RECEVOIR VOTRE PISTE D'AUDIT DANS

Point HTTPS

Recevez la piste d'audit REVIEWIT sur un collecteur que vous exploitez : un POST HTTPS signé par lot, NDJSON gzippé, une signature HMAC-SHA256 avec un secret que REVIEWIT engendre et montre une fois — le contrat générique pour tout collecteur.

VERSION
1.0 · Octobre 2026

DURÉE
Trente minutes, en trois étapes

DIFFUSION
Confidentiel · Clients REVIEWIT

CE QU'IL VOUS FAUT

01 **Un exploitant du collecteur** capable d'exposer un point HTTPS sur une adresse publique, en 443 ou 8088, avec un certificat reconnu publiquement ou son empreinte, et d'implémenter le contrôle de signature ci-dessous.

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 Point HTTPS

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

Point HTTPS

Votre collecteur : un POST par lot et par battement, corps NDJSON gzippé, en-têtes `X-Beacon-*`, signature HMAC.

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 **Point HTTPS** de l'assistant de destination (profil HTTP egress). REVIEWIT **engendre** le secret HMAC et le montre une fois ; seule son enveloppe scellée est conservée, et chaque lot est signé par l'office de l'autorité de scellement — jamais par un processus qui tiendrait le secret en clair. Acquittement : tout **2xx** ; un 3xx est un échec, les redirections ne sont jamais suivies. La résidence est déclarative.

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
`Corps du POST : les maillons, puis la ligne du manifeste`	Une requête par lot, `Content-Type: application/x-ndjson`, `Content-Encoding: gzip`	Les maillons en NDJSON, puis `{"reviewit_manifest/v1":{…}}` en dernière ligne
`X-Beacon-Batch-Id · X-Beacon-Chain · X-Beacon-Sequence-Range`	En-têtes de requête d'un lot	Identifient le lot, sa chaîne et sa plage `from-to` sans lire le corps
`X-Beacon-Timestamp · X-Beacon-Signature`	Sur chaque requête	Secondes Unix, et `v1=<base64>[,v1=<base64>]` — deux signatures pendant une rotation ouverte
`X-Beacon-Kind: heartbeat`	Une requête toutes les 15 minutes, corps = une ligne JSON gzippée	L'état de chaque chaîne — alertez sur son absence

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 Point HTTPS

Votre collecteur expose un point HTTPS qui accepte un POST, vérifie la signature, conserve le corps tel que reçu, et ne répond 2xx qu'une fois le lot durablement écrit. La garde d'egress de REVIEWIT exige une adresse publique, le port 443 ou 8088, et ne suit jamais une redirection.

01 Exposer le point

`https://<hôte>[:8088]/<chemin>` sur une adresse routable publiquement, HTTPS seulement. Un certificat reconnu publiquement n'a besoin de rien ; un certificat d'AC privée peut être **épinglé** par votre administrateur depuis le panneau de la destination (section 7).

02 Vérifier la signature avant de conserver

Décodez `X-Beacon-Signature` (`v1=…`, éventuellement deux valeurs pendant une rotation), recalculez `HMAC-SHA256(secret, "{timestamp}.{sujet}.{sha256}")` où `timestamp` est `X-Beacon-Timestamp`, `sujet` est `X-Beacon-Batch-Id` (ou le littéral `heartbeat` quand `X-Beacon-Kind: heartbeat`), et `sha256` est le SHA-256 hexadécimal minuscule des **octets du corps tels qu'ils voyagent — les octets gzippés**. Acceptez si l'une des valeurs `v1=` correspond ; rejetez un horodatage vieux de plus de quelques minutes.

03 Conserver le corps tel que reçu, puis acquitter

Gardez le corps gzippé (ou le NDJSON décompressé, inaltéré) : `raw_data` doit rester verbatim pour la vérification hors ligne. Répondez **2xx seulement après l'écriture durable** : un lot acquitté n'est jamais renvoyé, sauf après un doute réseau — dédoublonnez sur `X-Beacon-Batch-Id`, puis sur `metadata.uid`.

04 Alerter sur le silence

Un battement arrive toutes les 15 minutes avec la tête et la position acquittée de chaque chaîne. Levez une alerte quand aucun n'est arrivé depuis 30 minutes : le silence n'est pas une information.

CONTRÔLE DE SIGNATURE (PYTHON, RÉFÉRENCE)

```
import base64, hashlib, hmac

def verify(secret: str, headers: dict, body_bytes: bytes) → bool:
    subject = headers.get("X-Beacon-Batch-Id") or ("heartbeat" if headers.get("X-Beacon-
Kind") == "heartbeat" else None)
    digest = hashlib.sha256(body_bytes).hexdigest()            # les octets gzippés, te
ls qu'ils voyagent
    message = f'{headers["X-Beacon-Timestamp"]}.{subject}.{digest}'.encode("utf-8")
    expected = hmac.new(secret.encode("utf-8"), message, hashlib.sha256).digest()
    for part in headers["X-Beacon-Signature"].split(","):     # "v1=…" ou "v1=neuve,v1=a
ncienne" pendant une rotation
        if hmac.compare_digest(base64.b64decode(part.split("=", 1)[1]), expected):
            return True
    return False
```

CE QUE VOUS SAISIREZ DANS L'ASSISTANT

CHAMP DE L'ASSISTANT	VALEUR
Transport	**Point HTTPS** (NDJSON, signature HMAC)
Nom	Comment la destination se lira dans votre liste — « Collecteur — SOC ».
URL de la cible	`https://<hôte>[:8088]/<chemin>`. HTTPS seulement, port 443 ou 8088, un hôte routable publiquement, aucun identifiant dans l'URL, aucune redirection.
Secret HMAC	Rien à saisir : REVIEWIT l'**engendre** à la création et le montre **une fois** (`bcsec_…`). Copiez-le à ce moment dans la configuration de votre collecteur ; il n'est plus jamais affiché.
Géographie de la cible	Là où votre collecteur garde les données. 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 **Point HTTPS**, 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 où le collecteur conserve la piste**. Pour un point HTTP elle ne s'observe pas, 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 (une requête signée, semblable à un battement). « 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

REVIEWIT envoie une requête signée à votre point et attend un 2xx. Un refus nomme sa cause : un statut HTTP (401/403 — votre collecteur a rejeté la signature : vérifiez le secret et le condensat du corps gzippé ; 404 — mauvais chemin ; 3xx — une redirection, jamais suivie), ou un refus TLS (épinglez le certificat, section 7).

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 **Garder ce qui est arrivé**

Votre collecteur a conservé chaque corps tel que reçu : NDJSON gzippé (ou le NDJSON en clair). Rassemblez les corps d'une chaîne dans un dossier ; le vérificateur lit aussi bien le gzip (1F 8B) que les fichiers en clair, un dossier en entier.

02 **Lancer le vérificateur**

`beacon-verify ./collector-store --keys seal-keyring.json --root <racine> --namespace <t-…>`. Code 0 quand tout ce qui précède le dernier sceau est scellé ; 3 quand une chaîne n'a pas encore de sceau sous votre espace.

03 **Vérifier une signature de transport depuis une requête capturée**

Si vous avez gardé les en-têtes de la requête avec le corps : `beacon-verify hmac <corps> --timestamp <X-Beacon-Timestamp> --subject <X-Beacon-Batch-Id|heartbeat> --signature <X-Beacon-Signature>`, avec le secret dans `BEACON_HMAC_SECRET`. La signature de transport prouve l'origine et l'intégrité sur le fil ; seul le sceau SEAL prouve la piste.

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.

Faire tourner le secret HMAC	Panneau › **Faire tourner le secret** : le nouveau secret est montré une fois ; les deux secrets signent chaque requête (`v1=neuve,v1=ancienne`) jusqu'à ce que vous cliquiez **Clôturer la rotation**, une fois que votre collecteur vérifie avec le nouveau. Une seconde rotation est refusée tant qu'une est ouverte.
Épingler le certificat du collecteur	Panneau › Transport › « Lire le certificat TLS de la cible » montre le sujet, l'émetteur, l'échéance et l'empreinte SHA-256 lus par la garde d'egress. Comparez-la à celle que votre exploitant vous a donnée, puis **Faire confiance à ce certificat TLS** (authentification forte, tracée). Les livraisons acceptent ensuite les certificats reconnus publiquement ou exactement celui-ci. Un certificat renouvelé doit être ré-épinglé ; l'épinglage part avec la cible quand l'URL change.
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 secret est conservé ; un certificat épinglé est abandonné et doit être ré-épinglé.
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
Votre collecteur rejette toutes les signatures	Le condensat a été calculé sur le corps décompressé, ou le sujet est faux	Hachez les octets gzippés tels que reçus ; sujet = `X-Beacon-Batch-Id`, ou le littéral `heartbeat`.
HTTP 3xx à chaque envoi	Votre point redirige (http → https, barre finale)	Saisissez l'URL finale ; REVIEWIT ne suit jamais une redirection.
La garde d'egress refuse la cible	Port autre que 443/8088, adresse privée, identifiants dans l'URL	Exposez le collecteur en 443 ou 8088 sur une adresse publique ; gardez l'URL propre.
Une erreur TLS à chaque envoi	Certificat d'AC privée ou auto-signé	Épinglez-le depuis le panneau (section 7). La vérification n'est jamais désactivée.
Le même lot arrive deux fois	Un doute réseau après un 2xx	Normal (au moins une fois) : dédoublonnez sur `X-Beacon-Batch-Id`, puis `metadata.uid`.
Destination en pause, « après échecs »	Douze réponses non-2xx consécutives	Lisez la cause dans le panneau, corrigez le collecteur, **Tester la cible**, puis **Reprendre**. Rien n'a été perdu.

Liste de contrôle avant de commencer

8 POINTS

- [ ] Un point HTTPS sur une adresse publique, port 443 ou 8088, sans redirection
- [ ] Un certificat reconnu publiquement, ou son empreinte SHA-256 prête à épingler
- [ ] Le contrôle de signature implémenté sur les octets gzippés, les deux valeurs `v1=` acceptées
- [ ] Une écriture durable avant le 2xx, dédoublonnage sur les identifiants de lot et d'événement
- [ ] Une alerte sur 30 minutes sans battement
- [ ] La géographie du collecteur et sa justification si elle diffère de celle de votre souscription
- [ ] Un administrateur de la souscription avec son second facteur prêt, prêt à copier le secret une fois
- [ ] 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.