Processus de sécurité à mettre en œuvre avec Jitsi-Meet.

Jitsi Meet propose un chiffrement de bout en bout (E2EE), certaines caractéristiques spécifiques nécessitent une attention particulière pour une surveillance efficace en matière de sécurité.

Les points essentiels à surveiller dans le cadre d'une supervision sécurité spécifique à Jitsi Meet **E2EE** :

1. Surveillance des certificats TLS/SSL

Le chiffrement est de bout en bout et sécurise le contenu échangé**. Il est essentiel de surveiller l'état des certificats :**

- **Expiration du certificat**
- **Validité du certificat (autorité reconnue, révocations)**
- **Configuration TLS adéquate (protocole minimum TLS 1.2)**

2. Surveillance du serveur Prosody (XMPP)

Prosody est le serveur responsable de la gestion des sessions et des authentifications :

- Tentatives échouées ou multiples d'authentification (brute-force)
- Anomalies de connexion inhabituelles
- État des modules spécifiques E2EE (mod_token_verification, mod_auth_jwt, etc.)

3. Surveillance des accès réseau et flux entrants/sortants

Même si les échanges vidéo/audio sont chiffrés, il est pertinent de surveiller :

- **Pics inhabituels du trafic entrant/sortant**
- **Connexions inattendues ou non autorisées (adresses IP suspectes)**
- **Activité inhabituelle des services TURN/STUN**

4. Vérification de la configuration E2EE

- **Validation régulière** de la configuration **E2EE côté client/serveur.**
- Contrôle que **le mode E2EE** est effectivement **activé** (certains participants peuvent accidentellement ou intentionnellement désactiver le chiffrement E2EE).

5. Surveillance comportementale (Anomalie d'usage)

- Surveiller les pics soudains de consommation CPU/mémoire sur le serveur Jitsi (possible signe de tentative de déni de service ou d'exploitation logicielle).
- Surveillance des modifications des fichiers de configuration critiques (**jicofo.conf, prosody.cfg.lua, sip-communicator.properties**).

6. Logs critiques à surveiller impérativement

- **Prosody :**
  - `/var/log/prosody/prosody.log`
  - `/var/log/prosody/prosody.err`
- **Jicofo :**
  - `/var/log/jitsi/jicofo.log`
- **Jitsi-Videobridge** :
  - `/var/log/jitsi/jvb.log`

7. Surveillance IDS (Intrusion Detection System)

- Activation d'un IDS comme **Suricata** ou **Wazuh** pour détecter des tentatives d'exploitation via signatures spécifiques (attaques SIP, XMPP, HTTP spécifiques à Jitsi).

8. Surveillance des mises à jour et correctifs sécurité

- Vérification régulière de la disponibilité de patchs critiques pour Prosody, Jicofo, et JVB.
- Supervision de l'état d'application des correctifs.

9. Surveillance des tentatives d'accès aux API (JWT, token)

- Surveillance des tentatives d'utilisation frauduleuses ou répétées des **tokens JWT**.
- Surveillance des échecs répétés d'authentification via tokens (possible brute-force **JWT**).

10. Supervision de la performance cryptographique

- Surveiller la latence ou les anomalies dans les échanges des clés cryptographiques.
- Indicateurs liés au fonctionnement du protocole de chiffrement (ex. DTLS-SRTP, AES-GCM) pour détecter toute anomalie dans les échanges de clés.

Résumé des indicateurs clés :

Composants à surveiller	Points de vigilance principaux
Certificats TLS/SSL	Expiration, révocation, protocole minimum
Prosody/XMPP	Brute-force, authentifications, modules JWT/E2EE
Flux réseau	Activité inhabituelle, trafic non autorisé
Configuration E2EE	Désactivation, configuration altérée
Anomalies système	Usage CPU/RAM inhabituel
Logs	Erreurs critiques Prosody/Jicofo/JVB
IDS	Attaques ciblant Jitsi (XMPP, SIP, HTTP)
Gestion des patchs	Mise à jour des composants critiques
API (JWT)	Abus de tokens JWT, tentatives échouées
Performance crypto	Latences ou anomalies clés DTLS-SRTP

Recommandations opérationnelles :

- Intégrer ces indicateurs dans **Prometheus/Grafana**.
- Définir des règles d'alerte pour chaque point.
- Coupler avec un SIEM (**Wazuh**) pour détecter des menaces en temps réel.
- Programmer des vérifications automatiques régulières (scripts Bash/Python via `cron`).