Istituto Nazionale di Fisica Nucleare

Tutorial days di CCR: Cybersicurezza

selinux disabled?

Enrico M.V. Fasanelli

Prologo

- Questa non è la prima volta che ne parliamo ad un corso di formazione
  - 4-5 Novembre 2015 GGI Arcetri (Antonelli)
  - Aprile e Novembre 2018 nel Corso RH per sistemisti (Brunengo)
- L'utilizzo non è ancora «molto diffuso»
  - Menzionato solo da 2 sedi nel questionario (ma sono sicuro ce ne sia almeno una terza)
- Ancora molto diffuso è il setup

  «selinux disabled»

Perché SELinux?

- La sicurezza di un «normale» sistema Linux (uno con SELinux disabilitato) dipende dal kernel, da tutti i processi privilegiati e da ognuna delle configurazioni di questi.
  - Basta un «errore» in uno qualsiasi dei «sotto-sistemi» a compromettere l'intero sistema
- Un sistema Linux con SELinux abilitato, può essere compromesso solo se il **kernel** o la **configurazione** delle **security-policy** lo permettono.
  - Qualunque compromissione di un sotto-sistema rimane confinata in esso e non può causare la compromissione dell'intero sistema (in assenza di falle nel kernel o nelle security-policy)

Perché SELinux disabled?

- Nostalgia?
  - Dopotutto SELinux è nel kernel di Linux solo dal 2000 (!)
- Perché SELinux è complesso?
  - Il solo SELinux notebook è un manuale da circa 400 pagine (molte delle quali un po' ostiche)
- Perché è poco conosciuto?
  - Solo 3 corsi di formazione INFN nel periodo 2015-2018 e poi niente più
- Perché la security è una rottura?
  - Mai quanto riparare i danni (!)
- Perché manca una figura di riferimento? Un guru-selinux INFN?
  - Io NON sono un guru-selinux e sto provando a usarlo da un po', con una certa fatica per tutti i motivi di cui sopra (!)

Come risolviamo?

- Nostalgia
  - Ognuno si rivolga al proprio psicoterapeuta di fiducia
- Complessità e ridotta conoscenza
  - Ci proviamo in quest'ora
- «Figura» di riferimento
  - Magari da qui esce un gruppo di entusiasti che collaboreranno attivamente con il gruppo security per il supporto ai colleghi (almeno su SELinux)

Zip VS Cut

- Non è «nelle mie corde» tenere in solo 1 ora corsi che altri colleghi hanno tenuto in 3 o più ore e quindi sono costretto a tagliare
- Nozioni essenziali e non esaustive
  - Vi darò una lista di riferimenti bibliografici
- Un paiesempi
  - Sperando che vi invoglino a considerare SELinux enforced

INFN
Istituto Nazionale di Fisica Nucleare

Tutorial days di CCR: Cybersicurezza

selinux disabled?

really?

Enrico M.V. Fasanelli

(con l'ignaro contributo di Antonelli e Brunengo)

DAC security

- DAC: Discretionary Access Control

- sistema di controllo di accesso utilizzato da tutti gli unix
- definisce se e come un processo possa operare su un file
- si basa sui concetti di proprietà e permessi
- in unix: ownership di file e processi, file permission (ACL)
- l'owner può definire le regole di accesso ai propri oggetti indipendentemente dalla volontà dell'amministratore del sistema

MAC security

MAC: Mandatory Access Control

- insieme di regole utilizzate dal sistema operativo per valutare se un **soggetto** (utente o un processo) possa accedere o effettuare operazioni su un **oggetto** (file, directory device, socket, network interface, processo, etc.)
- ai soggetti ed oggetti sono assegnati attributi di sicurezza (**security context**) che servono per verificare se l'operazione è permessa o meno

Un MAC

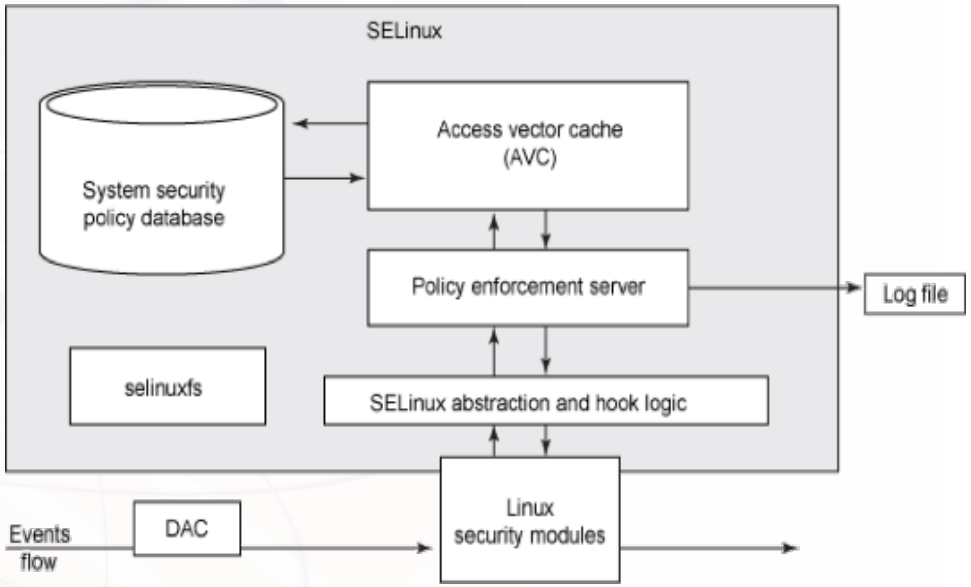
- permette l'isolamento dei processi (limita la privilege escalation)
- permette l'enforcing delle policy di accesso

SELinux

SELinux è l'implementazione di un MAC

Ad ogni tentativo di accesso

- Linux esegue il controllo di accesso standard (DAC)
- se il DAC consente l'operazione, viene interrogato SELinux (via LSM)
- le regole di SELinux sono basate sul **context** di subject e object
  - SELinux non usa informazioni quali user, group, ownership e permission
- Default DENY
- in questo modo può operare un controllo di accesso più fine



le regole di accesso sono definite dall'amministratore, non dall'owner

Access Vector (Cache)

- Il tipo di accesso da controllare è caratterizzato da:
  - chi vuole accedere (un identificativo del **soggetto**)
  - su chi vuole accedere (identificativo **dell'oggetto**)
  - la **classe** della risorsa oggetto dell'accesso (file, file system, device, ...)
  - la **tipologia** di accesso (read, open, kill, ...)
- Questa quaterna è detta Access Vector
- Le regole della policy definiscono, per un Access Vector, se l'operazione è permessa
- Il risultato del check viene messo in una cache (Access Vector Cache)
- La AVC viene utilizzata per migliorare le performance

MAC I – MAC II

- Giusto per correttezza, bisogna dire che SELinux supporta due tipi di MAC
  - Type Enforcement (TE) ovvero targeted policy
    - Il processo è eseguito in un «**dominio**» e l'azione sull'oggetto è controllata dalla **policy**
  - Multi-Level Security (MLS) [e la sua variante Multi-Category Security (MCS)]
    - Usata per mantenere la separazione di applicazioni (come nel caso di VM o di app in Android)

/etc/sysconfig/selinux

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Context (aka label)

- Ad ogni processo ed ogni file viene assegnata una «label»

SELinux user:role:type:level

- Il context gioca un ruolo essenziale nella decisione per la concessione del permesso di accesso

- Un processo può effettuare una operazione su un oggetto, solo se esiste una policy che permette tale operazione sull'oggetto

identificato dalla sua label

SELinux user:role:type:level

- Identità autorizzata dalle policy per specifici ruoli e specifici range di MLS/MCS
- Sono disgiunti dai Linux users, che comunque sono «mappati» su un SELinux user

```
]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
```

SELinux user:**role**:type:level

- SELinux usa il modello di sicurezza di tipo «Role-Based Access Control» (RBAC)
- Il ruolo è un attributo RBAC
- «Strato» intermedio tra ***SELinux user*** e ***domain***
- I ***SELinux users*** sono autorizzati per i ruoli ed i ruoli sono autorizzati per i domini

SELinux user:role:**type**:level

- Il tipo è un attributo del «Type Enforcement»
- Definisce il *dominio* per i processi
- Definisce il *tipo* per i files
- Le policy di SELinux definiscono quando i tipi possono accedere ai tipi
  - Come un *dominio* può accedere ad un *tipo* (un processo può accedere ad un file)
  - Come un *dominio* può accedere ad un altro *dominio* (un processo ad un altro processo)
- Se non esiste policy specifica, l'accesso è negato

SELinux user:role:type:**level**

- Il livello è un attributo del MLS/MCS
- Non viene utilizzato in caso di «targeted»
- Il valore `s0-s0:c0.c1023` significa «qualunque livello»

default context per gli objects

- Quando viene creato un object il suo context è definito da – un valore associato al suo pattern, se definito nella
- configurazione della policy
  - il context della directory in cui l'object viene creato (ereditarietà)
- La configurazione in base al pattern è definita in

/etc/selinux/<policy>/context/files/*

Operazioni sul context I

- Per modificare temporaneamente il context di un object:

  chcon <context> <path>
- Per riportare il context al valore previsto in configurazione:

  restorecon <path>

- per reset globale dei context di tutto il file system al reboot

  touch /.autorelabel && reboot
- operazione necessaria quando si abilita SELinux su un sistema utilizzato in precedenza senza SELinux

Operazioni sul context II

- Per visualizzare context associato al path inserito anche se il path non esiste nel filesystem!

  matchpathcon [-m <type>] <path>

- Per ripristinare il context al tipo previsto dalla configurazione

  restorecon [-R] <path>

- Per visualizzare o modificare la context configuration dei path

  semanage fcontext ...

- è l'utility con cui operi sulla configurazione
- non modifica il context, solo la sua configurazione

file type per undefined context

- Se un file non ha un context definito assume il type: **unlabeled_t**
  - unlabeled_t è usato solo per questo scopo
  - capita quando si abilita SELinux senza fare relabel
- Se un file ha un path che non è incluso nei path configurati nelle policy, assume il type **default_t**
  - default_t è usato solo per questo scopo
- Non ci sono regole nella base policy che permettano a processi confinati un accesso a tali type

SELinux context e mount

- Al mount è possibile definire l'opzione:
  - mount –o context=SELinux_user:role:type:level
  - per alcuni tipi di file system esistono context definiti nella base policy (nfs, iso9660, ...)
- Tutti i file del file system avranno il context specificato dal mount
  - override del context del file (da ext. attr.)
- Tecnica utlizzata per diversi motivi:
  - file system che non supportano ext. attr.
  - file system untrusted (non ci sono policy rule per permettere un accesso ai suoi file da processi confined)
  - file system che si vuole dedicare all'utilizzo per un service (httpd su un NFS file system, ad esempio)

SELinux file context: copy o move

- La copia crea un nuovo file
  - il suo context sarà stabilito dalle regole della creazione del destination file
- Il move tra diversi file system è come il copy
  - crei un nuovo file
- Il move all'interno dello stesso file system lascia il context inalterato
  - non crei un nuovo file
  - il context non è quindi conforme alla configurazione
- In generale cp e mv generano context che non sono necessariamente quello che si desidera

SELinux file context: backup

- tar per default non salva gli extended attributes, ma salva e ripristina il SELinux context se instruito:

  # tar --selinux ...

- Il reset della policy dopo il restore usa il default file context:

  # tar -xvf archive.tar | restorecon -f -

In generale:

- il context e' un extended attribute del file
- esiste un valore definito dalle policy in funzione del path
- in caso di cp/mv o backup/restore si deve sapere cosa si vuole

si deve operare mantenendo la funzionalità (vecchio context) ma in modo che sopravviva ad un relabel (corretta configurazione dei default file context)

context: ereditarietà

- In SELinux vale il principio di ereditarietà del context
- In assenza di una policy rule che affermi qualcosa di diverso:
  - un file eredita il context della directory nella quale viene creato
  - un processo eredita il context del processo che lo ha creato
- Esistono
  - il “root” domain context: kernel_t
  - il “root” type per i file type: root_t

domain transition

- Creazione di un processo in un domain diverso da quello del parent process

- Use case: un utente deve modificare la propria password: non ha accesso al file /etc/shadow.
  - un utente confinato (**domain user_t**) vuole modicifare /etc/shadow (**type shadow_t**) tramite /usr/bin/passwd (**type passwd_exec_t**) per cambiare la propria password
  - l'esecuzione di /usr/bin/passwd deve generare un processo in un dominio che abbia accesso in scirittura al type shadow_t

domain transition (cont.)

- per implementare la domain transition deve essere definita una regola che autorizzi il type passwd_exec_t ad essere “entrypoint” per il dominio passwd_t (che a sua volta ha accesso in scrittura ai file di type shadow_t)

  allow passwd_t passwd_exec_t : file { ... entrypoint ...}

- Accanto a questa regola, devono esistere altre due regole:
  - user_u deve potre eseguire file di tipo passwd_exec_t

    allow user_t passwd_exec_t:file { execute …}
  - user_u deve essere autorizzato a transire verso il domain passwd_t tramite passwd_exec_t

    allow user_t passwd_t:process transition;

Processi e domini I

- I processi figli ereditano il contesto e quindi anche il dominio, dal processo che li ha generati

```
~]$ ps -Z -u enrico
LABEL                             PID TTY          TIME CMD
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 9584 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 9585 pts/1 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 12531 pts/1 00:00:00 ps

unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 11178 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 11179 pts/2 00:00:00 bash
```

Domain traversal

- A meno che non sia consentita una transizione da una apposite regola della policy

```
~]$ ps -aZ -u enrico
LABEL                             PID TTY          TIME CMD
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 9584 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 9585 pts/1 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 12795 pts/1 00:00:00 ps

unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 11178 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 11179 pts/2 00:00:00 bash
unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 12580 pts/2 00:00:00 passwd
```

Policy ad hoc

- Caso d'uso: voglio esportare via rsyslog tutti i log verso un server
- Il file audit.log ha un contesto differente da tutti gli altri file presenti nella directory /var/log (e sotto-directory)

```
~]# ls -Z /var/log/audit/
system_u:object_r:auditd_log_t:s0 audit.log
```

- Le policy di devfault di SELinux non permettono a rsyslog di accedere a file con tale contesto

ausearch

~]# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts today | tail

type=PROCTITLE msg=audit(1665549605.580:366): proctitle=2F7573722F7362696E2F727379736C6F6764002D6E

type=PATH msg=audit(1665549605.580:366): item=0 name="/var/log/audit/audit.log" nametype=UNKNOWN cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0

type=CWD msg=audit(1665549605.580:366): cwd="/»

type=SYSCALL msg=audit(1665549605.580:366): arch=c000003e syscall=6 success=no exit=-13 a0=7f998fbfd110 a1=7f998fbfd1a0 a2=7f998fbfd1a0 a3=0 items=1 ppid=1 pid=1141 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="in:imfile" exe="/usr/sbin/rsyslogd" subj=**system_u:system_r:syslogd_t:**s0 key=(null)

type=AVC msg=audit(1665549605.580:366): avc: denied { search } for pid=1141 comm="in:imfile" name="audit" dev="dm-0" ino=826066 **scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:auditd_log_t:s0** tclass=dir permissive=0

policy per syslog e audit

```
]# sesearch -A | grep "^allow syslogd_t" | grep audit
allow syslogd_t kernel_t:netlink_audit_socket { append bind connect create getattr
getopt ioctl lock nlmsg_read read setattr setopt shutdown write };

allow syslogd_t syslogd_t:capability { audit_control chown dac_override
dac_read_search fsetid ipc_lock net_admin net_bind_service net_raw setgid setuid
sys_admin sys_nice sys_ptrace sys_resource sys_tty_config };

allow syslogd_t syslogd_t:netlink_audit_socket { append bind connect create getattr
getopt ioctl lock nlmsg_read nlmsg_relay nlmsg_write read setattr setopt shutdown
write };
```

Modulo ad hoc

```
]# cat rsyslog_read_audit_logs.te
policy_module(rsyslog_read_audit_logs, 1.0)

gen_require(`
type syslogd_t;
')

logging_read_audit_log(syslogd_t)
```

Installazione del modulo

```
]# make -f /usr/share/selinux/devel/Makefile rsyslog_read_audit_logs.pp
Compiling targeted rsyslog_read_audit_logs module
Creating targeted rsyslog_read_audit_logs.pp policy package
rm tmp/rsyslog_read_audit_logs.mod.fc tmp/rsyslog_read_audit_logs.mod

]# semodule -i rsyslog_read_audit_logs.pp

]# sesearch -A | grep "^allow syslogd_t" | grep audit
allow syslogd_t auditd_log_t:dir { getattr ioctl lock open read search };
allow syslogd_t auditd_log_t:file { getattr ioctl lock open read };
```

Biblio

- Corso di formazione sulla Sicurezza Informatica – Tutorial SELinux
- Corso RedHat per sistemisti INFN – SELinux
- SELinux User's and Administrator's Guide – RHEL7
- SELinux project
- SELinux project (old) wiki
- Using SELinux – RHEL8
- SELinux Security RockyLinux
- stack overflow