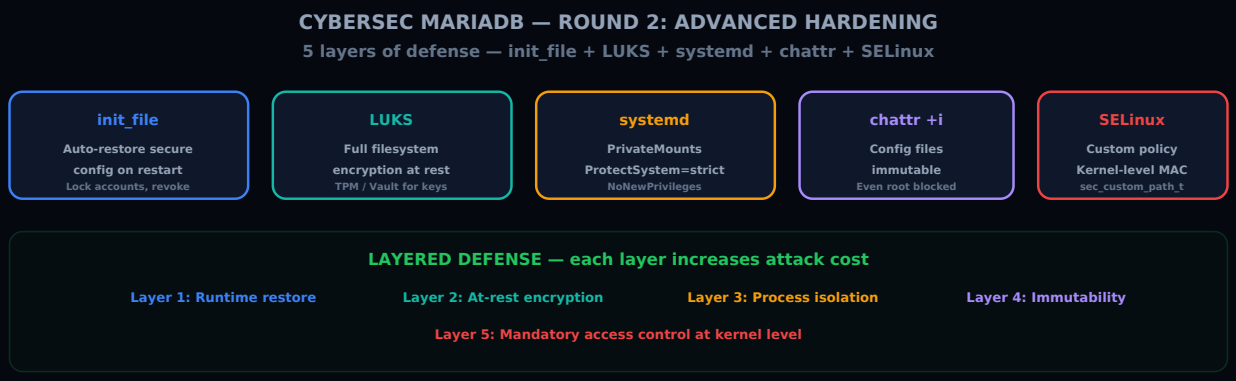
الأمن السيبراني MariaDB: الجولة الثانية

Sylvain ARBAUDIE · 8 يونيو 2025

MARIADB SECURITY HARDENING SYSTEMD SELINUX



ما وراء الأساسيات

تغطي الجولة الأولى من الأمان MariaDB / MySQL الأساسيات: كلمات مرور قوية، والحد الأدنى من المستخدمين، وتمكين TLS، وتكوين جدار الحماية. الجولة الثانية تذهب إلى أبعد من ذلك. نحن ندخل منطقة التشدد المتقدم – وهي تقنيات يطبقها عدد قليل من مسؤولي قواعد البيانات ولكنها تحدث فرقاً ضد مهاجم مصمم.

init_file: النص الصامت

يتيح لك المتغير `init_file` الخاص بـ MariaDB / MySQL تحديد ملف SQL والذي سيتم تنفيذه تلقائياً عند بدء تشغيل الخادم. إنها أداة قوية للتصلب:

```
[mysqld]
init_file = /etc/mysql/conf.d/init_security.sql
```

قد يحتوي الملف `init_security.sql` على:

```
-- Désactiver les comptes par défaut
ALTER USER 'root'@'localhost' ACCOUNT LOCK;

-- Révoquer les privilèges excessifs
```

```
REVOKE ALL PRIVILEGES ON *.* FROM 'app_user'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON app_db.* TO 'app_user'@'%';

-- Supprimer les bases de test
DROP DATABASE IF EXISTS test;

-- Activer l'audit
INSTALL SONAME 'server_audit';
SET GLOBAL server_audit_logging = ON;
```

المميزة: حتى إذا قام أحد المهاجمين بتعديل قاعدة البيانات أثناء عملية التطفل، فإن إعادة تشغيل الخادم تؤدي تلقائيًا إلى استعادة التكوين الآمن.

LUKS: تشفير نظام الملفات

يجب أن تكون بيانات MariaDB غير مشفرة. يدعم InnoDB تشفير مساحة الجدول الأصلية، لكن LUKS (Linux Unified Key Setup) يوفر حماية أكثر شمولًا: فهو يقوم بتشفير نظام الملفات بأكمله، بما في ذلك السجلات والملفات المؤقتة وملفات التكوين.

```
# Créer un volume chiffré LUKS pour le datadir
cryptsetup luksFormat /dev/sdb1
cryptsetup luksOpen /dev/sdb1 mariadb_data
mkfs.ext4 /dev/mapper/mariadb_data
mount /dev/mapper/mariadb_data /var/lib/mysql
```

ال ينبغي أبدًا تخزين مفتاح LUKS على نفس القرص الذي توجد به البيانات. استخدم TPM، أو رمز USB، أو خدمة إدارة المفاتيح الخارجية (Vault، AWS KMS).

systemd: ملف الإعدادات الافتراضية وPrivateMounts

يمكن تقوية ملف MariaDBunit systemd بعدة طرق:

--ملف افتراضي صريح

```
[Service]
ExecStart=/usr/sbin/mariadbd --defaults-file=/etc/mysql/mariadb.cnf
```

يؤدي تحديد `--defaults-file` إلى منع MariaDB من قراءة ملفات التكوين الأخرى (مثل ملف `~/.my.cnf` الضار الذي أسقطه المهاجم).

الجبال الخاصة ومساحات الأسماء

```
[Service]
PrivateMounts=yes
ProtectHome=yes
ProtectSystem=strict
ReadWritePaths=/var/lib/mysql /var/run/mysqld /tmp
NoNewPrivileges=yes
PrivateTmp=yes
```

- **PrivateMounts=yes**: يرى MariaDB مساحة اسم التثبيت الخاصة به. تغييرات التحميل التي تم إجراؤها بواسطة العمليات الأخرى غير مرئية.
- **ProtectHome=yes**: الدليل /home غير قابل للوصول.
- **ProtectSystem=strict**: نظام الملفات للقراءة فقط باستثناء المسارات المسموح بها بشكل صريح.
- **NoNewPrivileges=yes**: لا يمكن لعملية MariaDB الحصول على امتيازات جديدة (لا يوجد معرف).
- **PrivateTmp=yes**: MariaDB له /tmp المعزول الخاص به.

chattr: الثبات

تمنع السمة `+i` (غير القابلة للتغيير) لنظام الملفات ext4 تعديل الملف، حتى عن طريق الجذر:

```
# Rendre le fichier de configuration immutable
chattr +i /etc/mysql/mariadb.cnf
chattr +i /etc/mysql/conf.d/init_security.sql

# Vérifier
lsattr /etc/mysql/mariadb.cnf
# ----i--------e-- /etc/mysql/mariadb.cnf
```

لن يتمكن المهاجم الذي حصل على حق الوصول إلى الجذر من تعديل تكوين MariaDB دون إزالة السمة غير القابلة للتغيير أولاً - مما يترك آثاراً في سجلات التدقيق.

لتعديل الملف بشكل شرعي:

```
chattr -i /etc/mysql/mariadb.cnf
# ... modifier le fichier ...
chattr +i /etc/mysql/mariadb.cnf
systemctl restart mariadb
```

SELinux: السياسات المخصصة

يعد SELinux في وضع الفرض أقوى طبقة أمان مهملة. يأتي MariaDB مزودًا بسياسة SELinux افتراضية، ولكن يمكن للسياسة المخصصة أن تذهب أبعد من ذلك بكثير.

قم بإنشاء نوع SELinux مخصص

```
# Définir un type pour les fichiers de configuration sensibles
semanage fcontext -a -t sec_custom_path_t "/etc/mysql/conf.d(/.*)?"
restorecon -Rv /etc/mysql/conf.d/
```

سياسة الوحدة المخصصة

قم بإنشاء ملف `.te` (فرض النوع) الذي يقيد الوصول إلى MariaDB:

```
# mariadb_custom.te
module mariadb_custom 1.0;

require {
    type mysqld_t;
    type sec_custom_path_t;
    class file { read open getattr };
}

# MariaDB peut lire les configs mais pas les modifier
allow mysqld_t sec_custom_path_t:file { read open getattr };
# Pas d'écriture autorisée sur les configs
```

تجميع وتثبيت:

```
checkmodule -M -m -o mariadb_custom.mod mariadb_custom.te
semodule_package -o mariadb_custom.pp -m mariadb_custom.mod
semodule -i mariadb_custom.pp
```

باستخدام هذه السياسة، حتى إذا قام أحد المهاجمين باختراق عملية MariaDB، فلن يتمكن من تعديل ملفات التكوين - حيث يقوم SELinux بحظر الوصول على مستوى النواة.

دفاع متعدد الطبقات

كل تقنية معروضة هنا هي طبقة من الدفاع. لا شيء يكفي وحده. معًا، يشكلون درعًا مهمًا:

طبقة	الحماية	ضد
init_file	استعادة تلقائية	تغييرات تكوين وقت التشغيل
لوكس	التشفير في حالة الراحة	سرقة القرص الفعلي
مساحات الأسماء systemd	عزل العملية	تصعيد الامتياز
الدردشة +i	ثبات التكوينات	التعديل عن طريق الجذر المخترق
سي لينكس	التحكم الإلزامي في الوصول	استغلال عملية MariaDB

الخلاصة

تتطلب عملية التصلب المتقدمة MariaDB وقتًا وخبرة. كل طبقة مضافة تجعل الهجوم أصعب وأبطأ وأكثر قابلية للاكتشاف.

الأمن ليس حالة ثنائية، بل هو طيف. الهدف ليس أن تكون محصنًا (وهذا مستحيل)، ولكن جعل الهجوم مكلفًا بدرجة كافية بحيث ينتقل المهاجم إلى هدف أسهل.

تم نشر هذه المقالة في الأصل على متوسط.