Wenn rwx mit ugo zu ungenau ist - SELinux und AppArmor

Secure Linux Admin Conference 2017

Jörg Brühe

Senior Support Engineer, FromDual GmbH

joerg.bruehe@fromdual.com

Über FromDual GmbH

www.fromdual.com

- FromDual bietet neutral und unabhängig:
  - Support für MySQL, Galera Cluster und MariaDB
  - remote-DBA Dienstleistungen
  - Beratung für MySQL, Galera Cluster und MariaDB
  - MySQL und MariaDB Schulungen
- Oracle Silber Partner (OPN)

- Mitglied bei DOAG, SOUG, /ch/open und OSBA

www.fromdual.com

Zur Person

- **Entwicklung verteiltes SQL-DBMS:**
  Unix-Portierung,
  Anschluss Archivierungs-Tools (ADSM, NetWorker)
- **MySQL Build Team:**
  Release-Builds inkl. Tests, Paketierung, Skripte, ...
- **DBA:**
  MySQL für eine Web-Plattform
  (Master-Master-Replikation)
- **Support-Ingenieur (FromDual):**
  Support + Remote-DBA für MySQL / MariaDB / Percona
  mit oder ohne Galera Cluster; Beratung, Schulung

Inhalt

- Zugriffsschutz in Unix
- Warum weitere Mechanismen?
- AppArmor
- SELinux
- Abschluss

• Zugriffsschutz in Unix

- Warum weitere Mechanismen?
- AppArmor
- SELinux
- Abschluss

Warum Zugriffsschutz?

- **Vertraulichkeit:**

  Schutz der einzelnen Benutzer voreinander

- **Zuverlässigkeit:**

  Schutz der System-Komponenten gegen unberechtigte Änderung

”Security“ == ”Informationssicherheit“

Original-Unix in den 70ern (1)

Inode enthält (u.a.)

- Eigentümer und Gruppe
- Neun Rechte-Bits:
  jeweils Lesen (”read“ = r),
  Schreiben (”write“ = w),
  Ausführen (”execute“ = x)
  für Eigentümer (”user“ = u),
  Gruppe (”group“ = g),
  Andere (”other“ = o)
- ...

Original-Unix in den 70ern (2)

- ...
- Ein Bit ”set-user-ID“:
  ”If the tenth bit is on, the system will temporarily change the user identification of the current user to that of the creator of the file whenever the file is executed as a program“

  Beispiele: Abrechnungs-Datei (Spiele?), Programm für ”mkdir()“ (damals privilegiert!)

Bell System Technical Journal, 1978, S. 1910 f.

Original-Unix in den 70ern (3)

- Inode ursprünglich ohne Gruppe (Artikel in cacm, 1974)
- ”Unix Programmer's Manual (Sixth Edition)“ (1973 – 1975) mit ”setuid()“ und ”setgid()“
- ”set-group-ID“ beschrieben in ”On the Security of Unix“, Dennis Ritchie 1975

  Dort auch Unterscheidung zwischen ”real“ und ”effective“ UID und GID

Supplementary Group ID (1)

- Eingeführt mit 4.2BSD (ca. 1983),
- übernommen in System V Release 4 (1990)

  ”Advanced Programming in the Unix Environment“, W. Richard Stevens 1993

- Definiert in ”X/Open Portability Guide“ (1989) sowie IEEE 1003.1-2001 (”POSIX“)
- Vorhanden z.B. in Digital Unix V 4.0 (ca. 1996)

Supplementary Group ID (2)

- Weitere Gruppe = Rechte auf deren Files: `cdrom, lp, lpadmin, ...`
- Weitere Gruppe = Markierung für Rechte:

```
# Members of the admin group
# may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo
# to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Tendenz

- User, Other: Persönlich oder öffentlich
- User, Group, Other:
  Persönlich, Arbeitsgruppe oder öffentlich
- Supplementary Group:
  Benutzer in mehreren Arbeitsgruppen /
  mit mehreren Rollen

Group, Set-group-ID, Supplementary Group:
Zugriffe durch verschiedene Benutzer,
auch wenn Dateien nicht ”öffentlich“

- Zugriffsschutz in Unix
- **Warum weitere Mechanismen?**
- AppArmor
- SELinux
- Abschluss

Grenzen von Permission Bits

- Unterscheidung nur nach Benutzer, nicht nach Anwendung / Programm: Fehlfunktion?

- Begrenzung nur durch die Rechte-Vergabe des Eigentümers:
  keine Einschränkung bei Leichtsinn

  ”DAC“ = ”Discretionary Access Control“,
  ”discretion“ (hier) = ”Ermessen“

Programm ist wichtig!

- Adressbuch:
  Mail-Client ok, Browser nicht
- Öffentliche Files (”`/etc/passwd`“):
  Keine Herausgabe durch Webserver,
  kein Einlesen durch DBMS

=> Programm-spezifische Beschränkungen!

Strengere Mechanismen

- Rechte programm-spezifisch

=> Rechte-Profile für Anwendungen

- Auch wenn Benutzer nicht einschränkt

=> Vorgaben durch Admin

”MAC“ = ”Mandatory Access Control“

Portable Operating System Interface

”POSIX“ = IEEE 1003.1-2001 definiert

- ”File Access Permissions“ (`rwx` für ugo)
- Set User ID, Set Group ID
- ”Supplementary Group ID“
- ”Additional File Access Control Mechanism“: darf gegenüber den Permission-Bits nur einschränken, nicht erweitern

- ”Alternate File Access Control Mechanism“: nicht näher beschrieben

AppArmor und SELinux (1)

- Basis: ”Linux Security Modules“
- Selektiv: Nur ausgewählte Anwendungen werden überwacht

  ”unconfined“ = unbeschränkt
- Distribution liefert Profile/Policies, Admin kann ändern oder weitere erstellen
- Lern-Modus für Profil-Pflege: Alle Verstöße werden zugelassen und protokolliert

AppArmor und SELinux (2)

- Zuerst: Permission-Bits prüfen,
  unzulässige Operationen sind verboten

- Positiv-Listen:
  Was eingetragen ist, ist erlaubt,
  alles Andere ist verboten

- Verstöße kommen ins System-Log,
  Programm erhält Fehler, z.B. EACCESS

- Zugriffsschutz in Unix
- Warum weitere Mechanismen?
- **AppArmor**
- SELinux
- Abschluss

Apparmor: Geschichte

- 1998: Immunix
- 2005: Novell (Kauf) bis 2007
- 2009: Canonical (Wiederaufnahme)
- 2010: im Linux-Kernel 2.6.36 (Oktober)

- Distributionen: openSuSE, Ubuntu

- Basis: Pfadnamen in Textfiles

Allgemeines

- Läuft als Service,
  Kontrolle über ”service“ bzw. ”systemctl“

  Aus: `service apparmor teardown`
  (mit systemctl: ???)

  Start: `service apparmor start`
  oder: `systemctl start apparmor`

- Profil-Modus ”complain“, ”enforce“, ”audit“

```
aa-complain /etc/apparmor.d/PROFIL
aa-enforce  /etc/apparmor.d/PROFIL
aa-audit    /etc/apparmor.d/PROFIL
```

sudo aa-status


```
apparmor module is loaded.
5 profiles are loaded.

5 profiles are in enforce mode.
   /sbin/dhclient
   /usr/lib/NetworkManager/nm-dhcp-client.action
   /usr/lib/NetworkManager/nm-dhcp-helper
   /usr/lib/connman/scripts/dhclient-script
   /usr/sbin/mysqld

0 profiles are in complain mode.
2 processes have profiles defined.

2 processes are in enforce mode.
   /sbin/dhclient (791)
   /usr/sbin/mysqld (784)

0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Reihenfolge

- Profil muss bei Prozess-Start geladen sein, kann sonst nicht benutzt werden

=> Teardown, Start AppArmor
macht laufende Prozesse ”unconfined“

=> Prozess-Start aktiviert geladenes Profil

- Wird der Prozess überwacht?

```
ps -Z
cat /proc/PID/attr/current
```

Testumgebung


```
joerg@kub164:~$ uname -a
Linux kub164 4.4.0-64-generic #85-Ubuntu SMP Mon Feb 20 11:50:30 ...


joerg@kub164:~$ dpkg -l 'apparmor*' 'mysql*' | grep -v '^un'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/...
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                        Version                  Description
+++-===========================-========================-================
ii  apparmor                    2.10.95-0ubuntu2.5       user-space parser
                                                        utility for AppArmor
ii  mysql-client                5.7.17-0ubuntu0.16.04.1 client metapackage ...
ii  mysql-client-5.7            5.7.17-0ubuntu0.16.04.1 client binaries
ii  mysql-client-core-5.7       5.7.17-0ubuntu0.16.04.1 core client binaries
ii  mysql-common                5.7.17-0ubuntu0.16.04.1 common files, ...
ii  mysql-server                5.7.17-0ubuntu0.16.04.1 server metapackage ...
ii  mysql-server-5.7            5.7.17-0ubuntu0.16.04.1 server binaries and
                                                        system database setup
ii  mysql-server-core-5.7       5.7.17-0ubuntu0.16.04.1 server binaries
```

Ausgangs-Zustand

- Datadir = /var/lib/mysql
- Port = 3306
- Bind-Address = 0.0.0.0 # geändert, für Netz

=> Server startet, läuft,
Zugriff lokal und remote möglich

Änderung Datadir: Effekt


```
Datadir = /other_disk/mysql
```

/usr/share/mysql/mysql-systemd-start angepasst

```
joerg@kub164:~$ sudo systemctl start mysql
Job for mysql.service failed because the
control process exited with error code. See
"systemctl status mysql.service" and
"journalctl -xe" for details.
```

- **Beide Befehle zeigen Scheitern, aber wenig hilfreiche Info zur Ursache**

Änderung Datadir: Analyse

Der Kernel loggt Ablehnungen durch AppArmor, u.a. in ”/var/log/syslog“:

```
[ZEIT] audit: type=1400 audit(ZÄHLER):
  apparmor="DENIED" operation="mknod"
  profile="/usr/sbin/mysqld"
  name="/other_disk/mysql/ibtmp1"
  pid=2049 comm="mysqld" requested_mask="d"
  denied_mask="d" fsuid=108 ouid=108

... name="/other_disk/mysql/ibdata1"
  pid=2049 comm="mysqld" requested_mask="wr"
  denied_mask="wr" fsuid=108 ouid=108
```

Zugriff auch über ”dmesg | tail“

Änderung Datadir: Korrektur (1)


```
joerg@kub164:~$ sudo diff -u ALT NEU
--- /home/joerg/ORI-usr.sbin.mysqld  ...
+++ /etc/apparmor.d/usr.sbin.mysqld  ...
@@ -46,8 +46,8 @@
   /usr/share/mysql/** r,

  # Allow data dir access
-   /var/lib/mysql/ r,
-   /var/lib/mysql/** rwk,
+   /other_disk/mysql/ r,
+   /other_disk/mysql/** rwk,

  # Allow data files dir access
    /var/lib/mysql-files/ r,
```

Änderung Datadir: Korrektur (2)

- **/etc/apparmor.d/usr.sbin.mysqld gehört zu ”mysql-server-5.7“ => nicht ändern!**
- **Apparmor sieht lokale Anpassung vor:**

```
joerg@kub164:~$ ls -l /etc/apparmor.d/local/usr.sbin.mysqld
-rw-r--r-- 1 root root 175 Mär  8 17:34
                       /etc/apparmor.d/local/usr.sbin.mysqld

joerg@kub164:~$ dpkg -S /etc/apparmor.d/local/usr.sbin.mysqld
dpkg-query: no path found matching pattern
                         /etc/apparmor.d/local/usr.sbin.mysqld

joerg@kub164:~$ cat /etc/apparmor.d/local/usr.sbin.mysqld
# Site-specific additions and overrides for usr.sbin.mysqld.
# For more details, please see /etc/apparmor.d/local/README.

  /other_disk/mysql/ r,
  /other_disk/mysql/** rwk,
```

Änderung Port: Keine Aktion

- Port = 3307
- In Standard-Profilen nicht eingeschränkt

=> keine Anpassung in AppArmor nötig

Tipps zu AppArmor

- Einzelne Dienste temporär (ab/um)schalten durch Kommando (s.o.)
- Einzelne Dienste permanent abschalten durch Symlink in `/etc/apparmor.d/disable`

`https://wiki.ubuntuusers.de/AppArmor/`

`https://de.opensuse.org/AppArmor`

`http://wiki.apparmor.net/index.php/`

- Zugriffsschutz in Unix
- Warum weitere Mechanismen?
- AppArmor
- **SELinux**
- Abschluss

SELinux: Geschichte

- 1973: Sicherheitsmodell Bell-LaPadula:
  Informationsfluss nur ”aufwärts“
- 1992 - ?: NSA und Partner für (u.a.) Linux:
  ”FLASK“ Sicherheitsarchitektur
- 2004: Enthalten in Fedora Core 3
- 2005: Übernommen in RHEL 4 (Kernel 2.6.9)

- Distributionen: Fedora, RHEL, ...
  Optional / alternativ: openSuSE, Debian, ...
- Basis: Erweiterte File-Attribute

Allgemeines

- Läuft im Kernel, kein sichtbarer Prozess

  Einstellung: `/etc/selinux/config`
  (Änderung: Reboot)

- Modi ”disabled“, ”permissive“, ”enforcing“

```
setenforce 0  # -> permissive
setenforce 1  # -> enforcing
getenforce
```

- Für einzelne Prozesse:

```
semanage -a service_t  # -> permissive
semanage -d service_t  # -> enforcing
```

Status anzeigen

- Globalen Status: `getenforce`
- Für einzelne Prozesse: `semodule -l`

- Prozess-Kontext: `ps -Z`
  Benutzer-Kontext: `id -Z`
  Datei-Kontext: `ls -Z`

- In Minimal-Installation fehlen etliche Management- und Analyse-Tools: optionale Pakete installieren!

Testumgebung


```
[joerg@oel-72 ~]$ uname -a
Linux oel-72 3.8.13-98.7.1.el7uek.x86_64 #2 SMP ...


[joerg@oel-72 ~]$ rpm -qa | egrep 'mysql|selinux|policy'
checkpolicy-2.5-4.el7.x86_64
libselinux-2.5-6.el7.x86_64
libselinux-python-2.5-6.el7.x86_64
libselinux-utils-2.5-6.el7.x86_64
mysql-commercial-client-5.7.16-1.1.el7.x86_64
mysql-commercial-common-5.7.16-1.1.el7.x86_64
mysql-commercial-libs-5.7.16-1.1.el7.x86_64
mysql-commercial-libs-compat-5.7.16-1.1.el7.x86_64
mysql-commercial-server-5.7.16-1.1.el7.x86_64
policycoreutils-2.5-9.0.1.el7.x86_64
policycoreutils-python-2.5-9.0.1.el7.x86_64
selinux-policy-3.13.1-102.0.2.el7_3.4.noarch
selinux-policy-targeted-3.13.1-102.0.2.el7_3.4.noarch
```

Ausgangs-Zustand

- Datadir = /var/lib/mysql
- Port = 3306
- Bind-Address = 0.0.0.0 # geändert, für Netz

=> Server startet, läuft,
Zugriff lokal und remote möglich

Änderung Datadir: Effekt


```
Datadir = /other_disk/subdir


[joerg@oel-72 ~] sudo systemctl start
mysql

Job for mysqld.service failed because the
control process exited with error code.
See "systemctl status mysqld.service" and
"journalctl -xe" for details.
```

- **Beide Befehle zeigen Scheitern,**
  aber wenig hilfreiche Info zur Ursache

Änderung Datadir: Analyse

Der Kernel loggt Ablehnungen durch SELinux, u.a. in ”/var/log/audit/audit.log“:

```
type=AVC msg=audit(ZEIT): avc:
denied  { append } for  pid=2684
comm="mysqld" name="oel72_error.log"
dev="dm-0" ino=67322722
scontext=system_u:system_r:mysqld_t:s0
tcontext=system_u:object_r:default_t:s0
tclass=file
```

”AVC“ = ”Access Vector Cache“:
speichert Entscheidungen für bessere Performance

Konzept (1): Begriffe

- Subjekt = Prozess
- Objekt = File, Directory, Socket, Port, ...

- Kontext = User : Rolle : Typ
- SELinux-User (Kontext) ergibt sich aus Linux-User (/etc/passwd): `semanage login -l`
- User sind für bestimmte Rollen zugelassen, und Rollen für bestimmte Typen

Konzept (2): Mechanismus

- Subjekte dürfen nur auf Objekte zugreifen, wenn ihr Prozess-Typ (”domain“) für den Objekt-Typ zugelassen ist
- ”Type Enforcement is the main permission control used in SELinux targeted policy. For the most part, SELinux users and roles can be ignored.”
  (RedHat SELinux Admin Guide)

- `scontext=system_u:system_r:mysqld_t:s0`
  `tcontext=system_u:object_r:default_t:s0`

Konzept (3): Weiterer Kontext

- Kontext hat weitere Komponenten ”level“ und ”category“ für ”Multi-Level-“ bzw. ”-Category-Security“ (”MLS“ bzw. ”MCS“), in der Praxis irrelevant

  ”MLS on a desktop workstation is unusable.“ (RedHat SELinux Admin Guide)

Konzept (4): Vererbung

- File- und Directory-Kontext wird vererbt, kann mit ”semanage“ geändert werden
- Prozess-Kontext wird vererbt, überwachte Binaries lösen beim ”exec()“ einen Typwechsel aus (”domain“)
- Beispiele:

`/usr/sbin/mysqld`	`mysql_db_t`
`/usr/bin/passwd`	`passwd_t`

- Bestimmte Files und Directories haben den zum Bearbeitungs-Programm passenden Typ

Änderung Datadir: Korrektur (1)

- Für das Datadir den Kontext (intern!) so ändern, dass er richtigen Typ enthält:

```
[joerg@oel-72 ~]$ sudo
     semanage fcontext
         -a -t mysqld_db_t
                "/other_disk/subdir(/.*)?"
```

- Diesen Kontext für das Datadir und rekursiv absteigend ins Filesystem schreiben:

```
[joerg@oel-72 ~]$ sudo
     restorecon -R -v /other_disk/subdir
```

Änderung Datadir: Korrektur (2)

- **Alternative:**
  Den Kontext des Default-Datadir übernehmen und (intern!) auf das neue kopieren:

```
[joerg@oel-72 ~]$ sudo
    semanage fcontext
        -a -e /var/lib/mysql
              /other_disk/subdir
```

- **Diesen Kontext für das Datadir und rekursiv absteigend ins Filesystem schreiben:**

```
[joerg@oel-72 ~]$ sudo
    restorecon -R -v /other_disk/subdir
```

Änderung Port

- Port = 3307

- ```
  semanage port
        -a -t service_port_t
             -p tcp 3307
  ```

- ```
  semanage port -l   # Anzeige
  ```

Erweiterte Attribute

- Achtung bei Backup und Restore!
- Datei-Operationen wirken verschieden:
  mv, cp, tar, star, ...
- Neue Optionen z.B. bei ”cp“:
  `--preserve=context  --context=...`

Defaults bei RedHat

- Mitgelieferte Policy heißt ”targeted“
- ... hat Regeln für fast alle Netzwerk-Dienste (sshd, httpd, ...) und die meisten suid-root-Programme (passwd, ...)
- Umfang: 262.798 ”allow“-Regeln, 156.712 ”dontaudit“(!)-Regeln
- Log-Datei `/var/log/audit/audit.log` oder `/var/log/messages` je nach auditd, rsyslogd und setroubleshootd

Tipps zu SELinux

- **Admin Guide ist sehr hilfreich:**
  `https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/index.html`
- **Weitere Pakete installieren! Besonders**
  `policycoreutils-python # semanage`
- **Ca. 300 ”Booleans“ für schaltbares Verhalten, z.B. ”deny_ptrace“**

`http://fedoraproject.org/wiki/SELinux`

- Zugriffsschutz in Unix
- Warum weitere Mechanismen?
- AppArmor
- SELinux
- **Abschluss**

Persönliche Empfehlungen

- Beide Tools erhöhen die Sicherheit
- Abschalten ist keine Lösung!
- Beide unterstützen lokale Anpassungen und die eigene Regel-Entwicklung
- AppArmor scheint weniger weit verbreitet
- SELinux hat deutlich mehr Default-Regeln
- SELinux wirkt komplexer und verlangt mehr Einarbeitung

Weitere Literatur

Zu beiden:
https://www.heise.de/ct/ausgabe/2015-4-SELinux-und-AppArmor-schuetzen-nach-dem-Einbruch-2521204.html (kostenpflichtig)

Zu AppArmor:
https://blogs.oracle.com/jsmyth/entry/apparmor_and_mysql

Zu SELinux:
https://blogs.oracle.com/jsmyth/entry/selinux_and_mysql
https://www.heise.de/security/artikel/Rollenspiele-270784.html
http://www.admin-magazin.de/Online-Artikel/Mandatory-Access-Control-MAC-mit-SE-Linux
http://www.admin-magazin.de/Das-Heft/2010/02/Probleme-mit-SELinux-finden-und-loesen # Kommandos ... !
https://www.heinlein-support.de/sites/default/files/SELinux-Vortrag.pdf
http://people.redhat.com/tcameron/Summit2014/cameron_w_340_selinux_for_mere_mortals/cameron_w_340_selinux_for_mere_mortals.pdf

Q & A

- Fragen ?
- Kommentare ?

- Folien zum Download:

  `www.fromdual.com/presentations`