RECEVOIR VOTRE PISTE D'AUDIT DANS

# Microsoft Sentinel

Recevez la piste d'audit REVIEWIT dans votre workspace Log Analytics par la Logs Ingestion API d'Azure Monitor : une règle de collecte, une table, une inscription d'application avec un certificat — pas de Function, pas de Logic App, pas d'agent, pas de secret client.

VERSION
1.0 · Octobre 2026

DURÉE
Quarante-cinq minutes, en trois étapes

DIFFUSION
Confidentiel · Clients REVIEWIT

## CE QU'IL VOUS FAUT

01 **Un administrateur Azure** capable de créer un workspace Log Analytics, un point de terminaison et une règle de collecte, une inscription d'application dans votre tenant Microsoft Entra, et d'attribuer un rôle sur la règle.

02 **Un administrateur de la souscription** REVIEWIT. Créer, activer et remplacer un identifiant exigent une authentification forte de moins de dix minutes : la console vous envoie chez votre fournisseur d'identité et vous ramène.

03 **L'ordre de ce guide** : d'abord la cible chez vous, puis la destination dans REVIEWIT, puis le test. Rien n'est livré avant qu'une écriture réelle sous vos identifiants ait réussi.

LE PARCOURS, EN TROIS ÉTAPES

ÉTAPE 1

### Préparer la cible

Créer le point de réception et l'identifiant que REVIEWIT utilisera — droits minimaux, rien de plus.

ÉTAPE 2

### Déclarer la destination

Quatre écrans dans le Subscription Manager : cible, résidence, contenu, récapitulatif.

ÉTAPE 3

### Tester, puis activer

Une écriture de sonde sous vos identifiants ; un refus dit pourquoi. La livraison commence dans la minute.

DANS CE GUIDE


1 Comment BEACON livre votre piste d'audit

2 Ce que vous recevez

3 Préparer la cible chez Microsoft Sentinel

4 Déclarer la destination dans REVIEWIT

5 Tester et activer

6 Vérifier hors ligne avec beacon-verify

7 Exploiter

8 Dépannage et liste de contrôle

# 01 Comment BEACON livre votre piste d'audit

BEACON est le canal de diffusion de la piste d'audit REVIEWIT. Il ne produit pas la preuve et ne possède pas les événements : **la piste d'audit prouve, REVIEWIT SEAL scelle, BEACON livre.** Chaque événement d'audit devient un maillon — un rang strictement croissant, le SHA-256 de l'enregistrement, l'empreinte du maillon précédent et l'enregistrement original verbatim — et la tête de chaque chaîne est signée par l'autorité de scellement. Ce qui arrive chez vous est une chaîne que vous recalculez et un sceau que vous contrôlez, sans jamais rappeler REVIEWIT.

VOTRE RÉGION

**Piste d'audit**

Événements chaînés par domaine et classe de rétention, scellés toutes les heures par REVIEWIT SEAL.

→

VOTRE RÉGION

**BEACON**

Une passe par minute : des lots de 500 événements ou 5 Mo au plus, un manifeste par lot, un battement toutes les 15 minutes.

→

CHEZ VOUS

**Microsoft Sentinel**

La Logs Ingestion API d'Azure Monitor, dans la table `REVIEWIT_Audit_CL` de votre workspace, sous le certificat de votre inscription d'application.

PRINCIPE DIRECTEUR

Rien n'est perdu et rien n'est prétendu. La livraison avance chaîne par chaîne, sur un acquittement strict seulement ; un échec reprend exactement où il s'est arrêté ; après douze tentatives, la destination est mise en pause et une alerte est levée. Une purge de rétention est livrée comme une coupe signée, un filtre laisse des jalons qui gardent la chaîne vérifiable, et le silence est couvert par le battement : jamais de trou non signé.

PRIS EN CHARGE PAR

La carte **Microsoft Sentinel** de l'assistant de destination (profil Logs Ingestion). L'authentification est le **certificat de votre inscription d'application** : REVIEWIT obtient un jeton Entra par assertion signée (RS256) avec la clé privée que vous confiez — PFX ou PEM — scellée sous garde REVIEWIT SEAL. Aucun secret client, jamais. L'acquittement est le **HTTP 204** d'Azure Monitor. Prouvé de bout en bout le 19 septembre 2026 sur un essai Azure (France Central).

## 02 Ce que vous recevez

Trois sortes d'enregistrements voyagent vers votre cible, toutes en JSON UTF-8 dont l'ordre des propriétés est le contrat. Un **maillon** est un événement d'audit : `metadata.uid` (l'identifiant stable de l'événement), `reviewit.chain` (clé, genèse), `reviewit.integrity` (`sequence`, `previous_hash`, `hash`, `raw_data`). Un **manifeste** clôt chaque lot : chaîne, plage, empreintes d'entrée et de sortie, décomptes, `batch_id`, et la référence du dernier sceau de la chaîne. Un **battement** porte, toutes les 15 minutes, la tête et la position acquittée de chaque chaîne.

| ENREGISTREMENT | OÙ IL ARRIVE | À QUOI IL SERT |
|---|---|---|
| `Kind == "link"` | Une ligne par maillon dans `REVIEWIT_Audit_CL` ; `TimeGenerated` est l'heure de l'occurrence ; `Event` est le maillon | L'événement d'audit, avec ses champs de preuve et `raw_data` verbatim |
| `Kind == "manifest"` | Une ligne par lot, manifeste nu | Plage, empreintes, décomptes, `batch_id`, référence de sceau |
| `Kind == "heartbeat"` | Une ligne toutes les 15 minutes | L'état de chaque chaîne — alertez sur son absence |
| `Kind == "probe"` | Une ligne à la création, à chaque test et à l'activation | L'écriture de sonde qui prouve la règle, le flux et le rôle |

**AU MOINS UNE FOIS, DÉDOUBLONNÉ PAR VOUS**

Après un doute réseau, le même lot peut être présenté de nouveau, avec le même `batch_id` et les mêmes identifiants d'événements. Dédoublonnez sur `metadata.uid`, et ordonnez entre chaînes par l'heure de l'occurrence. Un enregistrement qui n'arrive jamais est un trou que votre SIEM peut voir : chaque maillon est numéroté, et le battement dit où se tient la tête.

**LE PULL, MÊME CONTRAT**

Votre SIEM peut aussi lire la piste lui-même par l'API publique REVIEWIT, sous la portée `audit.read` : `GET /api/v1/audit/chains` liste les chaînes de votre souscription avec leur curseur de départ, `GET /api/v1/audit/stream?cursor=…&limit=500&manifest=true` rend les mêmes maillons et les mêmes manifestes que le push. Push et pull se vérifient avec le même outil.

# 03 Préparer la cible chez Microsoft Sentinel

Un administrateur Azure crée, une fois, le workspace et son rattachement à Sentinel, puis la table et sa règle de collecte, puis l'inscription d'application qui est l'identité de REVIEWIT dans votre tenant. Les libellés ci-dessous sont ceux du portail Azure ; l'assistant du portail exige un point de terminaison de collecte même si une règle directe s'en passerait.

### 01 Workspace Log Analytics, puis Sentinel

Créer une ressource › **Espace de travail Log Analytics** : groupe de ressources, nom (`law-reviewit-beacon`), région — la géographie que vous déclarerez dans REVIEWIT. Puis Microsoft Sentinel › **Ajouter** le workspace.

### 02 Point de terminaison de collecte

Moniteur › Points de terminaison de la collecte de données › **Créer** : nom `dce-reviewit-beacon`, mêmes groupe et région. Notez son **URI d'ingestion des journaux** (`https://dce-…-xxxx.<région>-1.ingest.monitor.azure.com`).

### 03 Table et règle de collecte

Workspace › Paramètres › **Tables** › **Créer** › **Nouveau journal personnalisé (basé sur DCR)**. Nom de table `REVIEWIT_Audit` (le portail ajoute `_CL`), plan Analytique, créez la règle `dcr-reviewit-beacon` depuis le lien, point de terminaison `dce-reviewit-beacon`. Schéma : chargez l'échantillon ci-dessous comme un fichier ; le portail déduit `TimeGenerated` (datetime), `Kind` (string), `Event` (dynamic), transformation `source`. Créer.

### 04 Notez l'ID immuable de la règle

Règle › Vue d'ensemble › **ID immuable** (`dcr-` + 32 caractères hexadécimaux), copié avec le bouton de copie : une lettre fausse et REVIEWIT répond « flux inconnu » (404). L'URL d'ingestion à saisir dans REVIEWIT est `https://<hôte de l'URI d'ingestion>/dataCollectionRules/<id immuable>/streams/Custom-REVIEWIT_Audit_CL?api-version=2023-01-01`, sur une ligne.

### 05 Inscription d'application avec un certificat

Microsoft Entra ID › Inscriptions d'applications › **Nouvelle inscription** : nom `reviewit-beacon`, locataire unique, aucune URI de redirection. Copiez l'**ID de l'annuaire (locataire)** et l'**ID d'application (client)**. Certificats et secrets › Certificats › **Charger un certificat** : le `.cer` public d'une clé que votre PKI a émise pour cette application (RSA 2048 au moins, signature numérique). Gardez l'export **avec la clé privée** (PFX + mot de passe, ou PEM) : il est confié à REVIEWIT à la section 4. Ne créez **aucun** secret client.

06 **Avant de confier le PFX, relisez-le avec le mot de passe que vous saisirez**

C'est exactement le contrôle que fait REVIEWIT ; un mot de passe qui ne correspond pas est refusé sans autre diagnostic. L'empreinte doit être celle qu'affiche Entra.

### FICHIER D'ÉCHANTILLON POUR LE SCHÉMA DE LA TABLE (DEUX LIGNES)

```
[
  { "TimeGenerated": "2026-09-19T12:00:00.0000000Z", "Kind": "link",
    "Event": { "metadata": { "uid": "00000000-0000-0000-0000-000000000001" }, "raw_dat
a": "{\"exemple\":true}",
               "reviewit": { "chain": { "key": "…##BUSINESS#G1", "genesis_at": "2026-09-
19T00:00:00.000000Z" },
                             "integrity": { "sequence": 1, "hash": "0000", "integrity_ve
rsion": 2, "hash_algorithm": "SHA-256" } } } },
  { "TimeGenerated": "2026-09-19T12:00:01.0000000Z", "Kind": "manifest",
    "Event": { "batch_id": "00000000-0000-0000-0000-000000000002", "from_sequence": 1,
"to_sequence": 1 } }
]
```

### CERTIFICAT AUTO-SIGNÉ POUR UN ESSAI, ET CONTRÔLE DE RELECTURE (POWERSHELL)

```
$c = New-SelfSignedCertificate -Subject "CN=reviewit-beacon-client, O=<votre organisatio
n>" -KeyAlgorithm RSA -KeyLength 2048 -HashAlgorithm SHA256 -KeyExportPolicy Exportable
-KeyUsage DigitalSignature -CertStoreLocation Cert:\CurrentUser\My -NotAfter (Get-Date).
AddYears(1)
Export-PfxCertificate -Cert $c -FilePath .\reviewit-beacon.pfx -Password (Read-Host -AsS
ecureString "Mot de passe du PFX")
Export-Certificate  -Cert $c -FilePath .\reviewit-beacon.cer     # celui-ci se charge da
ns Entra

$p = Read-Host -AsSecureString "Mot de passe du PFX"; $plain = [Runtime.InteropServices.
Marshal]::PtrToStringUni([Runtime.InteropServices.Marshal]::SecureStringToGlobalAllocUni
code($p))
$t = [System.Security.Cryptography.X509Certificates.X509Certificate2]::new(".\reviewit-b
eacon.pfx", $plain, "EphemeralKeySet"); $plain = $null
"$($t.Subject) | clé privée : $($t.HasPrivateKey) | $($t.Thumbprint)"   # l'empreinte q
u'affiche Entra
```

### CE QUE VOUS SAISIREZ DANS L'ASSISTANT

| CHAMP DE L'ASSISTANT | VALEUR |
|---|---|
| **Transport** | **Microsoft Sentinel** (Azure Monitor Logs Ingestion API) |
| **Nom** | Comment la destination se lira dans votre liste — « Sentinel — SOC ». |
| **URL de la cible** | L'URL d'ingestion complète de la règle : `https://<point>.<région>-1.ingest.monitor.azure.com/dataCollectionRules/dcr-…/streams/Custom-REVIEWIT_Audit_CL?api-version=2023-01-01`. |
| **ID de l'annuaire (locataire) Entra · ID d'application (client)** | Sur la page Vue d'ensemble de l'inscription d'application. Ce ne sont pas des secrets. |
| **Certificat avec sa clé privée** | Le PFX (avec son mot de passe) ou le PEM exporté à l'étape 05. REVIEWIT vérifie qu'il porte une clé privée RSA non expirée, scelle la clé, garde le certificat public pour l'empreinte et l'échéance. Un `.cer` seul est refusé, en le disant. |
| **Géographie de la cible** | La région du workspace (France Central = FRA). Déclarative. |

# 04 Déclarer la destination dans REVIEWIT

Subscription Manager › **Paramètres** › **Intégrations & IA** › **BEACON** › **Nouvelle destination**. L'assistant a quatre écrans ; rien n'est créé avant le dernier, et la cible est contrôlée par la garde d'egress avant que la destination existe.

### 01 Cible

Nommez la destination, choisissez la carte **Microsoft Sentinel**, saisissez l'URL de la cible et l'identifiant de la section 3. L'identifiant est ouvert en mémoire pour ce seul geste, puis scellé sous garde REVIEWIT SEAL : l'écran ne le montre plus jamais.

### 02 Résidence

Déclarez la **géographie du workspace** (sa région Azure). Pour un point HTTP elle est déclarative. Si elle diffère de la géographie de votre souscription REVIEWIT, une justification écrite est exigée et consignée dans la piste d'audit.

### 03 Contenu

**Départ** : depuis le début de chaque chaîne (historique complet) ou depuis l'activation (seulement ce qui suit). **Événements envoyés en clair** : laissez vide pour le flux complet. Un filtre remplace chaque événement écarté par un jalon — rang et empreintes, aucune donnée métier — et la preuve scellée s'arrête au premier jalon ; l'écran le dit avant que vous confirmiez.

### 04 Récapitulatif

Relisez, puis cliquez **Créer**. La garde d'egress juge d'abord la cible : HTTPS seulement, aucun identifiant dans l'URL, port 443 ou 8088, un hôte routable publiquement. Une cible refusée nomme le contrôle fautif ; rien n'est créé.

**LA GARDE D'EGRESS, AVANT CHAQUE ENVOI**

Chaque cible passe les mêmes sept contrôles à la création, à chaque test et avant chaque livraison : URL valide · HTTPS · aucun identifiant dans l'URL · port 443 ou 8088 · aucune zone interne REVIEWIT · l'hôte se résout · l'hôte n'est pas une adresse privée ou réservée. Les redirections ne sont jamais suivies, et la vérification TLS n'est jamais désactivée.

# 05 Tester et activer

Une destination n'est activée qu'après une écriture réelle sous vos identifiants. Ouvrez le panneau de la destination et cliquez **Tester la cible** : REVIEWIT écrit un enregistrement de sonde (`Kind = probe` sur le flux, après un jeton obtenu par assertion signée). « La cible passe la garde d'egress » signifie que l'hôte, l'identifiant et ses droits sont prouvés. Un refus dit sa cause — jamais le secret.

### 01 Tester la cible

Deux choses sont prouvées en un geste : REVIEWIT obtient un jeton Entra par assertion signée (prouve le certificat), puis écrit une ligne `probe` sur le flux (prouve la règle, le flux et le rôle). Avant de tester, donnez le rôle : règle `dcr-reviewit-beacon` › **Contrôle d'accès (IAM)** › **Ajouter une attribution de rôle** › **Éditeur de métriques d'analyse** (Monitoring Metrics Publisher) › membre `reviewit-beacon` (type Application). C'est le seul droit dont REVIEWIT a besoin — sur la règle seule, pas sur le workspace. Le rôle est effectif dans la minute. Un refus nomme sa cause : « certificat non déposé », « rôle absent sur la règle », « flux inconnu ». La destination est créée en brouillon : activez-la depuis le panneau une fois la sonde passée.

### 02 Activer

Depuis le panneau, **Activer** (ou cochez « Activer dès la création » dans l'assistant). La livraison commence à la passe suivante, dans la minute ; le premier battement suit.

### 03 Lire le panneau

Par chaîne : le point de reprise acquitté, l'âge du plus ancien événement non livré, la marge avant la prochaine coupe de rétention, le dernier sceau et ce qui est au-delà. L'historique des lots liste chaque manifeste ; le journal du canal liste chaque geste d'administration avec son code d'audit.

# 06 Vérifier hors ligne avec beacon-verify

`beacon-verify` est un outil en ligne de commande, distribué avec sa spécification publique. Il lit ce que votre cible a reçu et le trousseau que REVIEWIT publie à `/.well-known/seal-keyring.json`, ancré sur l'empreinte de la clé racine imprimée dans l'annexe sécurité de votre contrat, et rend un verdict par chaîne et par plage — sans réseau et sans appel à REVIEWIT.

THE COMMAND

```
beacon-verify <export> --keys seal-keyring.json --root <empreinte racine> --namespace <v
otre espace de scellement>
```

Votre **espace de scellement** est affiché dans le panneau de la destination (`t-` suivi de 26 caractères). Passez-le avec `--namespace` : un sceau d'un autre espace est, pour vous, une rupture. Sans `--root` ni `--namespace`, les plages sont rendues scellées *sous réserve*.

01 **Requête, sur une fenêtre large**

`REVIEWIT_Audit_CL | where Kind in ("link", "manifest") | project TimeGenerated, Kind, Event`, intervalle **7 derniers jours** ou plus (`TimeGenerated` d'un maillon est l'heure de l'occurrence : une fenêtre courte coupe les lots), limite d'affichage **5 000** ou maximale. Le pied de page doit annoncer le nombre attendu.

02 **Partager › Exporter au format CSV (toutes les colonnes)**

Le portail dépose `query_data.csv`. Ne le retouchez pas : le vérificateur lit le CSV tel quel (en-tête, colonne `Event` en JSON cité), comme les tables JSON de l'API de requête. Log Analytics réécrit les dates des colonnes dynamiques à sept décimales ; sans effet sur les maillons (`raw_data` est une chaîne), et le vérificateur accepte les deux formes pour les dates d'une coupe.

03 **Lancer le vérificateur**

`beacon-verify query_data.csv --keys seal-keyring.json --root <racine> --namespace <t-…>`. L'export de l'essai du 19 septembre 2026 a rendu le code 3 (la chaîne scellée depuis la liaison SCELLÉ ; les chaînes antérieures NON VÉRIFIABLE jusqu'à leur sceau suivant) ; un caractère altéré dans un `raw_data` a rendu ROMPU à son rang, code 1.

| VERDICT | SIGNIFICATION |
|---|---|
| `SCELLÉ` | Chaîné jusqu'à une tête dont la signature est valide, trousseau ancré, espace épinglé. |
| `SCELLÉ SOUS RÉSERVE` | Idem, mais la racine ou l'espace n'ont pas été épinglés sur la ligne de commande. |
| `CHAÎNAGE SEUL` | Intact, au-delà du dernier sceau ou derrière un jalon — le prochain sceau le couvrira. |
| `NON VÉRIFIABLE` | Un sceau sans espace, un algorithme inconnu ou une clé absente du trousseau — jamais dit faux. |
| `ROMPU` | Une empreinte fausse, un trou, un doublon divergent, une signature invalide ou un espace étranger — avec la cause et le rang. |

Codes de sortie : **0** rien de rompu et tout ce qui précède le dernier sceau est scellé · **1** une rupture ou un espace étranger · **2** entrée inexploitable (une propriété JSON en double, une valeur à la fois brute et enveloppée, un enregistrement tronqué à l'indexation) · **3** rien de rompu, preuve incomplète.

# 07 Exploiter

Chaque geste sur une destination est posé par un administrateur de la souscription, tracé dans la piste d'audit sous son nom, et les gestes sensibles exigent une authentification forte de moins de dix minutes. Le canal qui exporte la piste est lui-même sous la piste.

| | |
|---|---|
| **Renouveler le certificat** | C'est votre échéance et votre geste : trente jours avant, le panneau et le condensé de santé de REVIEWIT l'annoncent. Chargez le nouveau certificat sur l'inscription d'application (Entra en accepte plusieurs), puis panneau › **Remplacer le certificat** : le formulaire s'ouvre dans le panneau (PFX ou PEM, mot de passe, Confirmer). L'ancienne clé cesse de signer aussitôt et le panneau affiche la nouvelle empreinte ; retirez ensuite l'ancien certificat d'Entra. Un fichier refusé (sans clé, non RSA, expiré, mot de passe faux) ne change rien. |
| **Pause pour échecs** | Un refus persistant — identifiant révoqué, droit retiré, cible disparue — met la destination en pause après douze tentatives et lève une alerte dans le Centre d'alertes. Rien n'est perdu : la reprise repart du point de reprise acquitté de chaque chaîne, automatiquement quand la sonde horaire revoit la cible, ou par **Reprendre**. |
| **Changer de cible** | **Modifier** › nouvelle URL, même forme. Le certificat et l'identité sont conservés ; attribuez le rôle sur la nouvelle règle, puis **Tester la cible**. |
| **Retirer** | **Désactiver**, puis supprimer ou archiver. Une destination qui a livré est archivée, jamais supprimée : ses manifestes sont des preuves de livraison. Révoquez ensuite l'identifiant de votre côté : il ne sert plus à rien. |

# 08 Dépannage

| CE QUE VOUS VOYEZ | CAUSE | CE QU'IL FAUT FAIRE |
|---|---|---|
| **« Certificat non déposé » au test** | Le `.cer` chargé dans Entra n'est pas celui dont vous avez confié la clé privée | Comparez l'empreinte SHA-1 du panneau (« Voir le certificat ») à celle d'Entra ; chargez le bon `.cer` ou remplacez le certificat dans le panneau. |
| **« Rôle absent sur la règle » au test** | Éditeur de métriques d'analyse non attribué sur la DCR, ou attribué sur le workspace | Attribuez-le sur la règle `dcr-reviewit-beacon`, membre `reviewit-beacon` (Application) ; attendez une minute ; retestez. |
| **« Flux inconnu » (404)** | Un caractère faux dans l'ID immuable, ou un nom de flux autre que `Custom-REVIEWIT_Audit_CL` | Copiez l'ID immuable avec le bouton de copie ; vérifiez la vue JSON de la règle (`streamDeclarations`, `outputStream`). |
| **Le PFX est refusé : « mot de passe faux »** | Le mot de passe ne correspond pas, ou l'export n'a pas de clé privée | Faites le contrôle de relecture de la section 3 ; réexportez depuis votre magasin de certificats avec la clé privée. |
| **Destination en pause, « après échecs »** | Certificat retiré ou expiré, rôle retiré, application supprimée | Corrigez côté Azure, **Tester la cible**, puis **Reprendre**. Rien n'a été perdu. |
| **Private Link / point de terminaison privé** | Le point d'ingestion n'est joignable qu'en privé | La garde d'egress de REVIEWIT n'atteint que des adresses publiques ; gardez le point d'ingestion public pour cette règle. |

## Liste de contrôle avant de commencer

8 POINTS

- [ ] Un workspace Log Analytics, Sentinel ajouté, dans la région que vous déclarerez
- [ ] Un point de terminaison et une règle de collecte, table `REVIEWIT_Audit_CL` à trois colonnes
- [ ] L'ID immuable de la règle et l'URI d'ingestion du point, assemblés en URL d'ingestion
- [ ] Une inscription d'application `reviewit-beacon` avec un certificat chargé, aucun secret client
- [ ] Le PFX ou PEM avec la clé privée, relu avec le mot de passe que vous saisirez
- [ ] Éditeur de métriques d'analyse attribué sur la règle à l'application
- [ ] Un administrateur de la souscription avec son second facteur prêt
- [ ] L'empreinte racine de l'annexe de votre contrat, pour beacon-verify

Ce guide est fourni aux clients REVIEWIT qui raccordent leur SIEM ou leur stockage à BEACON. Les chemins de menus sont ceux des consoles des éditeurs à la date de publication ; ils peuvent évoluer. En cas d'écart, le support REVIEWIT vous accompagne.