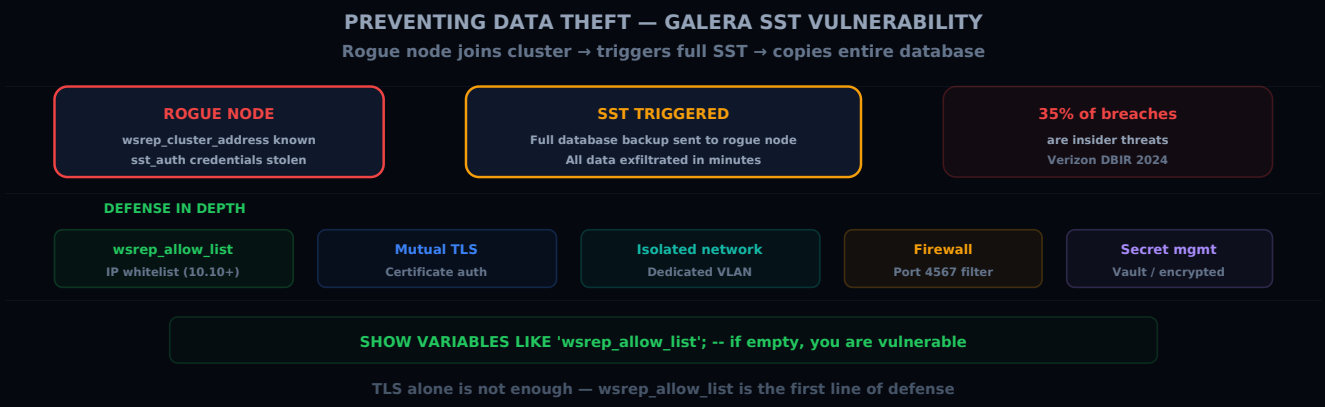
Prévenir le vol de données : édition Galera

Sylvain ARBAUDIE · 12 mars 2025

GALERA MARIADB SECURITY SST



Le scénario cauchemar

Imaginez : un attaquant configure un serveur MariaDB avec les bons paramètres wsrep, connaît l'adresse du cluster Galera et le mot de passe SST. Il rejoint le cluster. Galera détecte un nouveau nœud sans données et déclenche un **State Snapshot Transfer (SST)** — un transfert complet de toutes les données du cluster vers le nœud attaquant.

En quelques minutes (ou heures selon la taille de la base), l'attaquant possède une copie intégrale de votre base de données. Pas d'injection SQL, pas d'exploitation de vulnérabilité applicative. Juste un JOIN au cluster avec les bonnes credentials.

Ce n'est pas de la science-fiction. Selon le rapport Verizon 2024 sur les fuites de données, **35% des violations de données impliquent des menaces internes** — des employés, des sous-traitants, ou des personnes ayant accès légitime à l'infrastructure.

Comment fonctionne le SST

Le State Snapshot Transfer est le mécanisme par lequel Galera initialise un nouveau nœud. Quand un nœud rejoint le cluster sans données (ou avec des données trop anciennes pour un IST incrémental), le cluster déclenche un SST :

1. Le nœud donneur (un membre existant du cluster) est sélectionné

2. Le donneur effectue un backup complet (via mariabackup, rsync ou mysqldump)
3. Le backup est envoyé au nœud joignant via le réseau
4. Le nœud joignant restaure le backup et rejoint le cluster

Le problème : **par défaut, n'importe quel nœud avec les bonnes informations de cluster peut déclencher un SST**. Il n'y a pas de liste blanche, pas de vérification d'identité du nœud joignant.

La configuration minimale pour une attaque

Ce dont un attaquant a besoin :

```
[mysqld]
wsrep_cluster_address = gcomm://10.0.1.10,10.0.1.11,10.0.1.12
wsrep_sst_method = mariabackup
wsrep_sst_auth = sst_user:sst_password
```

Trois informations : l'adresse du cluster, la méthode SST, et les credentials SST. Dans beaucoup d'organisations, ces informations sont stockées dans des fichiers de configuration non chiffrés, des playbooks Ansible en clair, ou des dépôts Git privés.

Pourquoi TLS ne suffit pas

"Mais nous utilisons TLS pour le trafic Galera !" — c'est une objection fréquente. Et elle est insuffisante.

TLS chiffre le trafic entre les nœuds, mais il ne vérifie pas nécessairement l'identité du nœud joignant. Même avec TLS, si l'attaquant possède un certificat signé par la même CA (ce qui est souvent le cas dans les déploiements internes avec une PKI d'entreprise), il peut rejoindre le cluster.

De plus, beaucoup de déploiements Galera n'utilisent pas la vérification mutuelle des certificats (mutual TLS). Ils activent TLS pour le chiffrement mais pas pour l'authentification.

La solution : wsrep_allow_list

Depuis MariaDB 10.10, la variable `wsrep_allow_list` offre un mécanisme de liste blanche IP pour les nœuds autorisés à rejoindre le cluster :

```
[mysqld]
wsrep_allow_list = 10.0.1.10,10.0.1.11,10.0.1.12
```

Seuls les nœuds dont l'adresse IP figure dans la liste peuvent rejoindre le cluster. Un nœud avec une IP non listée sera rejeté, même s'il possède les bonnes credentials SST et les bons certificats TLS.

C'est simple, efficace, et c'est la première ligne de défense que tout cluster Galera devrait avoir.

Défense en profondeur

La sécurité d'un cluster Galera ne repose pas sur un seul mécanisme. Voici une approche de défense en profondeur :

1. wsrep_allow_list — Filtrage réseau

```
wsrep_allow_list = 10.0.1.10,10.0.1.11,10.0.1.12
```

Restreindre les IPs autorisées à rejoindre le cluster.

2. TLS mutuel — Authentification des nœuds

```
wsrep_provider_options = "socket.ssl=yes;socket.ssl_key=/etc/mysql/ssl/server-
key.pem;socket.ssl_cert=/etc/mysql/ssl/server-cert.pem;socket.ssl_ca=/etc/mysql/ssl/ca.pem"
```

Chaque nœud doit présenter un certificat signé par la CA du cluster. Pas de certificat valide = pas de connexion.

3. Réseau isolé — Segmentation

Le trafic Galera (ports 4567, 4568, 4444) devrait circuler sur un réseau dédié, isolé du réseau applicatif et du réseau de management. Un VLAN dédié ou un réseau overlay (WireGuard, IPsec) est recommandé.

4. Pare-feu — Filtrage des ports

```
# iptables : n'autoriser que les IPs du cluster sur les ports Galera
iptables -A INPUT -p tcp -s 10.0.1.10 --dport 4567 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.1.11 --dport 4567 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.1.12 --dport 4567 -j ACCEPT
iptables -A INPUT -p tcp --dport 4567 -j DROP
```

5. Credentials SST chiffrées

Ne stockez jamais les mots de passe SST en clair dans les fichiers de configuration. Utilisez des secrets managers (Vault, AWS Secrets Manager) ou au minimum le chiffrement de fichiers de configuration.

Auditer votre cluster

Vérifiez dès maintenant l'état de sécurité de votre cluster Galera :

```
-- Vérifier si wsrep_allow_list est configuré
SHOW VARIABLES LIKE 'wsrep_allow_list';

-- Vérifier l'état TLS de Galera
SHOW STATUS LIKE 'wsrep_connected';
SHOW VARIABLES LIKE 'wsrep_provider_options';

-- Lister les nœuds actuels du cluster
SELECT * FROM information_schema.WSREP_MEMBERSHIP;
```

Si `wsrep_allow_list` est vide, votre cluster est vulnérable. Configurez-le immédiatement.

Conclusion

La vulnérabilité SST de Galera est un vecteur d'attaque sous-estimé. Un nœud non autorisé peut obtenir une copie complète de votre base de données simplement en rejoignant le cluster. La solution est simple : `wsrep_allow_list` + TLS mutuel + réseau isolé + pare-feu.

35% des fuites de données sont des menaces internes. Votre cluster Galera est-il protégé ?

Cet article a été initialement publié sur Medium.