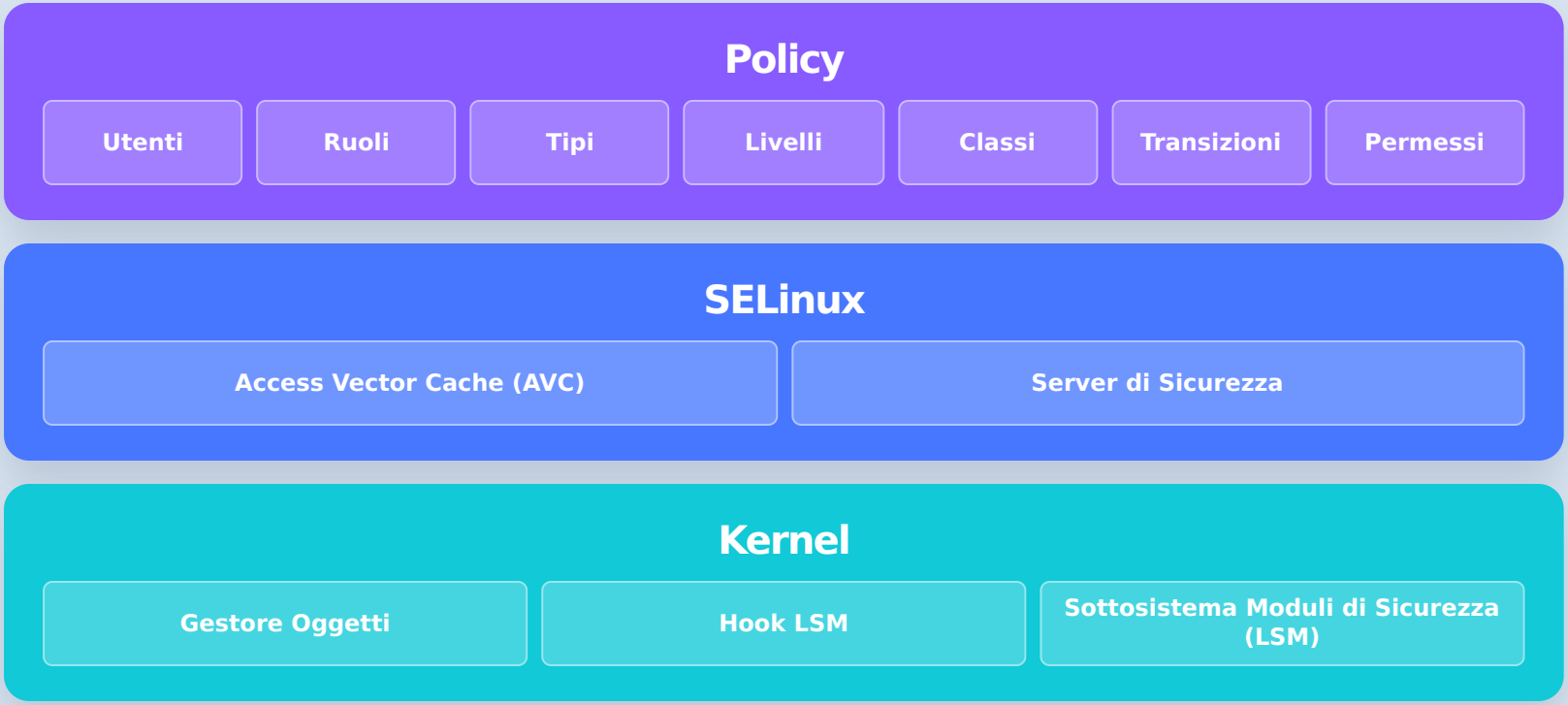
Alla scoperta delle sfaccettature di SELinux

Alessandro Celoria

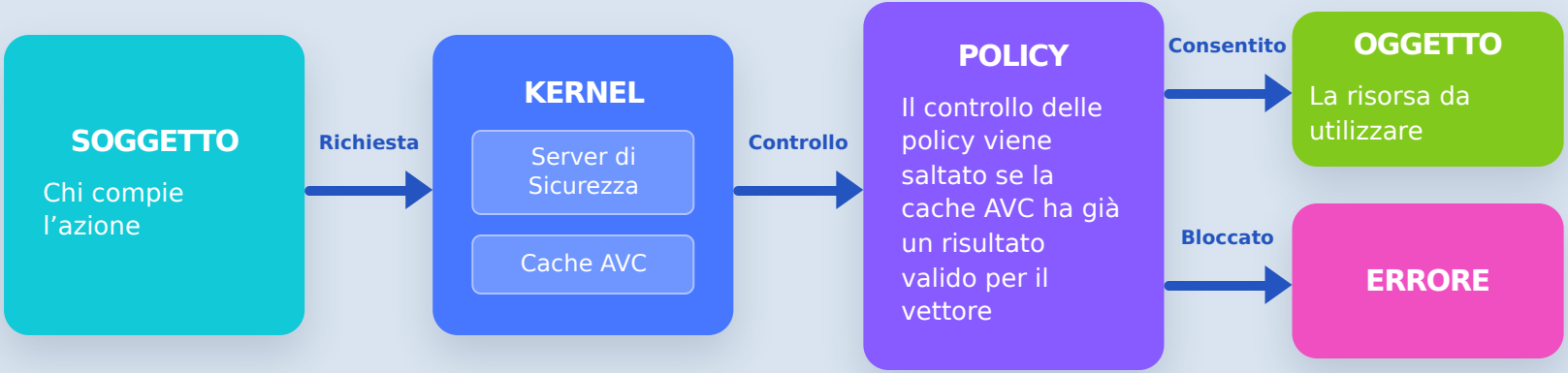
Cos’è SELinux?

We need to go deeper...



Anatomia di un’Azione

Per SELinux, un’azione è una richiesta di permesso: leggere un file, connettersi a un socket, leggere una chiave o caricare una policy. Il controllo avviene in un hook LSM, spesso durante una syscall, ma anche quando il kernel gestisce un pacchetto in arrivo destinato a un socket o quando un suo servizio cerca una chiave.



Anatomia di un’Entità

Ogni oggetto in un sistema con SELinux ha una **label** composto da una gerarchia di attributi. Il **tipo** dell’oggetto e del soggetto determinano quali policy si applicano a determinate azioni.

```
user_u:role_r:type_t:s0-s1:c0.c3
```

Utente

Identità SELinux assegnata al soggetto.

Ruolo

Definisce quali domini può assumere l’utente.

Tipo

Dominio del processo o tipo della risorsa.

Livello

Intervallo di sensibilità MLS, da s0 a s1.

Categoria

Intervallo di categorie MCS/MLS, da c0 a c3.

Anatomia di un Permesso

```
(allow httpd_t httpd_t (dbus (send_msg )))

(deny domain fapolicyd_t (process (ptrace sigkill sigstop )))
```

SOGGETTO Tipo sorgente: chi agisce.

OGGETTO Tipo target e classe dell’oggetto.

AZIONE Permesso ammesso o rimosso.

Anatomia di un Soggetto

defaultuser	`(defaultuser <class> source|target)`	Utente di default da contesto scelto.
defaultrole	`(defaultrole <class> source|target)`	Ruolo di default da contesto scelto.
defaulttype	`(defaulttype <class> source|target)`	Tipo di default da contesto scelto.

Sicurezza Multi-Livello e Multi-Classe

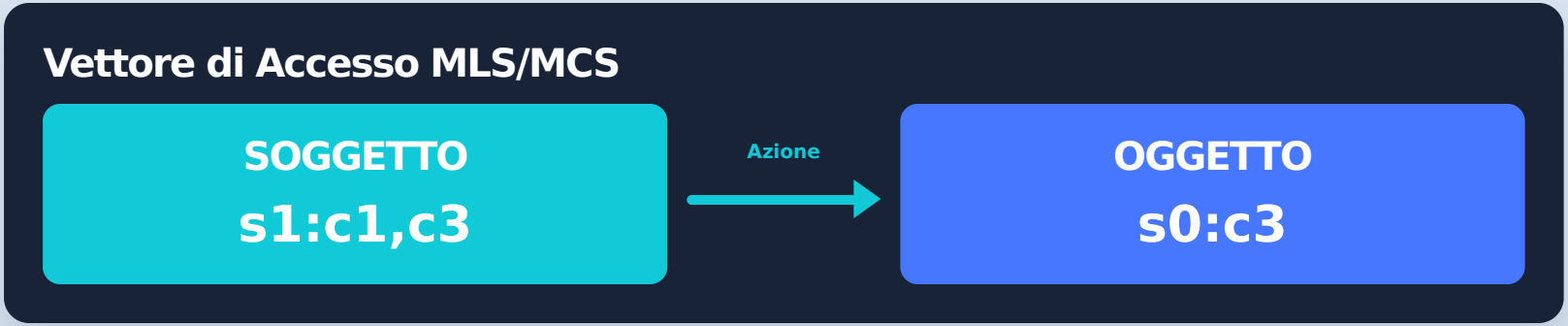
MLS

Una gerarchia di sensibilità

MCS

Compartimenti senza ordine

Sicurezza Multi-Livello e Multi-Classe



MLS

s1 ≥ s0

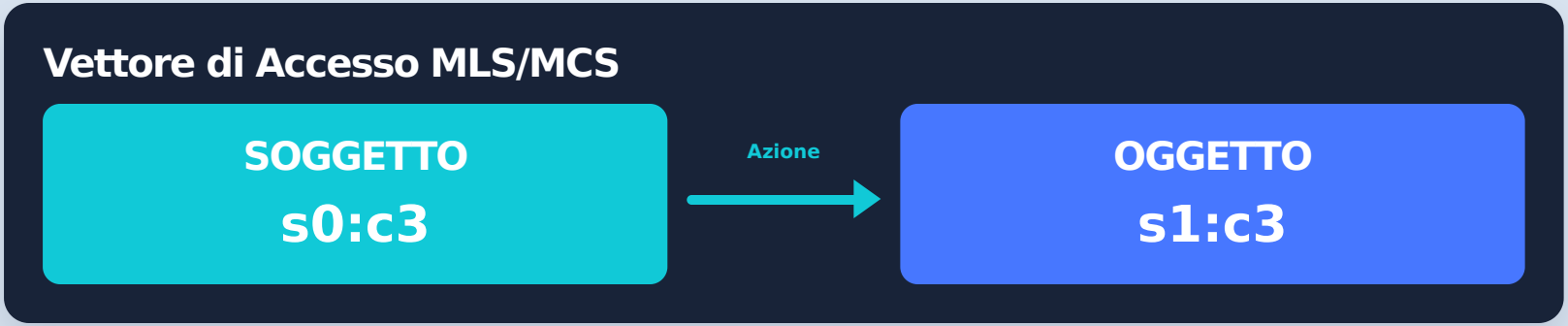
La sensibilità del soggetto domina quella dell’oggetto.

MCS

{c1,c3} ⊇ {c3}

Il soggetto possiede la categoria dell’oggetto.

Sicurezza Multi-Livello e Multi-Classe



MLS

s0 < s1

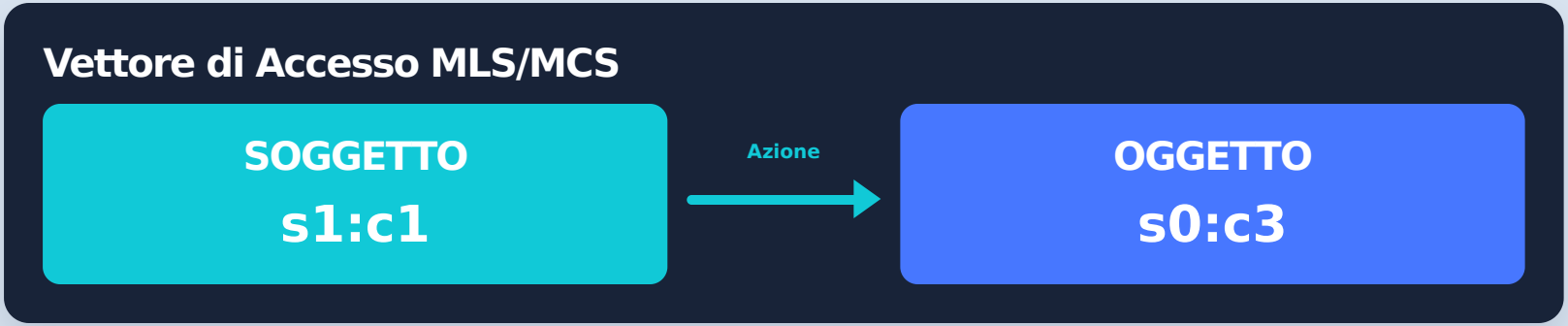
Il soggetto non domina il livello dell’oggetto.

MCS

{c3} ⊇ {c3}

La categoria richiesta è presente.

Sicurezza Multi-Livello e Multi-Classe



MLS

$s1 \geq s0$

La sensibilità del soggetto è sufficiente.

MCS

$\{c1\} \not\supseteq \{c3\}$

Al soggetto manca la categoria c3 dell’oggetto.

Anatomia di un Range

Range MLS `s0-s2`: dal livello di sensibilità `s0` al livello `s2`.

Range MCS `s0:c1.c3`: sensibilità fissa `s0`, categorie da `c1` a `c3`.

Entrambi `s0-s2:c1.c3`: sensibilità da `s0` a `s2` e categorie da `c1` a `c3`.

Per il soggetto, il range delimita i livelli di sensibilità (MLS) e le categorie (MCS) che può assumere. Il controllo degli accessi usa il livello e le categorie correnti: un range più ampio non concede accesso automaticamente, necessitando invece di una transizione valida.

defaultrange `(defaultrange <class> source|target <range>)` Range di default da contesto scelto.

Transizioni di contesto

typetransition `(typetransition <source_type> <target_type|self> <class> [object_name] <new_type>)`

roletransition `(roletransition <current_role> <target_type> <class> <new_role>)`

rangetransition `(rangetransition <source_type> <target_type> <class> <new_range>)`

Contesto di nuovi oggetti

Classe soggetto	Classe oggetto	Classe transizione	Effetto
`process`	`file` (eseguibile)	`process`	Tipo del processo dopo `execve`.
`process`	`dir` (padre)	`file`, `dir`, `lnk_file`, `sock_file`, `fifo_file`, `chr_file`, `blk_file`	Tipo del nuovo oggetto nel filesystem; il nome può selezionarlo.
`process`	`process` (self)	`tcp_socket`, `udp_socket`, `rawip_socket`	Tipo del socket creato.
`process`	`msgq`	`msg`	Tipo del messaggio, quando occorre calcolarlo.

Security ID - Contesti Predefiniti

`sid`	`(sid kernel)`	Dichiara il nome simbolico di uno SID iniziale.
`sidorder`	`(sidorder (kernel security unlabeled))`	Associa i nomi agli slot numerici attesi dal kernel.
`sidcontext`	`(sidcontext kernel kernel_context)`	Assegna un contesto allo SID dichiarato.

Il kernel seleziona uno SID iniziale tramite il proprio numero riservato, non cercando il nome CIL. Un nome come `kernel` può essere rinominato solo mantenendo la posizione e i riferimenti corretti; le policy conservano i nomi convenzionali per compatibilità.

Regole di Labeling

filecon	`(filecon "<path>" <file_type> <context>)`
fsuse	`(fsuse <xattr|task|trans> <fs_name> <context>)`
genfscon	`(genfscon <fs_name> <path> [<file_type>] <context>)`
portcon	`(portcon <tcp|udp|dccp|sctp> <port|(<low> <high>)> <context>)`
nodecon	`(nodecon <subnet> <netmask> <context>)`
netifcon	`(netifcon <interface> <interface_context> <packet_context>)`

Cambio Esplicito del Tipo

`(typechange <source_type> <target_type> <class> <new_type>)`

Calcola il nuovo tipo quando un’applicazione chiede di rietichettare un oggetto esistente.

`(typemember <source_type> <target_type> <class> <member_type>)`

Calcola il tipo di un contesto membro per un’applicazione che gestisce istanze separate.

Entrambe forniscono un tipo al calcolo del contesto: l’applicazione deve poi applicarlo e la policy deve autorizzare l’operazione.

Cambio Esplicito del Contesto

`setfilecon(path, ctx)`	Rietichetta l’oggetto indicato dal percorso.
`fsetfilecon(fd, ctx)`	Rietichetta l’oggetto aperto tramite descrittore.
`setfscreatecon(ctx)`	Imposta il contesto dei nuovi oggetti nel filesystem.
`setsockcreatecon(ctx)`	Imposta il contesto dei nuovi socket.
`setkeycreatecon(ctx)`	Imposta il contesto delle nuove chiavi del kernel.

Vincoli sul Rietichettamento

`(validatetrans <class_id> <expression> ...)`

Permette di cambiare l'etichetta di un oggetto solo quando la condizione indicata è vera.

`(mlsvalidatetrans <class_id> <expression> ...)`

Come `validatetrans`, ma permette di controllare anche i livelli MLS dei contesti.

- `u1:r1:t1` = Contesto precedente. Con MLS, `l1/h1` = livello basso/alto.
- `u2:r2:t2` = Contesto proposto. Con MLS, `l2/h2` = livello basso/alto.
- `u3:r3:t3` = Processo che richiede il cambio.

Esempio

```
; Il tipo dell'oggetto non cambia con il rietichettamento.
(validatetrans file (eq t1 t2))

; Il range MLS resta invariato durante il rietichettamento.
(mlsvalidatetrans file (and (eq l1 l2) (eq h1 h2)))
```

Vincoli sugli Accessi

`(constrain <classpermissionset_id> ... <expression> ...)`

Permette l’accesso ai permessi selezionati solo se la condizione sui contesti è vera.

`(mlsconstrain <classpermissionset_id> ... <expression> ...)`

Come `constrain`, ma la condizione può controllare anche i livelli MLS.

- `u1:r1:t1` = Soggetto. Con MLS, `l1/h1` = livello basso/alto.
- `u2:r2:t2` = Oggetto. Con MLS, `l2/h2` = livello basso/alto.

Esempio

```
; Il rietichettamento è consentito solo tra oggetti dello stesso utente SELinux.
(constrain (file (relabelto)) (eq u1 u2))

; La lettura segue la regola MLS “read down”.
(mlsconstrain (file (read)) (dom l1 l2))
```

Vincoli Padre-Figlio

`(typebounds <parent_type> <child_type>)`

Associa un dominio figlio a un dominio padre. Il kernel limita i permessi effettivi del figlio a quelli disponibili al padre: un `allow` per il figlio non può concedergli capacità che il padre non possiede. Il bound limita i permessi, non concede accessi né provoca una transizione di dominio.

`rolebounds` e `userbounds` sono statement CIL validi e convalidati dal compilatore, ma il kernel li ignora.

Classi Standard di SELinux

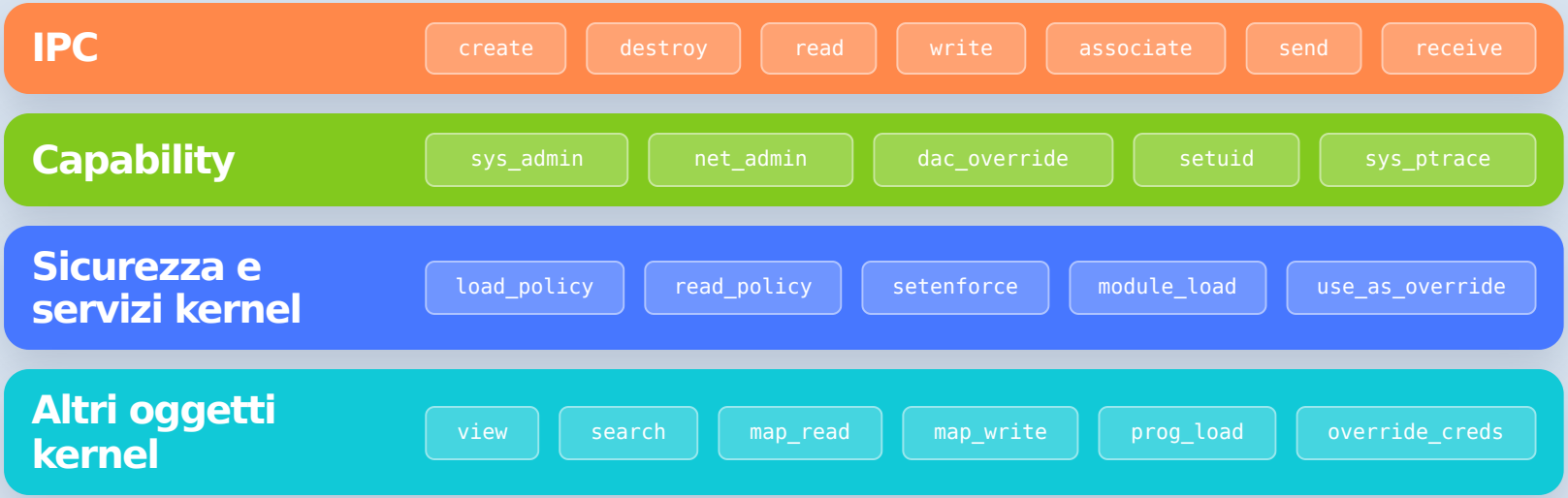
security	process	process2	system	capability	filesystem
file	dir	fd	lnk_file	chr_file	blk_file
sock_file	fifo_file	socket	tcp_socket	udp_socket	rawip_socket
node	netif	nl_socket	packet_socket	key_socket	unix_stream_socket
unix_dgram_socket	sem	msg	msgq	shm	ipc
nl_route_socket	nl_tcpdiag_socket	nl_nflog_socket	nl_xfrm_socket	nl_selinux_socket	nl_iscsi_socket
nl_audit_socket	nl_fib_lookup_socket	nl_connector_socket	nl_netfilter_socket	nl_dnrt_socket	association
nl_kobject_uevent_socket	nl_generic_socket	nl_scsitransport_socket	nl_rdma_socket	nl_crypto_socket	at_socket
packet	key	memprotect	peer	capability2	kernel_service
tun_socket	binder	cap_userns	cap2_userns	sctp_socket	icmp_socket
ax25_socket	ipx_socket	netrom_socket	atmpvc_socket	x25_socket	rose_socket
decnet_socket	atmsvc_socket	rds_socket	irda_socket	pppox_socket	llc_socket
can_socket	tipc_socket	bluetooth_socket	iucv_socket	rxrpc_socket	isdn_socket
phonet_socket	ieee802154_socket	caif_socket	alg_socket	nfc_socket	vsock_socket
kcm_socket	qipcrtr_socket	smc_socket	ib_pkey	ib_endport	bpf
xdp_socket	mctp_socket	perf_event	anon_inode	io_uring	user_namespace
memfd_file					

Classi Standard... Per Categoria

Permessi Comuni per Categoria

Processi e descrittori	`transition` `ptrace` `signal` `use`
Filesystem	`read` `write` `open` `getattr` `search` `create`
Socket	`create` `connect` `bind` `listen` `accept` `sendto` `recvfrom`
Rete	`ingress` `egress` `recvfrom` `sendto` `recv` `polmatch`

Permessi Comuni per Categoria



Opzioni di `/etc/selinux/config`

`SELINUX`	Stato: `enforcing` blocca le violazioni · `permissive` le registra · `disabled` disattiva SELinux
`SELINUXTYPE`	Policy: `targeted` confina servizi selezionati · `mls` abilita MLS

CIL: Classi e Permessi

```
; Definisce permessi comuni e classi, associandoli e ordinandoli.
(common common_id ( permission_id ...))
(class class_id ( permission_id ...))
(classcommon class_id common_id )
(classorder ( class_id ...))

; Dichiara insiemi nominati di permessi associati a una classe.
(classpermission classpermission_id )
(classpermissionset classpermissionset_id ( class_id ( permission_id ...)))
```

CIL: Nomi e Livelli

```
; Dichiara tipi e i loro alias.
(type type_id )
(typealias typealias_id )
(typealiasactual typealias_id type_id )

; Dichiara un alias di sensibilità e lo associa alla sensibilità.
(sensitivityalias alias_id )
(sensitivityaliasactual alias_id sensitivity_id )

; Dichiara un alias di categoria e lo associa alla categoria.
(categoryalias alias_id )
(categoryaliasactual alias_id category_id )
```

CIL: Nomi e Livelli

```
; Dichiara ruoli e associa a ciascuno i tipi ammessi.
(role role_id )
(roletype role_id type_id )

; Dichiara utenti, associa ruoli e intervalli e collega utenti Linux.
(user user_id )
(userrole user_id role_id )
(userlevel user_id level_id )
(userrange user_id levelrange_id )
(userprefix user_id prefix )
(selinuxuser linux_user_name user_id userrange_id )
(selinuxuserdefault user_id userrange_id )
```

CIL: Nomi e Livelli

```
; Dichiara le sensibilità e ne definisce l'ordine.
(sensitivity sensitivity_id )
(sensitivityorder ( sensitivity_id ...))\

; Dichiara categorie, le associa alle sensibilità e definisce livelli e intervalli.
(category category_id )
(categoryorder ( category_id ...))
(sensitivitycategory sensitivity_id categoryset_id )
(level level_id ( sensitivity_id ( category_id ...)))
(levelrange levelrange_id ( low_level_id high_level_id ))
(categoryset categoryset_id ( category_id ...))
```

CIL: Security ID (SID)

```
; Definisce un contesto SELinux con utente, ruolo, tipo e intervallo.
(context context_id ( user_id role_id type_id levelrange_id ))

; Dichiara SID, ne stabilisce l'ordine e associa ciascuno a un contesto.
(sid sid_id )
(sidorder ( sid_id ...))
(sidcontext sid_id context_id )
```

CIL: Attributi

```
; Dichiara attributi di tipo, ne assegna i membri e ne controlla l'espansione.
(typeattribute typeattribute_id )
(typeattributeset typeattribute_id ( type_id ...))
(expandtypeattribute typeattribute_id true|false )

; Dichiara attributi di ruolo e ne elenca i ruoli membri.
(roleattribute roleattribute_id )
(roleattributeset roleattribute_id ( role_id ...))

; Dichiara attributi utente e ne elenca gli utenti membri.
(userattribute userattribute_id )
(userattributeset userattribute_id ( user_id ...))
```

CIL: Booleani e Tunables

```
(boolean boolean_id true|false )
(booleanif boolean_id|expression (true cil_statement ...) (false cil_statement ...))
(tunable tunable_id true|false )
(tunableif tunable_id|expression (true cil_statement ...) (false cil_statement ...))

; Combina booleani in espressioni logiche.
(and expr1 expr2 )
(or expr1 expr2 )
(xor expr1 expr2 )
(eq expr1 expr2 )
(neq expr1 expr2 )
(not expr )
```

CIL: Blocchi, Optional e Macro

```
(block block_id cil_statement ...)

(block block_id (blockabstract block_id ) cil_statement ...)

(block block_id (blockinherit template_id ) cil_statement ...)

(in [before|after] container_id cil_statement ...)

(optional optional_id cil_statement ...)

(macro macro_id (( param_type param_id )...) cil_statement ...)

(call macro_id [ ( param ...) ] )
```

Esempio: Blocchi

```
; Il blocco astratto fa da template per quello successivo.
(block apache_base
  (blockabstract apache_base )
  (type httpd_t )
  (type httpd_config_t )
  (type httpd_log_t )
  (roletype .system_r httpd_t )
  (allow httpd_t httpd_config_t ( dir ( getattr search open )))
  (allow httpd_t httpd_config_t ( file ( getattr open read ))))

(block apache
  (blockinherit apache_base ))

(in apache
  (allow httpd_t httpd_log_t ( file ( open append ))))
```

Esempio: Macro

```
(macro read_files_pattern
  ((type client ) (type directory ) (type filetype ))
  (allow client directory ( dir ( getattr search open )))
  (allow client filetype ( file ( getattr open read ioctl lock ))))

(call read_files_pattern
  ( httpd_t httpd_config_t httpd_config_t ))

(call read_files_pattern
  ( httpd_t httpd_sys_content_t httpd_sys_content_t ))
```

Esempio: Optional

```
(optional apache_kerberos
  (allow httpd_t etc_t ( dir ( getattr search open )))
  (allow httpd_t krb5_keytab_t
    ( dir ( getattr search open read lock ioctl )))
  (allow httpd_t krb5_keytab_t
    ( file ( getattr open read ioctl lock ))))
```

Esempio: Boolean e Tunable

```
(boolean httpd_can_network_connect false)
(booleanif httpd_can_network_connect
  (true
    (allow httpd_t port_type ( tcp_socket ( name_connect )))))

; tunable selezionato in compilazione, salvo preserve-tunables.
(tunable build_httpd_setrlimit false)
(tunableif build_httpd_setrlimit
  (true
    (allow httpd_t self ( process ( setrlimit )))
    (allow httpd_t self ( capability ( sys_resource )))))
```

Esempio: Attributi - Tipi e Regole

```
(typeattribute httpdcontent )
(typeattributeset httpdcontent
  ( httpd_sys_content_t httpd_user_content_t ))
(typeattributeset httpdcontent ( httpd_sys_rw_content_t ))
(allow httpd_sys_script_t httpdcontent ( file ( entrypoint map )))

(typeattribute no_mmap_zero )
(typeattributeset no_mmap_zero
  (and ( domain ) (not ( mmap_low_domain_type ))))
(neverallow no_mmap_zero self ( memprotect ( mmap_zero )))

(expandtypeattribute no_mmap_zero true)
(expandtypeattribute httpdcontent false)
```

Esempio: Attributi - Operare su Insiemi

```
(typeattribute web_scripts )
(typeattributeset web_scripts
  (or ( httpd_sys_script_t ) ( httpd_user_script_t )))

(typeattribute executable_scripts )
(typeattributeset executable_scripts
  (and ( httpd_script_domains ) ( httpd_exec_scripts )))

(typeattribute other_scripts )
(typeattributeset other_scripts
  (xor ( httpd_script_domains ) ( httpd_exec_scripts )))

(typeattribute all_types )
(typeattributeset all_types (all))
```

Esempio: Attributi - Ruoli e Utenti

```
(roleattribute passwd_roles )
(roleattributeset passwd_roles ( system_r ))
(roleattributeset passwd_roles ( user_r staff_r ))
(roletype passwd_roles passwd_t )

(userattribute administrative_users )
(userattributeset administrative_users ( staff_u sysadm_u ))
(userrole administrative_users sysadm_r )
```

Esempio: Alias

```
(typealias phpfpm_t )
(typealiasactual phpfpm_t httpd_t )

(typealias user_chkpwd_t )
(typealiasactual user_chkpwd_t chkpwd_t )

(sensitivityalias public )
(sensitivityaliasactual public s0 )

(categoryalias documents )
(categoryaliasactual documents c0 )
```

Lettura di una Violazione

```
type=AVC msg=audit(1395177286.929:1638): avc: denied { read }
for pid=6591 comm="httpd" name="webpages" dev="0:37" ino=2112
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:nfs_t:s0 tclass=dir
```

L’AVC registra che l’operazione è stata negata.

Lettura di una Violazione

```
type=AVC msg=audit(1720000001.102:241): avc: denied { name_connect }
for pid=2648 comm="httpd" dest=5432
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:postgresql_port_t:s0 tclass=tcp_socket
```

Permesso richiesto dal soggetto su una porta TCP.

Lettura di una Violazione

```
type=AVC msg=audit(1720000002.208:242): avc: denied { write }
for pid=2714 comm="httpd" name="access.log" dev="sda2" ino=4182
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:var_log_t:s0 tclass=file
```

Nome del comando associato al processo che ha fatto la richiesta.

Lettura di una Violazione

```
type=AVC msg=audit(1720000003.314:243): avc: denied { read }
for pid=3045 comm="named" name="private.key" dev="sda2" ino=5220
scontext=system_u:system_r:named_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file
```

Contesto SELinux del soggetto che richiede l’accesso.

Lettura di una Violazione

```
type=AVC msg=audit(1720000004.420:244): avc: denied { getattr }
for pid=3251 comm="httpd" name="backup" dev="sda2" ino=6841
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:user_home_t:s0 tclass=dir
```

Contesto SELinux dell’oggetto a cui si tenta di accedere.

Lettura di una Violazione

```
type=AVC msg=audit(1720000005.526:245): avc: denied { connectto }
for pid=3462 comm="httpd" path="/run/example.sock"
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:example_t:s0 tclass=unix_stream_socket
```

Classe dell’oggetto target: qui un socket Unix di tipo stream.

Grazie per la vostra attenzione!