modifizierbar sein
  - Zeitstempel, Sequenznummern
  - senden an ferne Systeme
  - signieren
- Audit-Logs dürfen keine Geheimnisse enthalten