CERTYX

AUTONOMOUS CERTIFICATE LIFECYCLE MANAGEMENT

SIEM-Integration

Export des Sicherheitsjournals an Splunk, Cribl, IBM QRadar, ArcSight, Microsoft Sentinel und Webhooks

SPRACHE	REVISION	DATUM
Deutsch	8550060f	1. November 2026

Inhalt

1 Was CERTYX exportiert 5
1.1 Verarbeitungskette 5
1.2 Transporte 6
1.3 Formate 6
1.4 Schemaprofile 6
2 Zulässige Kombinationen 7
2.1 Ziele der ersten Version 7
3 Schnellstart 8
4 Exportierte Ereignisse 8
5 Bevor Sie beginnen 10
6 Ein Ziel in CERTYX hinzufügen 10
7 TLS und vertrauenswürdige Stammzertifikate 12
7.1 Clientzertifikat (mTLS) 12
7.2 Sperrprüfung 12
8 Signierter HTTPS-Webhook 13
8.1 Empfängerseite 13
8.2 Signatur prüfen 13
8.3 CERTYX-Seite 15
9 Splunk HTTP Event Collector 15
9.1 Den Splunk-Weg wählen 15
9.2 Empfängerseite (Splunk) 15
9.3 CERTYX-Seite 17
9.4 props.conf 18
9.5 Splunk CIM 18
10 Syslog über TLS 19
10.1 RFC-5424-Header der kanonischen Formate 19
10.2 Framing 20
10.3 CERTYX-Seite (alle Syslog-Formate) 20
10.4 Allgemeiner Syslog-Empfänger (rsyslog, syslog-ng) 20
11 Cribl Stream 21
11.1 Die Cribl-Quelle wählen 21
11.2 Empfängerseite (Cribl-Syslog-Quelle) 21
11.3 Empfängerseite (Cribl-Splunk-HEC-Quelle) 21
11.4 CERTYX-Seite 22
12 Splunk-TCP-SSL-Eingang 22
12.1 Empfängerseite (Splunk) 22
12.2 CERTYX-Seite 23
13 IBM QRadar 23
13.1 Das LEEF-Format wählen 23
13.2 Empfängerseite (QRadar) 23
13.3 CERTYX-Seite 24
14 ArcSight 24
14.1 Das CEF-Format wählen 24
14.2 Empfängerseite (ArcSight SmartConnector) 25
14.3 CERTYX-Seite 25
15 Elastic (ECS) 25

16 OCSF-Verarbeiter und Amazon Security Lake 26
17 Microsoft Sentinel 26
17.1 Was die Tabelle erhält 27
17.2 Empfängerseite (Azure) 27
17.3 CERTYX-Seite 28
18 OpenTelemetry Logs (OTLP/HTTP) 29
18.1 Was der Empfänger erhält 29
18.2 Empfängerseite (OpenTelemetry Collector) 30
18.3 Grafana Alloy und Loki 31
18.4 Andere OTLP-Backends 31
18.5 CERTYX-Seite 31
19 Verbindungsprüfung und Testereignis 31
20 Zustellung, Wiederholungen und Rückstand 32
20.1 Abgelehnte Ereignisse 33
20.2 Grenzen 33
21 Erneut senden ab einer Sequenz oder einem Zeitpunkt 33
22 Deduplizierung nach event_id 34
23 Nur senden: weniger Daten an das SIEM 34
23.1 Wie die Filter zusammenwirken 35
24 Ereignisgestaltung: Größe und Form 35
25 Feldzuordnung 37
26 Anreicherungsfelder 38
27 Mehrere Organisationen in einem SIEM 39
28 Dateiarchive 40
28.1 Einrichtung 40
28.2 Dateien 41
28.3 Herunterladen 41
28.4 Zustellung, Neustarts und Speicherplatz 41
29 Berechtigungen 42
30 Fehlerbehebung 42
30.1 Vorfälle 42
30.2 Ergebnisse von Prüfung und Test 42
30.3 Empfänger zeigt Fragmente oder binäre Daten 43
30.4 Splunk-HEC-Codes 44
30.5 Zustellstatus auf der Zielkarte 45
30.6 Formular- und Berechtigungsmeldungen 45
31 Beispielereignisse 46
31.1 HTTPS-Webhook 46
31.2 Splunk HEC 47
31.3 Syslog über TLS 48
31.4 Microsoft Sentinel 49
31.5 OpenTelemetry Logs 50
31.6 Dateiarchiv 51
32 Feldreferenz 52
32.1 Kanonisches Ereignis 52
32.2 JSON-Datensatz 53
32.3 CEF-Felder (erste Version) 53
32.4 LEEF-Felder (erste Version) 54

32.5 Sentinel-Spalten 54
33 Ereigniscode-Katalog 55
33.1 Nach Ergebnis gewählt 55
33.2 Journalaktionen 56

CERTYX exportiert sein Sicherheitsjournal fortlaufend in das SIEM, das Sie bereits betreiben. Dieses Handbuch erläutert, was exportiert wird, wie Format, Schemaprofil und Transport gewählt werden, und wie Sie beide Seiten – das empfangende System und CERTYX – Schritt für Schritt für Splunk, Cribl Stream, IBM QRadar, ArcSight, Elastic, OCSF-Verarbeiter, Microsoft Sentinel, OpenTelemetry-Backends und Ihren eigenen Collector konfigurieren.

1 Was CERTYX exportiert

Jedes Ziel erhält dieselben Ereignisse aus dem Sicherheitsjournal. Ein Ziel wird durch drei voneinander unabhängige Entscheidungen festgelegt:

- **Transport** – wie die Ereignisse übertragen werden: signierter HTTPS-Webhook, Splunk HTTP Event Collector, Syslog über TLS, die Azure Monitor Logs Ingestion API (Microsoft Sentinel), OpenTelemetry Logs über OTLP/HTTP oder Tagesdateien in einem serverseitigen Archiv.
- **Format** – wie ein Ereignis kodiert wird: JSON, NDJSON, RFC-5424-Syslog, CEF, LEEF 1.0, LEEF 2.0, Generic KV, CSV, die Sentinel-Tabellenzeile oder eine OTLP-Logs-Anfrage (Protobuf oder OTLP/JSON).
- **Schemaprofil** – welche Feldnamen ein JSON-Dokument verwendet: das kanonische CERTYX-Ereignis (`native`), Elastic Common Schema (`ecs`), OCSF (`ocsf`) oder das kanonische Ereignis mit indizierten Splunk-CIM-Feldern (`splunk-cim`).

Nur die Kombinationen unter Zulässige Kombinationen können gespeichert werden. Jedes Ziel steht dem Organisationsadministrator in der Ansicht **SIEM-Export** und dem Plattformbetreiber unter **Journal der Sicherheitsereignisse → SIEM-Export** zur Verfügung. Jedes Ziel führt seine eigene Position im Journal: Die Zustellung erfolgt fortlaufend, mindestens einmal, und nach einem Fehler geht nichts verloren.

Nur verschlüsselter Transport. CERTYX verwendet immer TLS, prüft immer das Zertifikat des Empfängers – gegen den Zertifikatspeicher des Betriebssystems oder genau gegen die Stammzertifikate, die Sie für das Ziel einfügen – und prüft den Sperrstatus. Reines TCP, UDP-Syslog und ungeprüftes TLS werden nicht angeboten. OpenTelemetry Logs werden über OTLP/HTTP mit TLS übertragen; OTLP über gRPC wird nicht angeboten.

1.1 Verarbeitungskette

1. **Journal.** Jedes Sicherheitsereignis wird in das hashverkettete Sicherheitsjournal in der Plattformdatenbank geschrieben. Das Journal ist die dauerhafte Warteschlange jedes Ziels.
2. **Kanonisches Ereignis.** Beim Export wird jeder Journaleintrag einem kanonischen Ereignis der Schemaversion `certyx.event/1.0` zugeordnet, zusammen mit dem Exportkontext (Installations-ID, Produktversion, Umgebung, Host) und der Anreicherung des Eintrags (Trace-ID, Quellport, Serveradresse, Rolle, Sitzungskorrelations-ID – siehe Anreicherungsfelder). Der gehashte Journaleintrag selbst wird nie verändert.
3. **Schwärzung.** Aufgezeichnete Änderungen durchlaufen einen Schwärzungsschritt, der in jeder Tiefe jede Eigenschaft entfernt, deren Name auf ein Passwort, Geheimnis, Token, einen Schlüssel, ein Cookie oder Anmeldedaten hinweist; `changes_redacted` ist `true`, wenn etwas entfernt wurde.
4. **Filter.** Zustandsprüfungen, Ergebnisse, Mindestschweregrad, Kategorien und Ereigniscodes entscheiden, ob das Ziel das Ereignis erhält (siehe Nur senden). Dateiarchive behalten jedes Ereignis.
5. **Feldzuordnung.** Die eigenen Regeln des Ziels benennen Felder des kanonischen Ereignisses um, entfernen, behalten oder ergänzen sie – einmal, bevor ein Format es schreibt (Feldzuordnung).
6. **Serialisierung.** Format und Schemaprofil wandeln das Ereignis in eine Zeile oder ein Dokument um.
7. **Transport.** Der Stapel wird gesendet; die Position des Ziels rückt erst vor, nachdem der Empfänger ihn angenommen hat.

Jedes Ereignis trägt eine weltweit eindeutige `event_id` = `<installation id>:<journal sequence>`, zum Beispiel `7d4c2a10-…-9e1f:48213`. Die Installations-ID ist `Clm:Siem:InstallationId`, wenn der Betreiber sie festlegt, andernfalls eine einmal erzeugte und in der Plattformdatenbank gespeicherte UUID, sodass jede Instanz und jeder Neustart dieselbe verwendet. Empfänger deduplizieren nach `event_id` (siehe Deduplizierung).

1.2 Transporte

Transport	ID	Funktion	Typischer Empfänger
Signierter HTTPS-Webhook	`https`	HTTPS- `POST` eines Stapels, signiert mit HMAC-SHA256 in `X-Certyx-Signature`	Ihr eigener Collector, Logstash, SOAR, eine Serverless-Funktion, eine Cribl-HTTP-Quelle
Splunk HTTP Event Collector	`splunk-hec`	HTTPS zu `/services/collector/event` mit einem HEC-Token, optional mit Indexer Acknowledgement	Splunk Enterprise, Splunk Cloud, Cribl-Splunk-HEC-Quelle
Syslog über TLS	`syslog-tls`	RFC-5424-Nachrichten über TCP mit TLS, Oktettzählung (RFC 5425) oder Zeilenumbruch-Framing	Cribl Stream, QRadar, ArcSight SmartConnector, Splunk-TCP-Eingang, rsyslog, syslog-ng, Logstash, Elastic Agent
Microsoft Sentinel	`azure-monitor`	Logs Ingestion API über eine Datensammlungsregel, authentifiziert mit einer Microsoft-Entra-Anwendung	Microsoft Sentinel, Log Analytics
OpenTelemetry Logs	`otlp-http`	HTTPS- `POST` einer OTLP- `ExportLogsServiceRequest` (Protobuf oder OTLP/JSON, optional gzip und ein Authentifizierungs-Header)	OpenTelemetry Collector, Grafana Alloy und Loki, jedes Backend mit OTLP/HTTP-Logs-Endpunkt
Dateiarchiv	`file`	Tagesdateien in einem serverseitigen Archivverzeichnis, aus der Konsole herunterladbar	Langfristige Aufbewahrung, Stapelimport (zum Beispiel OCSF-Pipelines)

1.3 Formate

Format	ID	Inhalt
JSON-Datensatz	`json`	Der CERTYX-Journaldatensatz der ersten Version (`Sequence` , `At` , `Category` , `EventCode` , `Tenant` , …) plus `EventId` ; ein JSON-Array pro Webhook-Anfrage, das HEC- `event` oder der Text der Syslog-Nachricht. Anführungszeichen in Zeichenketten werden als `\"` geschrieben und Zeichen wie `+` bleiben unverändert (nicht `\u0022` oder `\u002B`), sodass die Schlüssel-Wert-Extraktion keine Felder erfindet und Textsuchen treffen. `Changes` enthält verschachtelte Objekte, keine Zeichenfolgen mit maskierten Anführungszeichen (auch bei älteren Ereignissen)
NDJSON	`ndjson`	Ein JSON-Dokument des Schemaprofils pro Zeile (`application/x-ndjson`), UTF-8 ohne Byte Order Mark
RFC 5424	`rfc5424`	Eine RFC-5424-Syslog-Nachricht mit strukturierten Daten, deren MSG das JSON-Dokument des Schemaprofils ist
CEF	`cef`	ArcSight CEF 0 aus dem kanonischen Ereignis (Device Event Class ID = Ereigniscode, `externalId` = Ereignis-ID)
LEEF 1.0	`leef-1.0`	IBM QRadar LEEF 1.0 mit TAB-getrennten Attributen
LEEF 2.0	`leef-2.0`	IBM QRadar LEEF 2.0 mit dem im Header deklarierten Attributtrennzeichen (standardmäßig TAB)
Generic KV	`kv`	Das dokumentierte Profil `certyx-kv/1.0` : jedes kanonische Feld als `key=value` in fester Reihenfolge; weder CEF noch LEEF
CSV	`csv`	Kanonische CSV-Zeilen nach RFC 4180 mit der versionierten Kopfzeile (Dateiarchiv)
Sentinel-Zeile	`sentinel`	Die Spalten der benutzerdefinierten Sentinel-Tabelle plus `EventId`
OTLP Protobuf	`otlp-protobuf`	OpenTelemetry-Logs-Anfrage im binären Protobuf (`application/x-protobuf`), ein Log-Eintrag je kanonischem Ereignis
OTLP/JSON	`otlp-json`	Dieselbe Anfrage in der OTLP/JSON-Kodierung (`application/json`)
CEF (erste Version)	`cef-compat`	Das CEF-Layout der ersten Version plus `certyxEventId=`
LEEF (erste Version)	`leef-compat`	Das LEEF-2.0-Layout der ersten Version (Trennzeichen `^`) plus `eventId=`
CSV (erste Version)	`csv-compat`	Die Spalten des CSV-Archivs der ersten Version (`certyx-security-YYYY-MM-DD.csv`), unverändert

1.4 Schemaprofile

Profil	ID	Feldnamen
Kanonisch	`native`	Das kanonische CERTYX-Ereignis `certyx.event/1.0` in snake_case (`event_id` , `event_code` , `category` , `outcome` , …); siehe Feldreferenz

Profil	ID	Feldnamen
Elastic Common Schema	`ecs`	ECS 8.11.0: `@timestamp`, `event.*` (`event.id` = Ereignis-ID, `event.code` = Ereigniscode, `event.dataset` = `certyx.security`), `user.*`, `source.ip`, `host.hostname`, `service.*`, `http.*`, `url.path`, `organization.*`, `labels.*`
OCSF	`ocsf`	OCSF 1.1.0, Klasse je Ereignis gewählt und geprüft: Authentication (3002), Account Change (3001), API Activity (6003), andernfalls Base Event (0); `metadata.uid` = Ereignis-ID
Splunk CIM	`splunk-cim`	Das kanonische Ereignis als HEC-`event`, plus die Splunk-CIM-Felder (Datenmodelle Authentication und Change) als indizierte HEC-Felder; nur mit dem HEC-Transport

Die vollständige Zuordnung jedes kanonischen Felds zu CEF-Schlüsseln, LEEF-Attributen, ECS, OCSF und CIM steht im Feldreferenzdokument `docs/SIEM-EXPORT-FIELDS.md` der CERTYX-Dokumentation; die wichtigsten Felder sind unter Feldreferenz zusammengefasst.

2 Zulässige Kombinationen

Wählen Sie zuerst den Transport; das Formular bietet dann dessen Formate und für jedes Format dessen Profile an.

Transport	Format	Profile
`https`	`ndjson`	`native`, `ecs`, `ocsf`
`https`	`json`	`native`
`splunk-hec`	`ndjson`	`native`, `ecs`, `ocsf`, `splunk-cim`
`splunk-hec`	`json`	`native`
`syslog-tls`	`rfc5424`	`native`, `ecs`, `ocsf`
`syslog-tls`	`cef`	`native`
`syslog-tls`	`leef-1.0`	`native`
`syslog-tls`	`leef-2.0`	`native`
`syslog-tls`	`kv`	`native`
`syslog-tls`	`json`	`native`
`syslog-tls`	`cef-compat`	`native`
`syslog-tls`	`leef-compat`	`native`
`azure-monitor`	`sentinel`	`native`
`otlp-http`	`otlp-protobuf`	`native`
`otlp-http`	`otlp-json`	`native`
`file`	`ndjson`	`native`, `ecs`, `ocsf`
`file`	`csv`	`native`
`file`	`csv-compat`	`native`

Jede andere Kombination wird mit *Format, Schemaprofil und Transport: Wählen Sie einen Transport und dann eines der dafür angebotenen Formate und Profile.* abgelehnt. Dateiarchive können nur in der Konsole eingerichtet werden, nicht in der Serverkonfiguration.

2.1 Ziele der ersten Version

Ziele, die gespeichert wurden, bevor Format, Profil und Transport getrennte Entscheidungen wurden – und Konfigurationsziele, die nur `Kind` festlegen –, behalten ihr Layout und ihre Felder. Ihr Syslog-Header enthält die Prozess-ID und `[timeQuality tzKnown="1"]`, wo bisher der Platzhalter `-` stand. Ihre Art ist ein Alias einer Route:

Art (erste Version)	Transport	Format	Ergänzte Ereignis-ID
`webhook`	`https`	`json`	`EventId` in jedem Datensatz
`syslog`	`syslog-tls`	`json`	`EventId` in jedem Datensatz
`cef`	`syslog-tls`	`cef-compat`	Erweiterung `certyxEventId=`
`leef`	`syslog-tls`	`leef-compat`	Attribut `eventId=`
`splunk-hec`	`splunk-hec`	`json`	indiziertes Feld `event_id`
`sentinel`	`azure-monitor`	`sentinel`	Spalte `EventId`
`csv-archive`	`file`	`csv-compat`	keine: Das Layout ist unverändert; `Sequence` ergibt zusammen mit der Installations-ID die Ereignis-ID

Die Felder der Ereignis-ID werden nur ergänzt: Für die erste Version geschriebene Parser funktionieren weiter. Um den kanonischen Ereigniscode, die Kategorie und die übrigen kanonischen Felder zu erhalten, wählen Sie ein kanonisches Format (`ndjson`, `rfc5424`, `cef`, `leef-1.0`, `leef-2.0`, `kv`, `csv`). Für ein Dateiarchiv mit `event_id` in jeder Zeile verwenden Sie das kanonische `csv` oder das `ndjson`-Archiv.

3 Schnellstart

1. Melden Sie sich als Admin der Organisation mit MFA an und öffnen Sie **SIEM-Export** (Betreiber: **Journal der Sicherheitsereignisse → SIEM-Export**).
2. Bereiten Sie den Empfänger wie in seinem Abschnitt beschrieben vor: Splunk, Cribl, QRadar, ArcSight, Elastic, OCSF, Sentinel oder Webhook.
3. Geben Sie einen Zielnamen und eine Kennung ein und wählen Sie **Transport**, **Format** und **Schemaprofil** (empfohlene Ausgangspunkte: Splunk `splunk-hec` × `ndjson` × `native` oder `splunk-cim`; Cribl `syslog-tls` × `rfc5424` × `native`; QRadar `syslog-tls` × `leef-2.0`; ArcSight `syslog-tls` × `cef`; Elastic `https` × `ndjson` × `ecs`; OCSF-Pipelines `https` oder `file` × `ndjson` × `ocsf`; OpenTelemetry-Backends `otlp-http` × `otlp-protobuf`, siehe OpenTelemetry Logs).
4. Füllen Sie die Felder für Adresse, Geheimnis und Vertrauen aus, optional Filter, Zustellwerte und ein Clientzertifikat, und speichern Sie.
5. Wählen Sie **Verbindung prüfen** und anschließend **Testereignis senden**. Das Ergebnis nennt die `event_id` des Testereignisses (eine UUID); suchen Sie im Empfänger danach.
6. Beobachten Sie die Zielkarte: Sie zeigt die Position, die noch wartenden Ereignisse, die zugestellten Ereignisse, den letzten Erfolg und den letzten Fehler.

4 Exportierte Ereignisse

Der Export liest das CERTYX-Sicherheitsjournal. Die Ziele und das Archiv einer Organisation erhalten nur die Ereignisse dieser Organisation und nie Plattformeinträge, auch wenn diese sie betreffen: Aktionen und Anfragen von Betreibern, Sperrung und Wiederherstellung wegen Abrechnung, Bereitstellung und Neuverschlüsselung gespeicherter Geheimnisse. Die Organisation liest sie in ihrer Konsole. Plattformprotokolle sind ein eigener Export, den der Plattformbetreiber einrichtet: Betreiberereignisse (Bereich *Plattformbetreiber*) oder alle Ereignisse aller Organisationen (Bereich *Alle Ereignisse*).

Jedes kanonische Ereignis hat einen **Ereigniscode** und eine **Kategorie** aus dem Ereigniscode-Katalog, zum Beispiel `AUTHN-002` (`authentication`, fehlgeschlagene Anmeldung) oder `CONFIG-008` (`configuration-change`, SIEM-Ziel geändert). Codes sind stabil und werden nie neu nummeriert. Die zehn Kategorien sind `authentication`, `authorization`, `user-management`, `configuration-change`, `data-access`, `administrative`, `security`, `application-error`, `system` und `integration`.

Ergebnis (Filter in CERTYX)	Inhalt
Anmeldeereignisse (`authentication`)	Jeder Anmeldeschritt als eigenes Ereignis: `authentication.sign-in.succeeded` , `authentication.sign-in.failed` , `authentication.sign-out` , `authentication.locked` , `authentication.mfa.enrolled` , `authentication.mfa.confirmed` , `authentication.mfa.reset` , `authentication.mfa.recovery-codes-regenerated`
Geschäftliche Änderungen (`business`)	Änderungen außerhalb einer Benutzeranfrage, zum Beispiel Änderungen des Agent-Zustands (`agent.health.changed`), automatische Erneuerungsentscheidungen, Ereignisse des Hintergrundinventars und der Ausführungswarteschlange
Erfolge (`success`)	Abgeschlossene API-, Konsolen-, Plattform- und Agent-Anfragen mit den geschäftlichen Änderungen, die sie vorgenommen haben (vorher/nachher)
Empfangene Anfragen (`attempt`)	Der Eintrag, der beim Eingang einer Anfrage vor ihrer Ausführung geschrieben wird, sodass auch eine nie abgeschlossene Anfrage eine Spur hinterlässt; kein fehlgeschlagener Versuch. Wird nur gesendet, wenn angekreuzt
Abgelehnte Anfragen (`rejected`)	Mit HTTP 4xx (außer 401/403) abgewiesene Anfragen, zum Beispiel ungültige Eingaben
Zugriff verweigert (`denied`)	Mit HTTP 401 oder 403 abgewiesene Anfragen
Serverfehler (`error`)	Mit HTTP 5xx fehlgeschlagene Anfragen (Ausnahmetyp und Trace-ID, niemals ein Stacktrace)

Das kanonische `outcome` normalisiert diese Werte zu `success` , `failure` oder `unknown` ; das Journalergebnis bleibt als `outcome_original` verfügbar.

Anmeldeereignisse. Sie umfassen die Passwortanmeldung, das Passwort mit MFA-Code oder Wiederherstellungscode, die Firmenanmeldung mit OIDC oder SAML sowie die mobile CERTYX-App. Abmeldungen umfassen Sitzungen, die durch Ablauf, Inaktivität, eine Passwort-, E-Mail- oder MFA-Änderung oder eine vom Identitätsanbieter angeforderte Abmeldung beendet wurden. Jedes Ereignis enthält die E-Mail-Adresse des Benutzers, ob diese Identität verifiziert wurde, die Organisation, die Quell-IP-Adresse, den User-Agent, die Methode, das Ergebnis (`success` oder `failure`) und einen festen Grundcode. Das angegebene Passwort, der MFA-Code, der Wiederherstellungscode oder das Token wird niemals aufgezeichnet.

Feld	Werte
Methode	`password` , `password+mfa` , `password+recovery-code` , `oidc` , `saml` , `mobile-password` (auch `+mfa` , `+recovery-code`), `mobile-oidc` , `mobile-saml` , `mobile-sso` , `totp` (MFA-Änderungen)
Ergebnis	`success` , `failure`
Grund (Fehlschläge)	`unknown-account` , `invalid-credentials` , `disabled` , `locked` , `too-many-failures` , `mfa-required` , `mfa-invalid` , `mfa-enrollment-required` , `enrollment-expired` , `sso-required` , `not-operator` , `operator-entry` , `organization-unavailable` , `organization-mismatch` , `account-not-allowed` , `ticket-invalid` , `device-refused` , `invalid-request` , `sso-error` , `sso-disabled` , `sso-no-subject` , `sso-group-overage` , `sso-not-provisioned` , `sso-no-role` , `sso-expired` , `sso-not-configured` , `sso-unverified-domain`
Grund (Abmeldungen)	`user` , `credentials-changed` , `idp-logout` , `session-ended` , `administrator-reset`

Eine fehlgeschlagene Anmeldung, deren E-Mail-Adresse zu keinem Konto passt, wird der Organisation zugeordnet, deren verifizierte Domäne für die Firmenanmeldung zur Adresse passt, andernfalls der Plattform; in diesem Fall erhält Ihre Organisation sie nicht.

Geschäftliche Änderungen. Konfigurations- und Inventaränderungen enthalten die Aktion, die Ressource und die Werte vor und nach der Änderung. Eine Änderung, die die Datensätze unverändert lässt, etwa wenn ein Agent dasselbe Zertifikat auf einem Endpunkt erneut sieht (`endpoint.observe`), enthält die Aktion und die Ressource mit leeren Vorher- und Nachher-Werten; eine Vertrauensgraph-Revision, die nur Zeitpunkte der letzten Sichtung fortschreibt, wird nicht als `trust-graph.updated` erfasst. Eine Änderung des Agent-Zustands (`agent.health.changed` , Akteur `system:agent-health`) enthält den Agent, die vorherige und die neue Ursache und Schwere, die Erläuterung und die Schritte zur Behebung. Geheimnisse, private Schlüssel und Agent-Ausgaben sind niemals Teil eines Ereignisses.

Zwei Datensätze pro Anfrage. Jede Anfrage wird zweimal im Journal erfasst: ein `attempt` -Datensatz vor ihrer Ausführung (Code `SYSTEM-000` , sodass auch eine nie abgeschlossene Anfrage eine Spur hinterlässt) und danach ein Ergebnisdatensatz (`success` , `rejected` , `denied` oder `error`), beide mit derselben Korrelations-ID (`correlation_id` , im JSON-Datensatz `TraceId`). Ein Ziel sendet den `attempt` -Datensatz nur, wenn unter *Nur senden Empfangene Anfragen* angekreuzt ist: Ohne angekreuztes Ergebnis

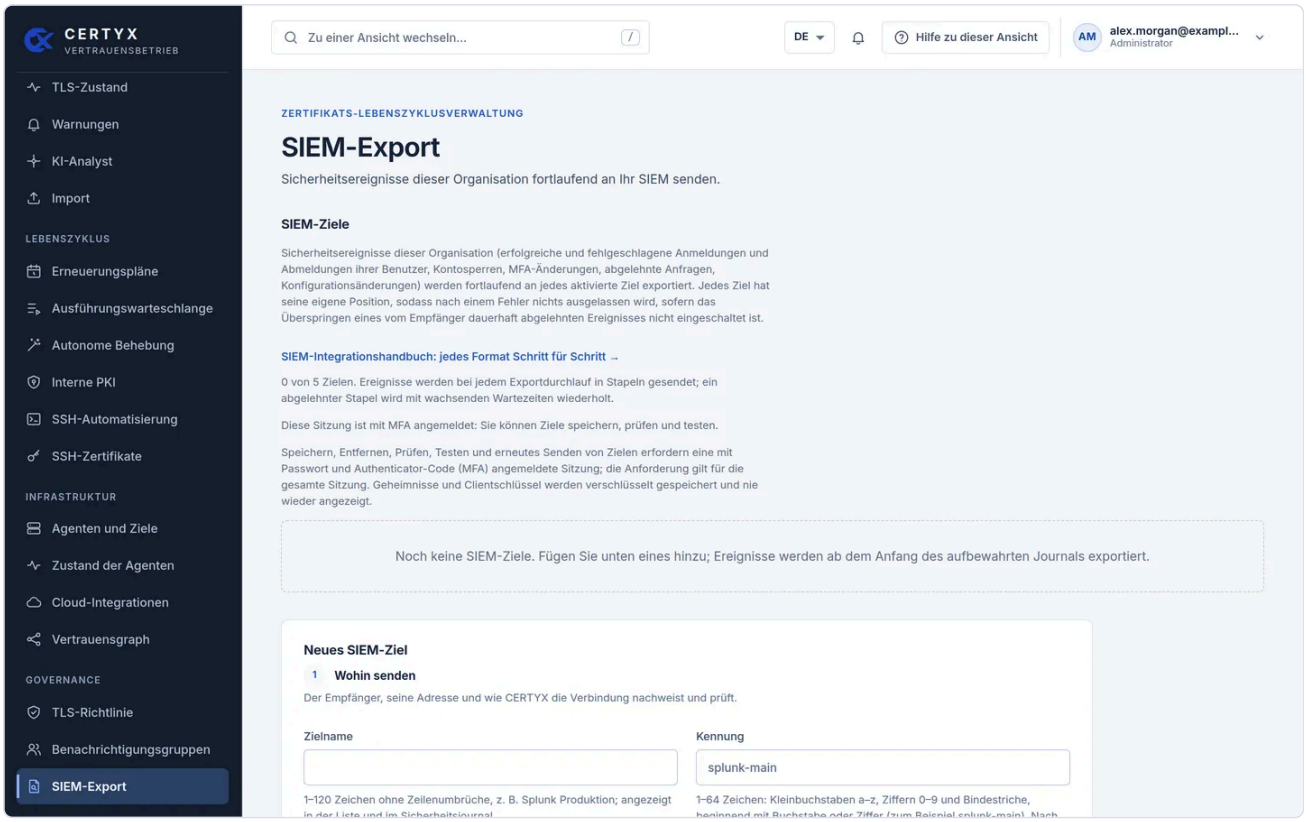
erhält es jedes Ergebnis außer `attempt`, also einen Datensatz pro Anfrage. Kreuzen Sie *Empfangene Anfragen* zusammen mit den anderen Ergebnissen an, wenn das SIEM nie abgeschlossene Anfragen sehen muss; der Empfangsdatensatz bleibt im Journal und in einem Dateiarchiv erhalten.

Zustandsprüfungen. Anfragen an `/health/ready` und `/health/live` (Bereitschaftsprüfungen von Load Balancer und Container) werden im Journal erfasst – `/health/live` gar nicht, `/health/ready` mit beiden Datensätzen, `Tenant` `unattributed` –, aber nicht an Netzwerkziele gesendet. Der Betreiber kann sie mit `Clm:Siem:ExportHealthProbes=true` senden lassen. Ein Dateiarchiv behält sie.

5 Bevor Sie beginnen

1. **Rolle und MFA.** Melden Sie sich als Admin der Organisation mit Ihrem Passwort und Authenticator-Code an (oder über die Firmenanmeldung, wenn der Identitätsanbieter MFA meldet). Speichern, Prüfen, Testen, erneutes Senden und Entfernen von Zielen erfordern eine MFA-Sitzung; die Ansicht teilt Ihnen genau mit, was zu tun ist, wenn die Sitzung diese Anforderung nicht erfüllt. Sitzungen der mobilen App können Ziele nicht ändern. Siehe Berechtigungen.
2. **Plattformdatenbank.** Der SIEM-Export liest das Journal aus der Plattformdatenbank. Führt eine Installation das Journal in einer lokalen Datei, zeigt die Ansicht an, dass der Export nicht verfügbar ist; der Betreiber muss die Plattformdatenbank konfigurieren.
3. **Netzwerkpfad.** Der CERTYX-Server (nicht Ihr Rechner) baut die Verbindung auf. Erlauben Sie ausgehendes TCP vom CERTYX-Server zum Port des Empfängers und stellen Sie sicher, dass der Hostname des Empfängers in dem DNS auflösbar ist, das der CERTYX-Server verwendet.
4. **Empfängerzertifikat.** Der Empfänger benötigt ein TLS-Serverzertifikat, dessen Namen den Hostnamen enthalten, den Sie in CERTYX eingeben. Wird es von einer privaten oder selbstsignierten CA ausgestellt, halten Sie das Stammzertifikat dieser CA (PEM) für *Vertrauenswürdige Stammzertifikate* bereit. Verlangt der Empfänger eine Clientauthentifizierung, bereiten Sie ein Clientzertifikat und seinen unverschlüsselten privaten PEM-Schlüssel vor (siehe Clientzertifikat (mTLS)).
5. **Grenzen.** Jede Organisation kann bis zu 5 Ziele festlegen (Betreibereinstellung `Clm:Siem:MaxDestinationsPerOrganization`). Der Betreiber einer gemeinsam genutzten Installation kann Ziele von Organisationen auf öffentliche IP-Adressen beschränken (`Clm:Siem:AllowPrivateDestinations`); standardmäßig sind interne Empfänger erlaubt.

6 Ein Ziel in CERTYX hinzufügen



SIEM-Export: das Formular für ein neues Ziel und die Ziele der Organisation (Beispieldaten)

1. Öffnen Sie **SIEM-Export** (Plattformbetreiber: **Journal der Sicherheitsereignisse** → **SIEM-Export**, dann den Bereich wählen).
2. Geben Sie **Zielname** (1–120 Zeichen, angezeigt in der Liste und im Journal) und **Kennung** (1–64 Kleinbuchstaben, Ziffern und Bindestriche, zum Beispiel `splunk-main`; nach dem Speichern nicht änderbar) ein.
3. Wählen Sie **Transport** und dann **Format** und **Schemaprofil** aus den dafür angebotenen Optionen (Zulässige Kombinationen). Der Kasten *Was die Empfangsseite braucht* fasst die Einstellungen des Empfängers zusammen; der Link neben jedem Feld öffnet den passenden Abschnitt dieses Handbuchs.
4. Füllen Sie die Felder des Transports aus: HTTPS-Adresse und Geheimnis (Webhook, Splunk HEC, Sentinel), Syslog-Host, Port und Framing (Syslog) oder Aufbewahrung und Komprimierung (Dateiarchiv). Für `leef-2.0` optional das **LEEF-Trennzeichen**.
5. Optional: Fügen Sie **Vertrauenswürdige Stammzertifikate** (TLS und vertrauenswürdige Stammzertifikate) und ein **Clientzertifikat** mit seinem privaten Schlüssel (mTLS) ein und wählen Sie die **Sperrprüfung** (Sperrprüfung).

Das Formular hat vier nummerierte Abschnitte: **1 Wohin senden** (Schritte 2–5), **2 Was senden** (Schritt 6), **3 Form und Größe jedes Ereignisses** (Schritt 6a) und **4 Zustellung** (Schritte 7–8). Erweiterte Teile sind eingeklappt, bis Sie sie öffnen.

1. Optional: **Nur senden** – Ergebnisse, Mindestschweregrad, **Nur senden (Kategorien)** und **Nur senden (Ereigniscodes)**; unter **Weitere Filter** die zu sendenden oder nie zu sendenden Routen, Akteure, Kategorien und Ereigniscodes, die nie gesendet werden (Nur senden).

6a. Optional: **Ereignisgestaltung** – eine Vorgabe-Karte (**Vollständig**, **Ausgewogen**, **Sparsam**, **Benutzerdefiniert**) und ihre **Detaileinstellungen** (Ereignisgestaltung); **Feldzuordnung** – Felder umbenennen, entfernen, behalten oder hinzufügen (Feldzuordnung); **Vorschau des Testereignisses** zeigt, was das Ziel senden wird und wie viele Byte die Gestaltung spart. Für OpenTelemetry: **Inhalt (Body) des Log-Eintrags**, **Name des Authentifizierungs-Headers** und **Anfragen mit gzip komprimieren** (OpenTelemetry Logs).

1. Optional: Zustellwerte – **Versuche bis zur kritischen Warnung**, **Erste Wartezeit (Sekunden)**, **Längste Wartezeit (Sekunden)**, **Stapelgröße (Ereignisse)**, **Rückstandswarnung ab (Ereignisse)** und **Ein Ereignis überspringen, das der Empfänger nach den Versuchen oben weiter ablehnt** (Zustellung). Leere Felder verwenden die Serverstandards.
2. Lassen Sie **An dieses Ziel exportieren** ausgewählt (heben Sie die Auswahl auf, um ein Ziel zu pausieren, ohne seine Position zu verlieren) und wählen Sie **Ziel speichern**.
3. Wählen Sie **Verbindung prüfen** und anschließend **Testereignis senden** (Verbindungsprüfung und Testereignis).

Ein neues Ziel beginnt am Anfang des aufbewahrten Journals, sodass der Empfänger zuerst den Verlauf und danach neue Ereignisse erhält, sobald sie eintreten. Das Entfernen eines Ziels löscht seine Position; ein erneutes Hinzufügen beginnt wieder am Anfang. Um einen Zeitraum erneut zu senden, verwenden Sie **Erneut senden ab** (Erneut senden).

Geben Sie das Geheimnis erneut ein, wenn Sie das Format ändern. *Bearbeiten* lädt ein Ziel in das Formular. Ein leer gelassenes Geheimnisfeld behält das gespeicherte Geheimnis; eine Änderung des Transports erfordert die erneute Eingabe des Geheimnisses. Geheimnisse (HMAC-Signaturgeheimnis, HEC-Token, geheimer Clientschlüssel) und der private Schlüssel des Clientzertifikats werden verschlüsselt gespeichert, nie wieder angezeigt und nie in das Journal geschrieben. Jede Änderung wird als `siem.destination.*` (Code `CONFIG-008`) im Journal erfasst.

7 TLS und vertrauenswürdige Stammzertifikate

CERTYX prüft das Zertifikat des Empfängers bei jeder Verbindung und für jeden Transport: Das Zertifikat muss innerhalb seiner Gültigkeitsdauer liegen, den eingegebenen Host nennen (DNS-Name oder IP-Adresse in den alternativen Antragstellernamen), zu einem vertrauenswürdigen Stammzertifikat führen und die Sperrprüfung bestehen.

- **Ohne vertrauenswürdige Stammzertifikate** entscheidet der Zertifikatspeicher des Betriebssystems des CERTYX-Servers. Verwenden Sie dies für Empfänger mit einem Zertifikat einer öffentlichen CA (zum Beispiel Splunk Cloud oder Azure).
- **Mit vertrauenswürdigen Stammzertifikaten** wird genau den eingefügten Stammzertifikaten vertraut (eigener Vertrauensspeicher): Fügen Sie einen oder mehrere PEM-Blöcke (`-----BEGIN CERTIFICATE-----`) ein, höchstens 64 KiB, ohne private Schlüssel. Verwenden Sie dies für eine interne CA oder ein selbstsigniertes Empfängerzertifikat.

Die Prüfung kann nicht abgeschaltet werden.

So lesen Sie die Zertifikatskette eines Empfängers von einem Rechner im selben Netzwerk aus:

```
openssl s_client -connect siem.example.internal:6514 -servername siem.example.internal -showcerts </dev/null
```

Das zuletzt ausgegebene Zertifikat ist in der Regel die ausstellende CA; deren Stammzertifikat ist das einzufügende Zertifikat.

7.1 Clientzertifikat (mTLS)

Empfänger, die ihre Absender per Zertifikat authentifizieren (Cribl *Authenticate client*, Logstash `ssl_client_authentication`, syslog-ng `peer-verify(required-trusted)`, ein Reverse Proxy vor einem Webhook), können verlangen, dass CERTYX ein Clientzertifikat vorlegt. Es ist für `syslog-tls` und jeden HTTPS-Transport (`https`, `splunk-hec`, `azure-monitor`) verfügbar; für Dateiarchive gilt es nicht.

1. Stellen Sie für CERTYX ein Clientzertifikat von der CA aus, der der Empfänger vertraut (Schlüsselverwendung *digital signature*, erweiterte Schlüsselverwendung *client authentication*).
2. Fügen Sie im Zielformular unter **Clientzertifikat** ein PEM-Zertifikat (`-----BEGIN CERTIFICATE-----`) ein, höchstens 64 KiB.
3. Fügen Sie seinen **privaten Schlüssel** ein: einen unverschlüsselten PEM-Schlüssel (`-----BEGIN PRIVATE KEY-----`, RSA oder EC), höchstens 64 KiB. Der Schlüssel kann nur geschrieben werden: Er wird mit dem Data-Protection-Schlüsselbund des Servers verschlüsselt gespeichert, nie wieder angezeigt und nie im Journal erfasst. Lassen Sie das Feld beim Bearbeiten leer, um den gespeicherten Schlüssel zu behalten.
4. **Verbindung prüfen** legt das Zertifikat in der TLS-Stufe vor.

Ein Zertifikat ohne passenden Schlüssel wird beim Speichern abgelehnt. Kann der gespeicherte Schlüssel nicht mehr entschlüsselt werden (der Schlüsselbund hat sich geändert), schlägt die Zustellung mit *Das Clientzertifikat oder sein privater Schlüssel kann nicht gelesen werden; geben Sie beide erneut ein.* fehl.

Konfigurationsziele verwenden `ClientCertificatePem` und `ClientKey` (eine `secret:<name>`-Referenz); siehe `docs/CONFIGURATION.md`.

7.2 Sperrprüfung

CERTYX prüft den Sperrstatus jedes Zertifikats der Empfängerkette unterhalb des Stammzertifikats online, über die CRL-Verteilungspunkte oder den OCSP-Responder, die in den Zertifikaten genannt sind, und wartet höchstens 10 Sekunden pro Adresse.

Modus	Gesperrtes Zertifikat	Status unbekannt (keine CRL-/OCSP-Adresse, Responder nicht erreichbar)	Verwenden für
Soft-Fail (Standard)	abgelehnt	angenommen	Die meisten Empfänger, auch interne CAs ohne veröffentlichte CRL
Hard-Fail	abgelehnt	abgelehnt	Empfänger, deren CA eine CRL oder einen OCSP-Responder veröffentlicht, den der CERTYX-Server erreichen kann

Damit die Prüfung ein Ergebnis liefert, benötigt der CERTYX-Server ausgehenden HTTP-Zugriff auf die CRL- und OCSP-Adressen. Ein abgelehntes Zertifikat lässt die TLS-Stufe der Prüfung mit `tls-untrusted` (oder mit eingefügten Stammzertifikaten `tls-untrusted-roots`) fehlschlagen.

8 Signierter HTTPS-Webhook

CERTYX sendet `POST`-Anfragen an Ihre HTTPS-Adresse. Mit dem Format `json` ist der Inhalt ein JSON-Array von Journaldatensätzen (`Content-Type: application/json`); mit dem Format `ndjson` ist der Inhalt ein Dokument des gewählten Profils pro Zeile (`Content-Type: application/x-ndjson`, jede Zeile mit LF abgeschlossen). Eine Anfrage enthält bis zur Stapelgröße des Ziels (standardmäßig 200). Der Header `X-Certyx-Signature` enthält `sha256=`, gefolgt vom hexadezimalen HMAC-SHA256 in Kleinbuchstaben über die exakten Bytes des Inhalts, mit dem Signaturgeheimnis als Schlüssel. Jede Antwort mit HTTP 2xx nimmt den Stapel an. HTTP 400, 413 oder 422 wird als Ablehnung der Daten behandelt (Abgelehnte Ereignisse); jede andere Antwort, eine Zeitüberschreitung (standardmäßig 20 Sekunden) oder ein TLS-Fehler veranlasst CERTYX, dieselben Ereignisse nach der Backoff-Wartezeit erneut zu senden (Zustellung).

8.1 Empfängerseite

1. Stellen Sie einen HTTPS-Endpunkt bereit, zum Beispiel `https://siem.example.com/certyx`, mit einem Zertifikat für diesen Hostnamen.
2. Lesen Sie den unveränderten Anfrageinhalt vor dem Parsen, berechnen Sie den HMAC-SHA256 mit dem Signaturgeheimnis als Schlüssel (die Geheimniszeichenfolge UTF-8-kodiert und unverändert verwendet – dekodieren Sie den hexadezimalen Text nicht) und vergleichen Sie `sha256=<hex>` in konstanter Zeit mit `X-Certyx-Signature`. Weisen Sie eine Abweichung mit 401 ab.
3. Parsen Sie den Inhalt anhand seines Content-Type: ein JSON-Array (`application/json`) oder ein JSON-Dokument pro Zeile (`application/x-ndjson`).
4. Speichern Sie die Ereignisse und antworten Sie dann mit 2xx. Antworten Sie erst mit 2xx, wenn der Stapel sicher gespeichert ist.
5. Deduplizieren Sie nach der Ereignis-ID: `EventId` im JSON-Datensatz, `event_id` (native), `event.id` (ECS) oder `metadata.uid` (OCSF). Die Zustellung erfolgt mindestens einmal, daher kann ein Stapel nach einer Zeitüberschreitung zweimal eintreffen.
6. Nehmen Sie das Testereignis an: Seine Ereignis-ID ist eine UUID; JSON-Datensatz: `Sequence` 0, `Actor` `siem-test`, `Method` `INTERNAL`, `Route` `business/siem.test`; kanonisch: `event_code` `INTEGRATION-001`, `category` `integration`, `action` `siem.test`, `user_id` `siem-test`.

8.2 Signatur prüfen

Python (Standardbibliothek):

```
import hashlib, hmac, json, os
from http.server import BaseHTTPRequestHandler, HTTPServer

SECRET = os.environ["CERTYX_SIGNING_SECRET"].encode("utf-8")

def events(body, content_type):
    if content_type.startswith("application/x-ndjson"):
        return [json.loads(line) for line in body.decode("utf-8").splitlines() if line]
    return json.loads(body)
```

```
class CertyxReceiver(BaseHTTPRequestHandler):
    def do_POST(self):
        body = self.rfile.read(int(self.headers.get("Content-Length", "0")))
        expected = "sha256=" + hmac.new(SECRET, body, hashlib.sha256).hexdigest()
        if not hmac.compare_digest(expected, self.headers.get("X-Certyx-Signature", "")):
            self.send_response(401)
            self.end_headers()
            return
        for event in events(body, self.headers.get("Content-Type", "")):
            store(event)  # your storage; deduplicate by the event ID
        self.send_response(204)
        self.end_headers()

# Run behind a TLS-terminating reverse proxy, or wrap the socket with ssl.SSLContext.
HTTPServer(("0.0.0.0", 8080), CertyxReceiver).serve_forever()
```

Node.js (Express):

```
import crypto from 'node:crypto';
import express from 'express';

const secret = process.env.CERTYX_SIGNING_SECRET;
const app = express();

app.post('/certyx', express.raw({ type: ['application/json', 'application/x-ndjson'], limit: '20mb' }), (req, res) => {
  const expected = 'sha256=' + crypto.createHmac('sha256', secret).update(req.body).digest('hex');
  const given = req.get('X-Certyx-Signature') ?? '';
  const valid = given.length === expected.length && crypto.timingSafeEqual(Buffer.from(given), Buffer.from(expected));
  if (!valid) return res.sendStatus(401);
  const text = req.body.toString('utf8');
  const events = req.is('application/x-ndjson')
    ? text.split('\n').filter(Boolean).map(line => JSON.parse(line))
    : JSON.parse(text);
  for (const event of events) store(event); // deduplicate by the event ID
  res.sendStatus(204);
});
```

C# (ASP.NET Core Minimal API):

```
app.MapPost("/certyx", async (HttpRequest request) =>
{
    using var buffer = new MemoryStream();
    await request.Body.CopyToAsync(buffer);
    var body = buffer.ToArray();
    var key = Encoding.UTF8.GetBytes(Environment.GetEnvironmentVariable("CERTYX_SIGNING_SECRET")!);
    var expected = "sha256=" + Convert.ToHexString(HMACSHA256.HashData(key, body)).ToLowerInvariant();
    var given = request.Headers["X-Certyx-Signature"].ToString();
    if (!CryptographicOperations.FixedTimeEquals(Encoding.ASCII.GetBytes(expected), Encoding.ASCII.GetBytes(given)))
        return Results.Unauthorized();
    var text = Encoding.UTF8.GetString(body);
    var documents = request.ContentType?.StartsWith("application/x-ndjson") == true
        ? text.Split('\n', StringSplitOptions.RemoveEmptyEntries).Select(line => JsonDocument.Parse(line).RootElement)
        : JsonDocument.Parse(text).RootElement.EnumerateArray();
    foreach (var e in documents)
        Store(e); // deduplicate by the event ID
```

```
    return Results.NoContent();
});
```

Prüfung eines mitgeschnittenen Inhalts auf der Kommandozeile (`body.json` Byte für Byte gespeichert):

```
printf 'sha256=%s\n' "$(openssl dgst -sha256 -hmac "$CERTYX_SIGNING_SECRET" -r body.json | cut -d' ' -f1)"
```

8.3 CERTYX-Seite

1. Transport **Signierter HTTPS-Webhook** (`https`); Format `ndjson` mit dem Profil `native`, `ecs` oder `ocsf` oder Format `json` (der Datensatz der ersten Version).
2. **HTTPS-Adresse**: die absolute `https://`-URL des Endpunkts (höchstens 2048 Zeichen, ohne Benutzername oder Passwort), zum Beispiel `https://siem.example.com/certyx`.
3. **HMAC-Signaturgeheimnis**: Lassen Sie es bei einem neuen Webhook leer. CERTYX erzeugt dann ein Geheimnis mit 64 Zeichen und zeigt es nach dem Speichern einmal an – kopieren Sie es in den Empfänger, bevor Sie den Hinweis schließen. Sie können auch ein eigenes Geheimnis eingeben (höchstens 1024 Zeichen).
4. **Neues Geheimnis** auf der Zielkarte ersetzt das Geheimnis sofort und zeigt das neue einmal an; das bisherige Geheimnis ist ab der nächsten Zustellung ungültig, aktualisieren Sie daher gleichzeitig den Empfänger.
5. Optional: ein Clientzertifikat, wenn der Endpunkt mTLS verlangt.
6. Speichern, **Verbindung prüfen**, **Testereignis senden**.

So sieht Erfolg aus: Die Prüfung zeigt DNS, TCP und TLS als erfolgreich; das Testereignis zeigt, dass das Ziel es angenommen hat, und nennt seine Ereignis-ID; der Empfänger hat ein Ereignis mit dieser ID protokolliert. Innerhalb des nächsten Durchlaufs zeigt die Karte *letzter Erfolg* und die Anzahl der zugestellten Ereignisse.

9 Splunk HTTP Event Collector

Der empfohlene Weg zu Splunk: Jeder Stapel wird von Splunk bestätigt, und mit Indexer Acknowledgement wartet CERTYX, bis die Ereignisse indiziert sind.

9.1 Den Splunk-Weg wählen

Ziel	Transport × Format × Profil	Sourcetype (Standard)
Kanonische Ereignisse, ohne Parsen nach `event_id`, `event_code`, `category` durchsuchbar	`splunk-hec` × `ndjson` × `native`	`certyx:event`
Splunk-CIM-Datenmodelle (Authentication, Change)	`splunk-hec` × `ndjson` × `splunk-cim`	`certyx:event`
ECS- oder OCSF-Dokumente in Splunk	`splunk-hec` × `ndjson` × `ecs` / `ocsf`	`certyx:event`
Bestehende Suchen auf dem Datensatz der ersten Version	`splunk-hec` × `json` × `native`	`certyx:security`
Kein HEC verfügbar, Syslog in einen Indexer	`syslog-tls`, Zeilenumbruch-Framing (Splunk-TCP-SSL-Eingang)	vom Eingang gesetzt

Verwenden Sie pro Profil einen eigenen Sourcetype (zum Beispiel `certyx:event`, `certyx:event:cim`, `certyx:event:ecs`), wenn mehrere CERTYX-Ziele in denselben Index schreiben, damit `props.conf` jeden einzeln behandeln kann.

9.2 Empfängerseite (Splunk)

1. **HEC aktivieren.** *Settings* → *Data inputs* → *HTTP Event Collector* → *Global Settings*: *All Tokens* **Enabled**, **Enable SSL** ausgewählt, *HTTP Port Number* 8088 (Standard). Speichern.
2. **Index anlegen** (falls nötig): *Settings* → *Indexes* → *New Index*, zum Beispiel `clm`.

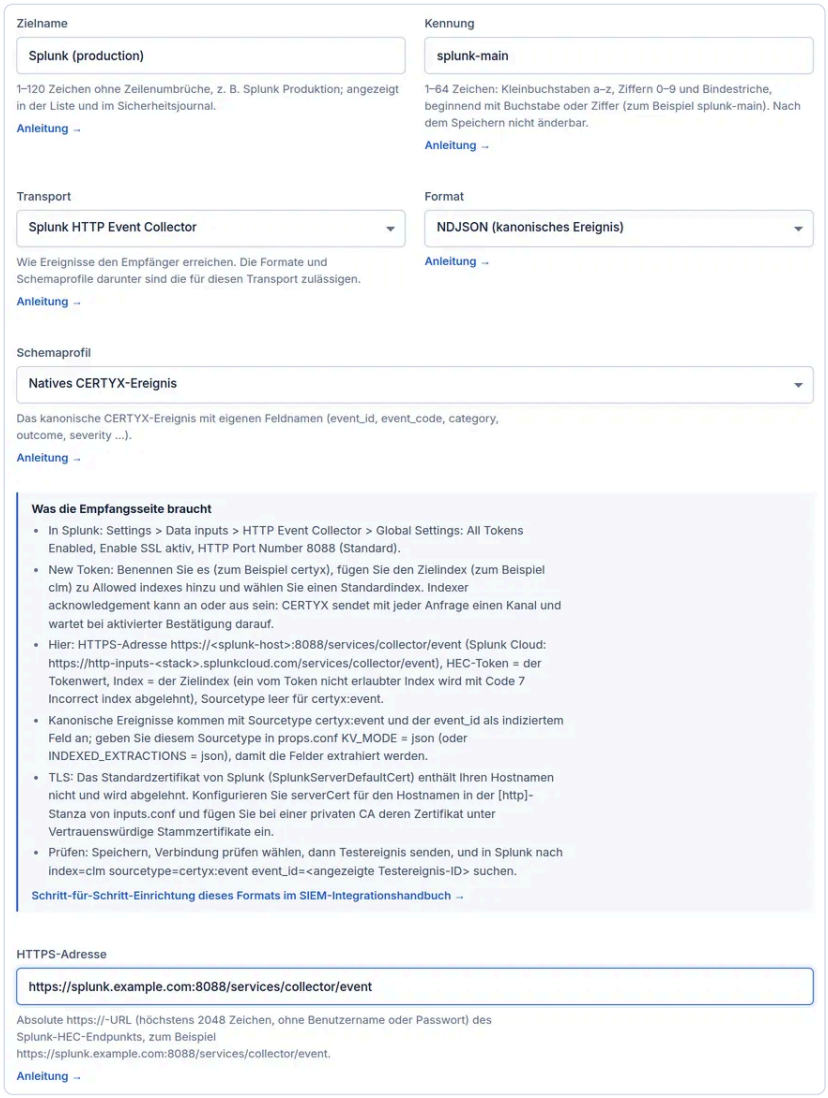
3. **Token anlegen.** *HTTP Event Collector → New Token*: Name `certyx`; *Source name override* leer lassen; *Input Settings*: optional ein Standard-*Source type* (CERTYX setzt den Sourcetype ohnehin bei jedem Ereignis), *Allowed Indexes* enthält `clm`, *Default Index* `clm`. *Enable indexer acknowledgement* kann an oder aus sein – CERTYX unterstützt beides. Kopieren Sie den **Token Value**.
4. **Serverzertifikat.** Das Standardzertifikat von Splunk (`SplunkServerDefaultCert`) enthält Ihren Hostnamen nicht und wird von CERTYX abgelehnt. Konfigurieren Sie ein Zertifikat für den Hostnamen in der `[http]`-Stanza von `inputs.conf` (zum Beispiel `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/inputs.conf`) und starten Sie Splunk neu:

```
[http]
disabled = 0
enableSSL = 1
port = 8088
serverCert = $SPLUNK_HOME/etc/auth/certyx/splunk-hec.pem
sslPassword = <password of the private key in the PEM file>
```

1. **Splunk Cloud.** Verwenden Sie `https://http-inputs-<stack>.splunkcloud.com/services/collector/event` (Port 443). Dessen Zertifikat stammt von einer öffentlichen CA; vertrauenswürdige Stammzertifikate sind nicht nötig.
2. **Firewall.** Erlauben Sie TCP 8088 (Splunk Cloud: 443) vom CERTYX-Server.
3. **Sourcetype-Einstellungen** wie unter props.conf.

Von einem Rechner im selben Netzwerk können Sie bestätigen, dass HEC antwortet: `curl https://splunk.example.com:8088/services/collector/health` liefert `{"text":"HEC is healthy","code":17}`.

9.3 CERTYX-Seite



Ein Ziel für Splunk HTTP Event Collector wird angelegt: Name, Kennung, Transport und Collector-Adresse (Beispieldaten)

1. Transport **Splunk HTTP Event Collector** (`splunk-hec`), Format und Profil wie oben gewählt.
2. **HTTPS-Adresse**: `https://<splunk-host>:8088/services/collector/event` – der Endpunkt `/event` , nicht `/raw` .
3. **HEC-Token**: der Token Value aus Splunk (`<HEC token>`).
4. **Index (optional)**: der Zielindex, zum Beispiel `clm` . Er muss zu den erlaubten Indizes des Tokens gehören (andernfalls lehnt Splunk mit Code 7 *Incorrect index* ab); leer verwendet den Standardindex des Tokens.
5. **Sourcetype (optional)**: leer verwendet `certyx:event` für `ndjson` (Betreibereinstellung `Clm:Siem:SplunkEventSourceType`) und `certyx:security` für `json` (`Clm:Siem:SplunkSourceType`).
6. **Vertrauenswürdige Stammzertifikate**: Fügen Sie die CA ein, die das HEC-Zertifikat ausgestellt hat, wenn es sich um eine private CA handelt.
7. Speichern, **Verbindung prüfen**, **Testereignis senden**.

Was CERTYX sendet. `POST` mit `Authorization: Splunk <HEC token>` , `X-Splunk-Request-Channel: <GUID>` (stabil pro Ziel; von Tokens mit Indexer Acknowledgement verlangt, von anderen ignoriert) und einem Umschlag pro Ereignis, aneinandergereiht. Jeder Umschlag enthält:

- `time` – die Ereigniszeit, Epochensekunden mit Millisekunden;
- `host` – die CERTYX-Instanz, die das Ereignis erfasst hat;
- `source` – `certyx:security-journal` ;
- `sourcetype` – wie konfiguriert; `index` – nur, wenn gesetzt;

- `event` – Format `json`: der JSON-Datensatz (JSON-Datensatz); Format `ndjson`: das kanonische Ereignis (`native`, `splunk-cim`), das ECS-Dokument (`ecs`) oder das OCSF-Dokument (`ocsf`);
- `fields` (indizierte Felder) – Format `json`: `event_id`; Format `ndjson`: immer `event_id`, `event_code` und `category`, mit `splunk-cim` zusätzlich die CIM-Felder (CIM).

Enthält die Antwort eine `ackId`, fragt CERTYX jede Sekunde `/services/collector/ack` ab, bis Splunk die Indizierung bestätigt, höchstens 30 Sekunden lang (Betreibereinstellungen `Clm:Siem:HecAckPollInterval`, `Clm:Siem:HecAckTimeout`); ein unbestätigter Stapel wird erneut gesendet. Die HEC-Codes 5, 6, 12, 13 und 15 werden als Ablehnung der Daten behandelt (Abgelehnte Ereignisse). Beispiele: Beispielereignisse.

So sieht Erfolg aus: *Splunk HEC hat das Testereignis angenommen (Code 0).* mit Index, Sourcetype und Ereignis-ID. Suchen Sie dann:

```
index=clm event_id=<event ID shown by the test>
index=clm sourcetype=certyx:event category=authentication outcome=failure
| stats count by username, src_ip, reason
index=clm sourcetype=certyx:security Outcome=authentication SignIn.Result=failure
| stats count by SignIn.User, SourceIp, SignIn.Reason
```

9.4 props.conf

Ereignisse vom Endpunkt `/event` übernehmen ihre Zeit aus dem `time` des Umschlags; `TIME_PREFIX`, `TIME_FORMAT` und `TIMESTAMP_FIELDS` werden nicht benötigt. Lange Ereignisse (viele Änderungen, nationale Zeichen) benötigen ein größeres `TRUNCATE` als den Splunk-Standard von 10.000 Bytes. Auf den Search Heads und Indexern (oder in Splunk Cloud über eine eigene App):

```
[certyx:event]
KV_MODE = json
TRUNCATE = 0
SHOULD_LINEMERGE = false

[certyx:security]
KV_MODE = json
TRUNCATE = 0
SHOULD_LINEMERGE = false
```

`TRUNCATE = 0` hebt die Grenze auf; setzen Sie einen großen Wert (zum Beispiel `200000`), wenn Ihre Richtlinie eine Grenze verlangt. Indizierte Felder (`event_id`, `event_code`, `category`) lassen sich auch mit der schnellen Syntax `field::value` durchsuchen, zum Beispiel `event_code::AUTHN-002`. Ein Feld, das sowohl indiziert als auch im JSON-Ereignis vorhanden ist, kann seinen Wert doppelt zeigen; verwenden Sie `| eval event_id=mvindex(event_id,0)` oder durchsuchen Sie die indizierte Form.

9.5 Splunk CIM

Mit dem Profil `splunk-cim` schreibt CERTYX die CIM-Felder jedes Ereignisses als indizierte Felder: die gemeinsamen Felder `vendor_product` (`CERTYX CLM`), `app` (`clm`), `signature` (Meldung), `signature_id` (Ereigniscode), `severity` (informational, low, medium, high, critical), `severity_id`, `src`, `src_port`, `dest`, `user`, `user_id`, `session_id`; für das Datenmodell Authentication (Kategorie `authentication`) `action` (`success`, `failure`, `unknown`), `authentication_method`, `reason`; für das Datenmodell Change (Kategorien `user-management`, `configuration-change` sowie `administrative` mit einer Änderungsart) `action` (`created`, `modified`, `deleted`), `change_type` (`AAA` für die Benutzerverwaltung, sonst `application`), `object`, `object_category`, `object_id`, `command`, `status`, `result`; für andere Ereignisse `action` = die kanonische Aktion.

Das kanonische Ereignis in `event` verwendet einige derselben Namen (`action`, `severity`, `reason`, `user_id`). Verhindern Sie bei einem CIM-Sourcetype, dass die JSON-Extraktion zur Suchzeit die CIM-Werte überschreibt, und lesen Sie kanonische Felder bei Bedarf mit `spath`:

```
[certyx:event:cim]
KV_MODE = none
```

```
TRUNCATE = 0
SHOULD_LINEMERGE = false
```

CERTYX liefert kein Splunk-Add-on: Die Event Types und Tags, die Ereignisse den CIM-Datenmodellen zuordnen, sind nicht enthalten. Legen Sie sie in Ihrer eigenen App an, zum Beispiel:

```
# eventtypes.conf
[certyx_authentication]
search = sourcetype=certyx:event:cim category::authentication

[certyx_change]
search = sourcetype=certyx:event:cim (category::user-management OR category::configuration-change OR category::administrative)

# tags.conf
[eventtype=certyx_authentication]
authentication = enabled

[eventtype=certyx_change]
change = enabled
```

Prüfen Sie anschließend mit `| datamodel Authentication search | search sourcetype=certyx:event:cim`.

10 Syslog über TLS

Alle Syslog-Formate teilen sich einen Transport. CERTYX öffnet eine TLS-Verbindung zu Host und Port (und legt dabei das Clientzertifikat vor, wenn eines festgelegt ist), schreibt den Stapel und schließt die Verbindung. Syslog hat keine Zustellbestätigung: Ein abgeschlossener TLS-Schreibvorgang gilt als angenommen.

Kanonische Formate (`rfc5424`, `cef`, `leef-1.0`, `leef-2.0`, `kv`) sind RFC-5424-Nachrichten mit dem unten beschriebenen Header; die MSG ist die Zeile des Formats. **Formate der ersten Version** (`json`, `cef-compat`, `leef-compat`) behalten ihren Header-Aufbau: `<86>` (informational) oder `<84>` (warning), APP-NAME `clm`, MSGID die Ereigniskategorie (zum Beispiel `authentication`; vor 1.2.19 `security-journal`), keine strukturierten Daten. Jedes Syslog-Ziel kann in der Ereignisgestaltung seine eigene MSGID (Kategorie, Ereigniscode, eigener Text wie `security-journal` oder keine) und APP-NAME wählen; ein Empfänger filtert dann nach Kategorie, ohne die Nachricht zu lesen, zum Beispiel `msgid=authentication` in Splunk.

10.1 RFC-5424-Header der kanonischen Formate

`<PRI>1 TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA MSG`

- **PRI** = Facility × 8 + Syslog-Schweregrad. Facility aus `Clm:Siem:Syslog:Facility` (Standard 10, authpriv; 16, local0, ist die übliche Alternative). Schweregrad aus dem kanonischen Schweregrad: 0–3 → 6 informational, 4–5 → 5 notice, 6–7 → 4 warning, 8 → 3 error, 9 → 2 critical, 10 → 1 alert.
- **VERSION** `1`; **TIMESTAMP** UTC mit Millisekunden (`2026-10-18T08:15:02.417Z`).
- **HOSTNAME** die CERTYX-Instanz, die das Ereignis erfasst hat; **APP-NAME** `Clm:Siem:Syslog:AppName` (Standard `clm`); **PROCID** die Prozess-ID der CERTYX-Instanz; **MSGID** die Ereigniskategorie (zum Beispiel `authentication`) oder das, was die Ereignisgestaltung des Ziels wählt (Ereigniscode, eigener Text oder keine); das Ziel kann auch einen eigenen APP-NAME festlegen. Header-Werte sind druckbares US-ASCII; andere Zeichen werden zu `_`.
- **STRUCTURED-DATA** immer `[timeQuality tzKnown="1"]` und `[origin software="CERTYX CLM" swVersion="<version>" ip="<server address>"]` (`ip` nur, wenn bekannt). Konfiguriert der Betreiber eine IANA Private Enterprise Number (`Clm:Siem:Syslog:IanaPrivateEnterpriseNumber`), zusätzlich `[certyx@<PEN> eventId="…" eventCode="…" category="…" outcome="…" schemaVersion="…" organizationId="…"]` und `enterpriseId` in `origin`. CERTYX hat keine eigene registrierte Enterprise-Nummer; eine SD-ID mit `@` muss eine registrierte verwenden, daher wird das private Element ohne konfigurierte Nummer nicht gesendet. 32473 (die Dokumentationsnummer aus RFC 5612) wird abgelehnt. Werte maskieren `"`, `\` und `]` mit einem Backslash.

- **MSG** – `rfc5424`: das JSON-Dokument des Profils (`native`, `ecs` oder `ocsf`), vorangestellt das UTF-8-Byte-Order-Mark, sofern nicht `Clm:Siem:Syslog:ByteOrderMark=false`; `cef`, `leef-1.0`, `leef-2.0`, `kv`: diese Zeile, ohne Byte Order Mark.

Zeigt ein Empfänger oder `ï»¿` vor dem JSON an, bitten Sie den Betreiber, `Clm:Siem:Syslog:ByteOrderMark=false` zu setzen.

10.2 Framing

Das Framing teilt dem Empfänger mit, wo eine Nachricht endet; es muss zum empfangenden Eingang passen.

Framing in CERTYX	Auf der Leitung	Verwenden für
Oktettzählung (RFC 5425) – Standard	`<length> <message>`	Syslog-Quelle von Cribl Stream mit *Octet count framing* Yes, syslog-ng, rsyslog, IBM QRadar, ArcSight
Zeilenumbruch (RFC 6587, Splunk-TCP-Eingang)	`<message>`, gefolgt von einem Zeilenvorschub; Zeilenvorschübe innerhalb einer Nachricht werden als `\n` geschrieben	Einen Splunk-Eingang `[tcp-ssl:<port>]`, Logstash `tcp` mit dem Codec `line`, den TCP-Eingang des Elastic Agent

Bei der Oktettzählung ist `<length>` die Anzahl der UTF-8-Bytes der Nachricht (nicht der Zeichen), gefolgt von einem Leerzeichen; vor und nach einer Nachricht steht kein Zeilenvorschub. CERTYX schreibt jeden Datensatz – Längenpräfix und Nachricht zusammen – in einem einzigen Schreibvorgang, sodass ein Datensatz von bis zu 16 KiB in einem TLS-Record übertragen wird. Beide Seiten müssen dasselbe Framing verwenden: Ein Empfänger, der an Zeilenvorschüben trennt, während CERTYX Oktette zählt, zeigt die Länge des nächsten Datensatzes an das Ende des vorherigen angehängt (`… }412`), Datensätze, die mit der Länge beginnen (`412 <84>1 …`), und in der Mitte abgeschnittene Datensätze. Siehe Fehlerbehebung.

10.3 CERTYX-Seite (alle Syslog-Formate)

1. Transport **Syslog über TLS** (`syslog-tls`), Format nach Bedarf des Empfängers (`rfc5424`, `cef`, `leef-1.0`, `leef-2.0`, `kv` oder ein Format der ersten Version).
2. **Syslog-Host**: nur der DNS-Name oder die IP-Adresse – kein `https://`, kein `:port` (zum Beispiel `cribl-worker1.example.internal`). Der Name muss auf dem CERTYX-Server auflösbar sein und im Zertifikat des Empfängers genannt werden.
3. **Port**: der TLS-Port des empfangenden Eingangs (zum Beispiel 6514 oder 10514).
4. **Framing**: wie in der Tabelle oben.
5. **LEEF-Trennzeichen** (nur `leef-2.0`): ein Zeichen oder sein hexadezimaler Code (`x09`, `0x5E`); leer bedeutet TAB. Erlaubt sind TAB und druckbare ASCII-Satzzeichen außer `=`, dem senkrechten Strich und dem Backslash.
6. **Vertrauenswürdige Stammzertifikate**: die CA des Empfängerzertifikats, wenn es sich um eine private CA handelt; optional ein Clientzertifikat.
7. Speichern, **Verbindung prüfen**, **Testereignis senden**.

So sieht Erfolg aus: *Testereignis über TLS geschrieben (octet-counting).* mit der Ereignis-ID. Syslog hat keine Zustellbestätigung, daher belegt dies DNS, TCP, TLS und das Schreiben. Bestätigen Sie beim Empfänger, dass das Ereignis mit dieser ID im erwarteten Index oder in der erwarteten Tabelle angekommen ist.

10.4 Allgemeiner Syslog-Empfänger (rsyslog, syslog-ng)

rsyslog mit einem TLS-Listener auf 6514 (Oktettzählung wird automatisch erkannt):

```
global(DefaultNetstreamDriver="gtls"
       DefaultNetstreamDriverCAFile="/etc/rsyslog.d/tls/ca.pem"
       DefaultNetstreamDriverCertFile="/etc/rsyslog.d/tls/siem.example.internal.pem"
       DefaultNetstreamDriverKeyFile="/etc/rsyslog.d/tls/siem.example.internal.key")
module(load="imtcp" StreamDriver.Name="gtls" StreamDriver.Mode="1" StreamDriver.AuthMode="anon")
input(type="imtcp" port="6514" ruleset="certyx")
ruleset(name="certyx") { action(type="omfile" file="/var/log/certyx/security.log") }
```

Um das CERTYX-Clientzertifikat zu verlangen, verwenden Sie `StreamDriver.AuthMode="x509/certvalid"` mit der ausstellenden CA in `DefaultNetstreamDriverCAFile`.

syslog-ng:

```
source s_certyx {
  syslog(transport("tls") port(6514)
         tls(key-file("/etc/syslog-ng/tls/siem.key") cert-file("/etc/syslog-ng/tls/siem.pem") peer-verify(optional-untrusted)));
};
destination d_certyx { file("/var/log/certyx/security.log"); };
log { source(s_certyx); destination(d_certyx); };
```

Mit einem CERTYX-Clientzertifikat verwenden Sie `peer-verify(required-trusted)` und `ca-dir()` oder `ca-file()` mit der ausstellenden CA. Die JSON-MSG kann mit dem JSON-Parser des Empfängers geparst werden (rsyslog `mmjsonparse` mit einer Konfiguration ohne Cookie, syslog-ng `json-parser()`); die strukturierten Daten stehen als SDATA-Felder des Empfängers zur Verfügung.

11 Cribl Stream

Cribl empfängt die Ereignisse, parst sie und leitet sie an Splunk oder ein anderes Ziel weiter. Cribl wählt keinen Splunk-Index von selbst; legen Sie ihn in der Route oder Pipeline fest.

11.1 Die Cribl-Quelle wählen

Cribl-Quelle	CERTYX-Transport × Format	Hinweise
Syslog (TCP mit TLS)	`syslog-tls` × `rfc5424` (Profil `native`, `ecs` oder `ocsf`) oder `cef`, `leef-2.0`, `kv`	Das Framing muss passen: CERTYX-Oktettzählung ↔ Cribl *Octet count framing* **Yes**
Splunk HEC	`splunk-hec` × `ndjson`	Cribl antwortet wie Splunk; CERTYX sendet an `https://<worker>:8088/services/collector/event`
HTTP (raw)	`https` × `ndjson`	Cribl prüft die HMAC-Signatur nicht; authentifizieren Sie CERTYX mit einem Clientzertifikat (mTLS)

11.2 Empfängerseite (Cribl-Syslog-Quelle)

1. **Zertifikat.** *Group Settings* → *Security* → *Certificates* → *Add certificate*: Laden Sie das Zertifikat (mit seiner Kette) und den privaten Schlüssel für den Hostnamen des Workers hoch, zum Beispiel `cribl-worker1.example.internal`. Der Name muss derjenige sein, den Sie in CERTYX als Syslog-Host eingeben.
2. **Eine Quelle pro CERTYX-Port.** *Data* → *Sources* → *Syslog* → *Add Source*: Input ID `certyx_tls`, Address `0.0.0.0`, *TCP Port* `10514`, *UDP Port* leer. Teilen Sie diesen Port nicht mit anderen Absendern, die ein anderes Framing oder kein TLS verwenden: Framing und TLS gelten für jede Verbindung der Quelle.
3. **TLS.** In den *TLS Settings (TCP Only)* der Quelle: **Enabled** Yes, *Certificate name* das Zertifikat aus Schritt 1. *Authenticate client (mutual auth)*: Yes, wenn CERTYX ein Clientzertifikat vorlegt (mTLS) – fügen Sie dann die ausstellende CA als *CA certificate* hinzu –, andernfalls No. Bei ausgeschaltetem TLS indiziert Cribl den TLS-Handshake von CERTYX als Ereignisse (binäre Daten, die mit `\x16\x03` beginnen und den Servernamen enthalten).
4. **Framing.** In den *Advanced Settings* der Quelle: **Octet count framing** Yes (CERTYX-Framing *Oktettzählung (RFC 5425)*). Alternativ belassen Sie No und wählen in CERTYX das Framing *Zeilenumbruch* – mischen Sie beides jedoch nie.
5. **Verarbeitung.** Eine Pipeline an der Quelle oder Route mit: - *Parser* (Typ JSON object) auf dem Feld `message` für `rfc5424` und das `json` der ersten Version; für CEF und LEEF der Parser Ihres Cribl-Packs; für `kv` ein *Parser* vom Typ *Key=Value pairs*; - *Eval* mit den Feldern `index` = `'clm'` und `sourcetype` = `'certyx:syslog'` (das Splunk-Ziel von Cribl verwendet diese Felder; ohne sie landen die Ereignisse im Standardindex des Ziels).
6. **Route.** *Routing* → *Data Routes* → *Add Route*: Filter `__inputId=='syslog:certyx_tls:tcp'`, Pipeline wie in Schritt 5, Output Ihr Splunk-Ziel. Platzieren Sie sie oberhalb von Auffangrouten.
7. **Committen und verteilen** Sie die Konfiguration an die Worker-Gruppe und erlauben Sie TCP 10514 vom CERTYX-Server zu den Workern.

11.3 Empfängerseite (Cribl-Splunk-HEC-Quelle)

1. *Data* → *Sources* → *Splunk HEC* → *Add Source*: Input ID `certyx_hec`, Port `8088`, HEC endpoint `/services/collector`.

2. *Auth tokens*: Fügen Sie ein Token (`<HEC token>`) und optional die Felder `index` und `sourcetype` hinzu, die seinen Ereignissen hinzugefügt werden.
3. *TLS Settings*: Enabled Yes, das Zertifikat des Workers; gegenseitige Authentifizierung wie bei der Syslog-Quelle.
4. In CERTYX: Transport `splunk-hec` , Format `ndjson` , HTTPS-Adresse `https://cribl-worker1.example.internal:8088/services/collector/event` , das Token und ein Index, sofern Ihre Cribl-Route ihn weitergibt.

11.4 CERTYX-Seite

Syslog-Host `cribl-worker1.example.internal` (oder der Name des Load Balancers vor den Workern, der im Zertifikat genannt sein muss), Port `10514` , Framing **Oktettzählung (RFC 5425)** (oder **Zeilenumbruch**, wenn die Cribl-Quelle *Octet count framing* No hat), vertrauenswürdige Stammzertifikate, wenn das Cribl-Zertifikat von einer privaten CA stammt. Wählen Sie dann **Verbindung prüfen** und **Testereignis senden** und verwenden Sie *Live Data* an der Quelle in Cribl, um das Ereignis mit der ID des Testereignisses zu sehen.

So sieht Erfolg aus: Das Testereignis erscheint im Live-Mitschnitt der Quelle in Cribl als ein Ereignis, dessen `message` mit dem JSON-Dokument (bei `rfc5424`) oder der CEF-/LEEF-/KV-Zeile beginnt – keine vorangestellte Länge, keine binären Daten – und mit `appname` `clm` und der Kategorie als `msgid` ; anschließend in Splunk unter dem Index, den die Route setzt. Ist es in Cribl, aber nicht im Index, prüfen Sie den Routenfilter und das Feld `index` . Fragmente oder binäre Ereignisse: siehe Empfänger zeigt Fragmente oder binäre Daten.

12 Splunk-TCP-SSL-Eingang

Verwenden Sie dies, wenn der Syslog-Strom ohne HEC direkt in einen Splunk-Indexer oder Heavy Forwarder gehen soll. Splunk trennt einen TCP-Strom an Zeilenumbrüchen, daher muss CERTYX das Framing **Zeilenumbruch** verwenden.

12.1 Empfängerseite (Splunk)

1. Legen Sie den Index an (zum Beispiel `clm`).
2. Installieren Sie ein Zertifikat für den Hostnamen und konfigurieren Sie den Eingang in `inputs.conf` (zum Beispiel `$SPLUNK_HOME/etc/system/local/inputs.conf`):

```
[tcp-ssl:6514]
index = clm
sourcetype = certyx:syslog
disabled = 0

[SSL]
serverCert = $SPLUNK_HOME/etc/auth/certyx/splunk-syslog.pem
sslPassword = <password of the private key in the PEM file>
requireClientCert = false
```

1. Zeilenumbruch und Zeitstempel in `props.conf` :

```
[certyx:syslog]
SHOULD_LINEMERGE = false
LINE_BREAKER = ([\r\n]+)
TIME_PREFIX = ^<\d+>1\s
TIME_FORMAT = %Y-%m-%dT%H:%M:%S.%3NZ
TZ = UTC
MAX_TIMESTAMP_LOOKAHEAD = 30
TRUNCATE = 0
```

1. Starten Sie Splunk neu und erlauben Sie TCP 6514 vom CERTYX-Server.

12.2 CERTYX-Seite

Transport `syslog-tls`, Format `rfc5424` (oder `json`, `cef`, `kv`), Syslog-Host = der Splunk-Hostname aus dessen Zertifikat, Port `6514`, Framing **Zeilenumbruch**, vertrauenswürdige Stammzertifikate für eine private CA. **Verbindung prüfen**, **Testereignis senden**, dann suchen (das JSON beginnt beim ersten `{` der Nachricht):

```
index=clm sourcetype=certyx:syslog "<event ID shown by the test>"
index=clm sourcetype=certyx:syslog
| rex "(?<payload>\{.*\})$"
| spath input=payload
| search category=authentication
```

13 IBM QRadar

QRadar empfängt LEEF über Syslog/TLS. Drei LEEF-Formate stehen zur Verfügung:

13.1 Das LEEF-Format wählen

Format	Event ID im Header	Attribute	Verwenden für
`leef-2.0` (empfohlen)	Ereigniscode	kanonische Felder, Trennzeichen standardmäßig TAB oder das konfigurierte	Neue QRadar-Integrationen
`leef-1.0`	Ereigniscode	kanonische Felder, TAB-getrennt	QRadar-Versionen oder DSMs, die LEEF 1.0 erwarten
`leef-compat`	`certyx:<outcome>` oder `certyx:<event name>`	Attribute der ersten Version plus `eventId`, Trennzeichen `^`	Bestehende QRadar-Inhalte, die auf der ersten Version aufbauen

Header:

```
LEEF:2.0|CERTYX|CLM|<version>|<event code>|<delimiter>|
LEEF:1.0|CERTYX|CLM|<version>|<event code>|
LEEF:2.0|CERTYX|CLM|<version>|certyx:<outcome or event name>|^|
```

Die QRadar-Event-ID ist die Ereignis-ID des LEEF-Headers: der Ereigniscode (`AUTHN-002`, `CONFIG-008`, …) bei `leef-1.0` und `leef-2.0`. Zeit: `devTime` mit `devTimeFormat` `yyyy-MM-dd'T'HH:mm:ss.SSSZ`. Schweregrad: `sev` 1–10. Wichtigste Attribute: `cat` (Kategorie), `usrName`, `src`, `identSrc` (Anmeldeereignisse), `proto`, `eventId` (Ereignis-ID), `schemaVersion`, `action`, `outcome`, `message`, `reason`, `userId`, `authMethod`, `method`, `resource`, `httpStatus`, `organization`, `organizationName`, `organizationKind`, `correlationId`, `resourceType`, `resourceId`, `changeType`, `instance`. Werte maskieren den Backslash und das Trennzeichen mit einem Backslash; Zeilenumbrüche und Tabulatoren werden zu Leerzeichen.

13.2 Empfängerseite (QRadar)

1. *Admin → Data Sources → Log Sources → Add* (QRadar-App Log Source Management).
2. **Log Source Type**: *Universal LEEF* (oder ein eigenes DSM, das Sie im DSM Editor anlegen). **Protocol**: *TLS Syslog*.
3. **Log Source Identifier**: die CERTYX-Instanz-ID (`Clm:Instance:Id`; in einer Container-Bereitstellung `CERTYX_INSTANCE_ID`, standardmäßig der Hostname der Konsole wie `certyx.example.com`, über Releases hinweg gleich) – das HOSTNAME-Feld jeder CERTYX-Syslog-Nachricht. Bei mehreren CERTYX-Instanzen legen Sie eine Protokollquelle pro Instanz-ID an oder lassen QRadar sie automatisch erkennen.
4. **TLS Listen Port**: 6514 (Standard) oder ein anderer Port. **Authentication Mode**: TLS oder, wenn CERTYX ein Clientzertifikat vorlegt, TLS mit Clientauthentifizierung. **Certificate Type**: das von QRadar erzeugte Zertifikat oder *Provide Certificate* mit einem Zertifikat für den QRadar-Hostnamen.
5. Verteilen Sie die Änderungen und erlauben Sie TCP zum Listen-Port vom CERTYX-Server.

6. **DSM Editor.** Ordnen Sie die Event-IDs (Ereigniscodes) QIDs zu; der Ereigniscode-Katalog führt jeden Code mit seiner Kategorie auf. Legen Sie benutzerdefinierte Ereigniseigenschaften für die CERTYX-Schlüssel an, die Sie durchsuchen oder auswerten möchten, zum Beispiel `eventId` (Deduplizierung), `schemaVersion`, `organization`, `organizationName`, `correlationId`, `authMethod`, `reason`, `resourceType`, `resourceId`, `changeType`. QRadar übernimmt die Ereigniszeit aus `devTime`, wenn `devTimeFormat` vorhanden ist; prüfen Sie, dass *Log Activity* die CERTYX-Zeit und nicht die Empfangszeit zeigt.

13.3 CERTYX-Seite

Transport `syslog-tls`, Format `leef-2.0` (oder `leef-1.0`, `leef-compat`), Syslog-Host = Hostname des QRadar-Event-Collectors (in dessen Zertifikat genannt), Port = TLS Listen Port, Framing **Oktettzählung (RFC 5425)**, optional **LEEF-Trennzeichen** für `leef-2.0` (standardmäßig TAB; zum Beispiel `^` oder `x5E`, wenn Ihr DSM das Caret erwartet). Verwendet QRadar sein erzeugtes Zertifikat oder eine private CA, fügen Sie das Stammzertifikat dieses Zertifikats (PEM) unter **Vertrauenswürdige Stammzertifikate** ein. **Verbindung prüfen**, **Testereignis senden**, dann suchen Sie das Ereignis in *Log Activity*, gefiltert nach der Protokollquelle (Event ID `INTEGRATION-001`). Beispiele: Beispielereignisse.

14 ArcSight

CEF über Syslog/TLS überträgt das ArcSight Common Event Format Version 0, Hersteller `CERTYX`, Produkt `CLM`.

14.1 Das CEF-Format wählen

Format	Device Event Class ID	Verwenden für
`cef` (empfohlen)	Ereigniscode (`AUTHN-002`, …); `externalId` = Ereignis-ID	Neue ArcSight-Integrationen, jeder CEF-Parser
`cef-compat`	`certyx:<outcome>` oder `certyx:<event name>`; `externalId` = Journalsequenz, `certyxEventId` = Ereignis-ID	Bestehende ArcSight-Inhalte, die auf der ersten Version aufbauen

Zuordnung des Formats `cef` (unbekannte und leere Werte werden ausgelassen; Header-Felder maskieren den Backslash und den senkrechten Strich und wandeln Zeilenumbrüche in Leerzeichen um; Erweiterungswerte maskieren `\`, `=`, CR und LF):

CEF-Schlüssel	Kanonisches Feld
Device Event Class ID	`event_code`
Name	`message`
Severity	`severity` (0–10)
`rt`	`timestamp` (Epochenmillisekunden)
`externalId`	`event_id`
`act`, `cat`, `outcome`, `msg`, `reason`	`action`, `category`, `outcome`, `message`, `reason`
`src`	`src_ip`
`suser`, `suid`	`username` (sonst `user_id`), `user_id`
`requestMethod`, `request`, `requestClientApplication`	`http_method`, `http_route`, `user_agent`
`app`, `dvchost`	`protocol`, `hostname`
`cn1` (`httpStatus`)	`http_status`
`cs1` … `cs6`	`organization_id`, `organization_name`, `correlation_id`, `resource_type`, `resource_id`, `schema_version` (Bezeichnungen in `cs1Label` … `cs6Label`)
`flexString1` (`changeType`), `flexString2` (`authMethod`)	`change_type`, `authentication_method`

14.2 Empfängerseite (ArcSight SmartConnector)

1. Installieren Sie einen SmartConnector vom Typ **Syslog NG Daemon** (den Syslog-Connector-Typ, der TLS annimmt) auf einem Host, den der CERTYX-Server erreichen kann.
2. Wählen Sie in der Connector-Einrichtung das Protokoll **TLS** (Raw TCP over TLS) und den Port, zum Beispiel 6514.
3. Importieren Sie das Serverzertifikat und den Schlüssel für den Hostnamen des Connectors in den Keystore des Connectors, wie in der SmartConnector-Dokumentation beschrieben; aktivieren Sie die Clientzertifikat-Authentifizierung nur, wenn CERTYX ein Clientzertifikat vorlegt (mTLS).
4. Registrieren Sie den Connector bei ArcSight ESM oder Logger als Ziel. CEF-Nachrichten werden vom Connector ohne Parser-Override geparst; die Bezeichnungen `cs1` – `cs6` kommen als Bezeichnungen der *Device Custom String*-Felder an.
5. Erlauben Sie TCP 6514 vom CERTYX-Server.

14.3 CERTYX-Seite

Transport `syslog-tls`, Format `cef` (oder `cef-compat`), Syslog-Host = Hostname des Connectors, Port 6514, Framing **Oktettzählung (RFC 5425)**, vertrauenswürdige Stammzertifikate für eine private CA. **Verbindung prüfen**, **Testereignis senden**, und suchen Sie nach einem Ereignis mit *Device Vendor* CERTYX, *Device Event Class ID* `INTEGRATION-001` und einer *External ID*, die der ID des Testereignisses entspricht. Beispiele: Beispielereignisse.

15 Elastic (ECS)

Das Profil `ecs` schreibt Elastic-Common-Schema-8.11.0-Dokumente: `@timestamp`, `message`, `ecs.version`, `event.id` (Ereignis-ID), `event.code` (Ereigniscode), `event.kind` `event`, `event.category`, `event.type`, `event.action`, `event.outcome`, `event.severity`, `event.created`, `event.dataset` `certyx.security`, `event.provider`, `event.reason`, `user.id`, `user.name`, `source.ip`, `host.hostname`, `service.*`, `http.request.method`, `http.request.id`, `http.response.status_code`, `url.path` (Routenvorlage), `organization.id`, `organization.name`, `labels.schema_version`, `labels.certyx_category`. Unbekannte Werte werden ausgelassen.

Option A – Logstash über HTTPS (empfohlen). CERTYX: Transport `https`, Format `ndjson`, Profil `ecs`, ein Clientzertifikat für mTLS. Logstash:

```
input {
  http {
    port => 8443
    ssl_enabled => true
    ssl_certificate => "/etc/logstash/tls/logstash.example.internal.pem"
    ssl_key => "/etc/logstash/tls/logstash.example.internal.key"
    ssl_client_authentication => "required"
    ssl_certificate_authorities => ["/etc/logstash/tls/certyx-client-ca.pem"]
    additional_codecs => { "application/x-ndjson" => "json_lines" }
    remote_host_target_field => "[@metadata][input][remote_host]"
    request_headers_target_field => "[@metadata][input][headers]"
  }
}
output {
  elasticsearch {
    hosts => ["https://elasticsearch.example.internal:9200"]
    data_stream => true
    data_stream_type => "logs"
    data_stream_dataset => "certyx.security"
    data_stream_namespace => "default"
    document_id => "%{[event][id]}"
  }
}
```

Der Logstash-Eingang `http` prüft die HMAC-Signatur nicht; stattdessen authentifiziert das Clientzertifikat CERTYX (die Signatur wird trotzdem gesendet). Die beiden Optionen `…_target_field` halten die vom Eingang ergänzten Anfragemetadaten aus dem Dokument heraus, sodass sie sich nicht mit den eigenen ECS-Feldern `http`, `url` und `host` des Dokuments vermischen. Der Datenstrom ist `logs-certyx.security-default`; Namen von Datenströmen müssen kleingeschrieben sein und dürfen im Dataset kein `-` enthalten.

Option B – Syslog über TLS an Logstash oder die Elastic-Agent-Integration *Custom TCP Logs* mit TLS: CERTYX-Transport `syslog-tls`, Format `rfc5424`, Profil `ecs`, Framing **Zeilenumbruch**; trennen Sie beim Empfänger die Zeilen, entfernen Sie den Syslog-Header bis zum ersten `{` (und das Byte Order Mark, oder bitten Sie den Betreiber, `Clm:Siem:Syslog:ByteOrderMark=false` zu setzen) und dekodieren Sie das JSON (Logstash-Filter `json`, Agent-Prozessor `decode_json_fields`).

Option C – Dateiarchiv mit Format `ndjson` und Profil `ecs`, stapelweise importiert (zum Beispiel mit Filebeat oder Elastic Agent, die die heruntergeladenen Dateien lesen).

Deduplizierung. Verwenden Sie `event.id` als `_id` des Dokuments (`document_id` oben): Ein erneut gesendetes Ereignis wird als Duplikat abgelehnt (`409`), statt doppelt indiziert zu werden. Ein Datenstrom prüft die `_id` nur innerhalb seines aktuellen Backing-Index; nach einem Rollover kann ein spätes Duplikat erneut indiziert werden, suchen Sie daher mit `event.id`, wenn es auf genaue Zahlen ankommt. Alternativ kann der Logstash-Filter `fingerprint` mit `source => "[event][id]"` die `_id` erzeugen.

16 OCSF-Verarbeiter und Amazon Security Lake

Das Profil `ocsf` schreibt OCSF-1.1.0-JSON-Dokumente. Für jedes Ereignis wählt CERTYX eine Klasse und prüft deren Pflichtattribute:

Klasse	Ereignisse
Authentication (3002)	Kategorie `authentication`: Aktivität 1 Logon (Anmeldung), 2 Logoff (Abmeldung), 99 Other
Account Change (3001)	Konto- und SCIM-Benutzeränderungen, MFA-Zurücksetzung: 1 Create, 5 Disable, 6 Delete, 11 MFA Factor Disable, 99 Other
API Activity (6003)	andere Anfrageereignisse: 1 Create (POST), 2 Read (GET/HEAD), 3 Update (PUT/PATCH), 4 Delete (DELETE), 99 Other
Base Event (0)	alles andere sowie der Rückfall

Jedes Dokument enthält `class_uid`, `category_uid`, `activity_id`, `type_uid`, `severity_id`, `status_id`, `time` (Epochenmillisekunden), `message`, `metadata.uid` (Ereignis-ID), `metadata.event_code`, `metadata.version`, `metadata.product` sowie `unmapped.certyx_schema_version`, `unmapped.certyx_category`, `unmapped.certyx_action`. Fehlen einem Ereignis die Pflichtattribute seiner Klasse – zum Beispiel einer anonymen Anfrage ohne `actor.user` oder einer Anfrage ohne Quelladresse –, wird es als Base Event geschrieben, und die Gründe werden in `unmapped.certyx_ocsf_fallback` aufgeführt. CERTYX beansprucht nie eine Klasse ohne Prüfung; die Prüfung kontrolliert Vorhandensein und Typ der Pflichtattribute, nicht das vollständige OCSF-Schema.

Zustellung. Transport `https` × `ndjson` × `ocsf` an Ihren Collector, `splunk-hec` × `ndjson` × `ocsf`, `syslog-tls` × `rfc5424` × `ocsf` oder das Dateiarchiv `file` × `ndjson` × `ocsf` (`certyx-events-YYYY-MM-DD.ndjson`).

Amazon Security Lake. Eine benutzerdefinierte Quelle (Custom Source) von Security Lake nimmt OCSF in Apache Parquet an, eine OCSF-Ereignisklasse pro benutzerdefinierter Quelle, geschrieben in den S3-Speicherort der Quelle. CERTYX liefert OCSF-JSON; die Umwandlung ist Aufgabe des Empfängers: Betreiben Sie einen Collector (zum Beispiel hinter dem HTTPS-Webhook oder einen, der das heruntergeladene `ndjson`-Archiv liest), der die Dokumente nach `class_uid` gruppiert, in Parquet umwandelt und in die benutzerdefinierte Quelle dieser Klasse schreibt. Als Base Event (`class_uid` 0) geschriebene Ereignisse benötigen eine eigene benutzerdefinierte Quelle oder müssen von Ihrer Pipeline herausgefiltert werden. Deduplizieren Sie nach `metadata.uid`.

17 Microsoft Sentinel

CERTYX schreibt über die Azure Monitor Logs Ingestion API in eine benutzerdefinierte Log-Analytics-Tabelle: Ein Datensammlungsendpunkt (DCE) empfängt die Daten, eine Datensammlungsregel (DCR) ordnet den Stream der Tabelle zu, und eine Microsoft-Entra-Anwendung authentifiziert CERTYX.

17.1 Was die Tabelle erhält

Transport `azure-monitor`, Format `sentinel`: eine Zeile pro Ereignis mit den Spalten der ersten Version plus `EventId` (string, `<installation id>:<journal sequence>`). Deklarieren Sie `EventId` in der Tabelle und im Stream, um die Spalte zu speichern; eine Regel ohne sie nimmt die übrigen Spalten weiterhin auf. Spaltenliste: Sentinel-Spalten.

17.2 Empfängerseite (Azure)

1. **Arbeitsbereich.** Ein Log-Analytics-Arbeitsbereich mit aktiviertem Microsoft Sentinel.
2. **Datensammlungsendpunkt.** *Monitor → Data Collection Endpoints → Create*, in der Region des Arbeitsbereichs. Kopieren Sie dessen **Logs Ingestion**-URL, zum Beispiel `https://certyx-dce-a1b2.westeurope-1.ingest.monitor.azure.com`.
3. **Tabelle und Regel.** Im Arbeitsbereich: *Tables → Create → New custom log (DCR-based)*: Tabellenname `CertyxSecurity` (Azure ergänzt `_CL`), *Create a new data collection rule*, den Endpunkt aus Schritt 2 auswählen, das Beispiel unten hochladen, die Transformation `source` beibehalten. Oder legen Sie beides mit der Azure CLI an (unten).
4. **Unveränderliche ID.** Öffnen Sie die Regel → *JSON View* → kopieren Sie `immutableId` (`dcr-`, gefolgt von 32 hexadezimalen Zeichen). Der Stream-Name lautet `Custom-CertyxSecurity_CL`.
5. **Anwendung.** *Microsoft Entra ID → App registrations → New registration*: Name `certyx-siem-export`, Single Tenant, kein Redirect-URI. Kopieren Sie **Directory (tenant) ID** und **Application (client) ID**. *Certificates & secrets → New client secret*: Kopieren Sie den **Value** des Geheimnisses (nicht dessen Secret ID) und notieren Sie sein Ablaufdatum.
6. **Berechtigungen.** Die Anwendung benötigt keine API-Berechtigungen und keine Administratorzustimmung. Weisen Sie ihr die Rolle **Monitoring Metrics Publisher** auf der Datensammlungsregel zu: Regel → *Access control (IAM) → Add role assignment* → Monitoring Metrics Publisher → die Anwendung. Eine neue Rollenzuweisung kann mehrere Minuten bis zur Wirksamkeit benötigen; bis dahin antwortet die API mit 403.
7. **Netzwerk.** Erlauben Sie ausgehendes HTTPS vom CERTYX-Server zu `login.microsoftonline.com` und zum DCE-Host.

Beispiel für den Tabellenassistenten (ein Datensatz mit jeder Spalte):

```
[{"TimeGenerated":"2026-10-11T08:15:02.417Z","EventId":"<installation id>:48213","Sequence":48213,"Organization":"3f2c9a6e-8d1b-4c55-9a0e-2b7f1d4c8e90","OrganizationName":"Example Ltd","OrganizationKind":"organization","Actor":"anonymous","SourceIp":"203.0.113.24","AgentFingerprint":"","Method":"POST","Route":"/auth/customer/login","Status":401,"Outcome":"authentication","TraceId":"0HN7J3K2L1M0:00000003","ErrorType":"","Instance":"certyx-web-1","EventHash":"9C1F","Changes":[],"User":"jane.doe@example.com","UserVerified":true,"UserOrganization":"3f2c9a6e-8d1b-4c55-9a0e-2b7f1d4c8e90","AuthAction":"authentication.sign-in.failed","AuthMethod":"password","AuthResult":"failure","Reason":"invalid-credentials","UserAgent":"Mozilla/5.0"}]
```

Alternative mit der Azure CLI (ersetzen Sie die Werte in spitzen Klammern):

```
az monitor log-analytics workspace table create --resource-group <rg> --workspace-name <workspace> \
  --name CertyxSecurity_CL --columns TimeGenerated=datetime EventId=string Sequence=long Organization=string \
  OrganizationName=string OrganizationKind=string Actor=string SourceIp=string AgentFingerprint=string \
  Method=string Route=string Status=int Outcome=string TraceId=string ErrorType=string Instance=string \
  EventHash=string Changes=dynamic User=string UserVerified=boolean UserOrganization=string AuthAction=string \
  AuthMethod=string AuthResult=string Reason=string UserAgent=string

az monitor data-collection rule create --resource-group <rg> --location <region> \
  --name certyx-dcr --rule-file certyx-dcr.json

az role assignment create --assignee <application-client-id> --role "Monitoring Metrics Publisher" \
  --scope /subscriptions/<subscription>/resourceGroups/<rg>/providers/Microsoft.Insights/dataCollectionRules/certyx-dcr
```

`certyx-dcr.json` (der Stream deklariert dieselben Spalten wie die Tabelle):

```
{
  "location": "<region>",
  "properties": {
    "dataCollectionEndpointId":
```

```
"/subscriptions/<subscription>/resourceGroups/<rg>/providers/Microsoft.Insights/dataCollectionEndpoints/<dce>",
    "streamDeclarations": {
      "Custom-CertyxSecurity_CL": {
        "columns": [
          { "name": "TimeGenerated", "type": "datetime" }, { "name": "EventId", "type": "string" },
          { "name": "Sequence", "type": "long" }, { "name": "Organization", "type": "string" },
          { "name": "OrganizationName", "type": "string" }, { "name": "OrganizationKind", "type": "string" },
          { "name": "Actor", "type": "string" }, { "name": "SourceIp", "type": "string" },
          { "name": "AgentFingerprint", "type": "string" }, { "name": "Method", "type": "string" },
          { "name": "Route", "type": "string" }, { "name": "Status", "type": "int" },
          { "name": "Outcome", "type": "string" }, { "name": "TraceId", "type": "string" },
          { "name": "ErrorType", "type": "string" }, { "name": "Instance", "type": "string" },
          { "name": "EventHash", "type": "string" }, { "name": "Changes", "type": "dynamic" },
          { "name": "User", "type": "string" }, { "name": "UserVerified", "type": "boolean" },
          { "name": "UserOrganization", "type": "string" }, { "name": "AuthAction", "type": "string" },
          { "name": "AuthMethod", "type": "string" }, { "name": "AuthResult", "type": "string" },
          { "name": "Reason", "type": "string" }, { "name": "UserAgent", "type": "string" }
        ]
      }
    },
    "destinations": {
      "logAnalytics": [
        { "name": "workspace", "workspaceResourceId":
"/subscriptions/<subscription>/resourceGroups/<rg>/providers/Microsoft.OperationalInsights/workspaces/<workspace>" }
      ]
    },
    "dataFlows": [
      { "streams": ["Custom-CertyxSecurity_CL"], "destinations": ["workspace"], "transformKql": "source", "outputStream":
"Custom-CertyxSecurity_CL" }
    ]
  }
}
```

Eine Regel, die angelegt wurde, bevor es `EventId`, `OrganizationName`, `OrganizationKind` oder die Anmeldespalten gab, nimmt die übrigen Spalten weiterhin auf; fügen Sie die fehlenden Spalten der Tabelle und dem Stream hinzu, um sie zu speichern.

17.3 CERTYX-Seite

1. Transport **Microsoft Sentinel** (`azure-monitor`), Format `sentinel`.
2. **HTTPS-Adresse**: die Logs-Ingestion-URL des Datensammlungsendpunkts.
3. **Datensammlungsregel (unveränderliche ID)**: `dcr-…` aus der JSON-Ansicht der Regel.
4. **Stream**: `Custom-CertyxSecurity_CL`.
5. **Microsoft-Entra-Mandanten-ID** und **Anwendungs-ID (Client)**: GUIDs aus der App-Registrierung.
6. **Geheimer Clientschlüssel**: der Value des Geheimnisses. Erneuern Sie ihn in Entra vor seinem Ablauf und geben Sie den neuen Wert hier ein.
7. Speichern, **Verbindung prüfen**, **Testereignis senden**.

CERTYX bezieht ein Token mit dem Client-Credentials-Flow von `https://login.microsoftonline.com/<tenant>/oauth2/v2.0/token` (Scope `https://monitor.azure.com//.default`) und sendet den Stapel per POST an `<endpoint>/dataCollectionRules/<immutable ID>/streams/<stream>?api-version=2023-01-01`. Tokens werden bis kurz vor ihrem Ablauf wiederverwendet. Souveräne Clouds verwenden den Autoritätshost, den der Betreiber in `Clm:Siem:SentinelAuthorityHost` festlegt.

So sieht Erfolg aus: Das Ziel hat das Testereignis angenommen, und seine Ereignis-ID wird angezeigt. Die Daten erscheinen nach der Aufnahmeverzögerung (typischerweise einige Minuten) in der Tabelle:

```
CertyxSecurity_CL
| where Actor == "siem-test"

CertyxSecurity_CL
| summarize arg_max(TimeGenerated, *) by EventId
| where Outcome == "authentication" and AuthResult == "failure"
| summarize Failures = count() by User, SourceIp, Reason, bin(TimeGenerated, 15m)
```

18 OpenTelemetry Logs (OTLP/HTTP)

CERTYX sendet das kanonische Ereignis als OpenTelemetry-Log-Einträge über OTLP/HTTP an jeden Empfänger, der den OTLP-Logs-Endpunkt implementiert: den OpenTelemetry Collector, Grafana Alloy, Grafana Loki oder den eigenen OTLP-Endpunkt eines Backends. Nutzen Sie es, um Sicherheitsereignisse mit den Traces und Metriken Ihrer Anwendungen zu korrelieren; die klassischen SIEM-Formate bleiben daneben verfügbar.

18.1 Was der Empfänger erhält

Transport `otlp-http`, Format `otlp-protobuf` (binäres Protobuf, `Content-Type: application/x-protobuf`, der Standard von OTLP/HTTP) oder `otlp-json` (OTLP/JSON, `Content-Type: application/json`), Profil `native`. Jeder Stapel ist ein `POST` einer `ExportLogsServiceRequest` (opentelemetry-proto `collector/logs/v1`) an die URL genau wie eingegeben – CERTYX hängt `/v1/logs` nicht an –, optional mit `Content-Encoding: gzip`. Die Protobuf-Nachricht kodiert CERTYX selbst nach den Feldnummern von opentelemetry-proto v1 (ohne Protobuf-Bibliothek).

Die Anfrage enthält eine `ResourceLogs` je unterschiedlicher Ressource (praktisch eine je Organisation des Stapels), eine `ScopeLogs` (Scope `certyx.security-journal`, Version = Produktversion) und einen `LogRecord` je Ereignis:

OTLP	Aus dem kanonischen Ereignis
Ressource `service.name`	`certyx`
Ressource `service.version`, `deployment.environment.name`, `host.name`, `host.ip`	`version`, `environment`, `hostname`, `host_ip` (Array)
Ressource `certyx.organization.id`, `certyx.organization.name`, `certyx.organization.kind`	`organization_id`, `organization_name`, `organization_kind`
`Timestamp`, `ObservedTimestamp`	`timestamp`, `observed_timestamp` (Unix-Nanosekunden)
`SeverityNumber`, `SeverityText`	`severity` 0–3 → 9 `INFO`, 4–6 → 13 `WARN`, 7–8 → 17 `ERROR`, 9–10 → 21 `FATAL`
`Body`	`message`; mit *Inhalt des Log-Eintrags: Kanonisches Ereignis* das ganze kanonische Dokument als Schlüssel-Wert-Liste
`TraceId`, `SpanId`	`trace_id`, `span_id` (W3C-IDs der Anfrage, Anreicherung); ausgelassen, wenn unbekannt
`event.name`	`action`
`user.id`, `user.name`, `user.roles`, `session.id`, `user_agent.original`	`user_id`, `username`, `user_role` (Array), `session_id`, `user_agent`
`client.address`, `client.port`, `server.address`, `server.port`	`src_ip`, `src_port`, `dst_ip`, `dst_port`
`http.request.method`, `http.route`, `http.response.status_code`, `network.protocol.name`, `error.type`	`http_method`, `http_route` (die Routenvorlage), `http_status`, `protocol`, `error_type`
`certyx.event.id`, `certyx.event.code`, `certyx.event.category`, `certyx.event.outcome`, `certyx.event.severity`, …	`event_id`, `event_code`, `category`, `outcome`, `severity`; jedes andere kanonische Feld als `certyx.<Feld>`; `certyx.changes` ist der kompakte JSON-Text der Änderungen
Eigene Felder einer Feldzuordnung	Attribute mit ihren eigenen Namen

Unbekannte Werte werden ausgelassen. In OTLP/JSON sind 64-Bit-Ganzzahlen und Zeiten dezimale Zeichenketten und Trace- und Span-IDs hexadezimale Zeichenketten; in Protobuf sind es Ganzzahlen und Bytes. Deduplizieren Sie nach dem Attribut `certyx.event.id`.

Antworten. 2xx ohne `partial_success` stellt den Stapel zu. Ein 2xx mit `partial_success.rejected_log_records` über null ist ein Teilerfolg: Wie die OTLP-Spezifikation verlangt, sendet CERTYX die Anfrage **nicht** erneut (der Empfänger hat die angenommenen Einträge behalten). Die Position rückt hinter den Stapel, die Ablehnung wird als `siem.event.rejected` (Code `INTEGRATION-013`) mit dem Ziel, der Zahl der abgelehnten Einträge, den Ereignis-IDs und Journalsequenzen der Anfrage (OTLP nennt die abgelehnten Einträge nicht, daher werden alle der Anfrage aufgeführt; bei mehr als 200 Ereignissen das erste und das letzte) und der `error_message` des Empfängers (druckbar, höchstens 200 Zeichen) protokolliert, und der Vorfall `siem-rejected` wird mit der Anzahl ausgelöst; er bleibt offen, bis ein Betreiber ihn schließt. **Nichts geht unbemerkt verloren**: Die Ereignisse bleiben im Journal und in Dateiarchiven, und nach der Korrektur des Empfängers sendet Erneut senden ab der ersten aufgeführten Sequenz sie erneut (der Empfänger dedupliziert die angenommenen nach `certyx.event.id`). HTTP 400, 413 und 422 teilen den Stapel bis zum abgelehnten Ereignis auf (Abgelehnte Ereignisse); 401, 403, 404, 415, 429 und 5xx wiederholen den ganzen Stapel mit Backoff, und ein `Retry-After` bei 429 oder 503 wird beachtet (der nächste Versuch wartet mindestens so lange, höchstens die längste Wartezeit des Ziels).

18.2 Empfängerseite (OpenTelemetry Collector)

Der `otlp`-Receiver des OpenTelemetry Collectors (Contrib-Distribution für die Erweiterung `bearertokenauth`) mit TLS auf Port 4318 und einem Bearer-Token:

```
extensions:
  bearertokenauth:
    token: ${env:CERTYX_OTLP_TOKEN}
receivers:
  otlp:
    protocols:
      http:
        endpoint: 0.0.0.0:4318
        tls:
          cert_file: /etc/otelcol/tls/server.crt
          key_file: /etc/otelcol/tls/server.key
          # client_ca_file: /etc/otelcol/tls/certyx-client-ca.pem
        auth:
          authenticator: bearertokenauth
processors:
  batch: {}
exporters:
  debug:
    verbosity: detailed
service:
  extensions: [bearertokenauth]
  pipelines:
    logs:
      receivers: [otlp]
      processors: [batch]
      exporters: [debug]
```

1. Geben Sie dem Receiver ein Zertifikat, dessen Namen den in CERTYX eingegebenen Hostnamen enthalten; fügen Sie eine private CA in CERTYX unter *Vertrauenswürdige Stammzertifikate* ein.
2. Für gegenseitiges TLS setzen Sie `client_ca_file` auf die CA des CERTYX-Clientzertifikats und geben dieses Zertifikat mit seinem Schlüssel in CERTYX ein (mTLS); das Bearer-Token ist dann optional.
3. Ersetzen Sie den `debug`-Exporter durch den Exporter Ihres Backends.
4. In CERTYX: URL `https://collector.example.com:4318/v1/logs`, Headername `Authorization` (Standard), Geheimnis `Bearer <Token>`.

18.3 Grafana Alloy und Loki

Grafana Alloy empfängt OTLP mit `otelcol.receiver.otlp` und leitet an den nativen OTLP-Endpunkt von Loki weiter (Loki 3, Pfad `/otlp/v1/logs` ; der Exporter hängt `/v1/logs` an):

```
otelcol.receiver.otlp "certyx" {
  http {
    endpoint = "0.0.0.0:4318"
    tls {
      cert_file = "/etc/alloy/tls/server.crt"
      key_file  = "/etc/alloy/tls/server.key"
    }
  }
  output {
    logs = [otelcol.exporter.otlphttp.loki.input]
  }
}

otelcol.exporter.otlphttp "loki" {
  client {
    endpoint = "http://loki:3100/otlp"
  }
}
```

Loki behält einige Ressourcenattribute wie `service.name` als Indexlabels und die übrigen Attribute als strukturierte Metadaten, wobei Punkte durch Unterstriche ersetzt werden, zum Beispiel `{service_name="certyx"} | certyx_event_code="AUTHN-002"` . CERTYX kann auch direkt an `https://<loki>/otlp/v1/logs` senden, wenn Loki über TLS erreichbar ist und höchstens einen Header benötigt; ein mandantenfähiges Loki, das zusätzlich `X-Scope-OrgID` braucht, muss hinter Alloy oder dem Collector stehen.

18.4 Andere OTLP-Backends

Datadog, Elastic und andere Anbieter dokumentieren ihre eigene OTLP-Annahme. Der allgemeine, spezifikationsbasierte Weg ist der OpenTelemetry Collector oder Alloy wie oben mit dem Exporter, den der Anbieter dokumentiert. Ein Backend, das einen OTLP/HTTP-Logs-Endpunkt über HTTPS anbietet und sich mit einem Header authentifiziert, kann direkt verwendet werden: Geben Sie die Logs-URL, den Headernamen und den Headerwert genau so ein, wie der Anbieter sie dokumentiert. CERTYX sendet genau einen Authentifizierungs-Header und keine weiteren anbieterspezifischen Header.

18.5 CERTYX-Seite

1. **Transport** *OpenTelemetry Logs (OTLP/HTTP)*; **Format** *OTLP Protobuf* (empfohlen) oder *OTLP/JSON*; Profil *native*.
2. **HTTPS-Adresse**: die vollständige Logs-URL des Empfängers, zum Beispiel `https://collector.example.com:4318/v1/logs` .
3. **Inhalt (Body) des Log-Eintrags**: *Nachricht* (die Ereignisnachricht) oder *Kanonisches Ereignis (Schlüssel-Wert-Liste)*. Alle Felder sind in beiden Fällen Attribute.
4. **Name des Authentifizierungs-Headers** (optional; leer ist `Authorization`) und **Wert des Authentifizierungs-Headers** als Geheimnis (zum Beispiel `Bearer <Token>` oder ein API-Schlüssel). Der Wert wird verschlüsselt gespeichert und nie wieder angezeigt; ohne Wert wird kein Header gesendet. `Host` , `Content-Type` , `Content-Length` , `Content-Encoding` , `Transfer-Encoding` , `Connection` , `Cookie` und `Proxy-` -Header werden abgelehnt.
5. **Anfragen mit gzip komprimieren** (empfohlen für entfernte Empfänger).
6. Vertrauenswürdige Stammzertifikate, Clientzertifikat und Sperrprüfung (TLS), *Nur senden*, Feldzuordnung und die Zustellwerte funktionieren wie bei den anderen Transporten.
7. Speichern, **Verbindung prüfen** und **Testereignis senden**: das Ergebnis `otlp-accepted` nennt die Ereignis-ID; suchen Sie im Backend nach `certyx.event.id` mit diesem Wert.

19 Verbindungsprüfung und Testereignis

Verbindung prüfen durchläuft die Stufen des Transports des Ziels, ohne etwas zu senden: DNS → TCP → TLS (mit dem Clientzertifikat und der Sperrprüfung) für jeden Netzwerktransport; bei einem Dateiarchiv prüft die Stufe *Archiv*, ob das Archivverzeichnis beschreibbar ist. **Testereignis senden** durchläuft dieselben Stufen und stellt anschließend ein synthetisches Ereignis zu, ohne die Position des Ziels zu verschieben. Die erste fehlschlagende Stufe beendet die Prüfung. Jede Stufe zeigt ihr Ergebnis und die ermittelten Fakten:

Stufe	Erfolgreich	Angezeigte Fakten
DNS	`dns-ok`, `dns-literal`	Die Adressen, zu denen der Host auf dem CERTYX-Server aufgelöst wird
TCP	`tcp-ok`	Die verbundene Adresse und der Port
TLS	`tls-ok`	TLS-Version, Antragsteller, Aussteller und Ablauf des Zertifikats
Zustellung	`hec-accepted`, `syslog-written`, `http-accepted`, `otlp-accepted`	Splunk-Index und Sourcetype; Syslog-Framing; die ID des Testereignisses
Archiv	`archive-ok`	Anzahl der Dateien, ihr Platzbedarf und der freie Speicherplatz

Das Testereignis. Seine `event_id` ist eine neue UUID (es hat keine Journalsequenz), `event_code` `INTEGRATION-001`, `category` `integration`, `action` `siem.test`, `user_id` / `Actor` `siem-test`, Ergebnis business; im JSON-Datensatz `Sequence` 0, `Method` `INTERNAL`, `Route` `business/siem.test`. Es durchläuft Format und Profil des Ziels wie jedes andere Ereignis und zeigt daher genau, was der Empfänger erhalten wird. Das Ergebnis nennt die Ereignis-ID: Suchen Sie im Empfänger danach.

Bei Syslog-Zielen nennt ein erfolgreicher Test zusätzlich, was der Empfänger mit dem gewählten Framing tun muss (Oktettzählung: Am empfangenden Eingang muss die Oktettzählung eingeschaltet sein; Zeilenumbruch: Er muss an Zeilenumbrüchen trennen). Nimmt ein Empfänger TCP an, schließt den TLS-Handshake aber nicht ab – sein Eingang hat TLS ausgeschaltet –, meldet die Prüfung `tls-timeout` oder `tls-handshake-failed`, und der Export vermerkt statt einer Zeitüberschreitung *Der Empfänger hat die TCP-Verbindung angenommen, aber keinen TLS-Handshake abgeschlossen …*. Ein Dateiarchiv hat kein Testereignis.

Das Ergebnis der letzten Prüfung bleibt auf der Zielkarte und wird in das Sicherheitsjournal geschrieben (`siem.destination.checked`, `INTEGRATION-003`; `siem.destination.tested`, `INTEGRATION-002`). Tokens, Geheimnisse und Antwortinhalte erscheinen darin niemals.

20 Zustellung, Wiederholungen und Rückstand

- **Dauerhafte Warteschlange.** Das Sicherheitsjournal ist die Warteschlange: Jedes Ziel hat darin seine eigene Position (Cursor), die erst vorrückt, nachdem der Empfänger einen Stapel angenommen hat. Ereignisse werden in Journalreihenfolge gesendet. Ist ein Empfänger nicht erreichbar, geht nichts verloren.
- **Mindestens einmal.** Nach einer Zeitüberschreitung oder einer abgebrochenen Verbindung kann der Empfänger einen Stapel bereits gespeichert haben, den CERTYX erneut sendet; die erneut gesendeten Ereignisse tragen dieselbe `event_id`. Deduplizieren Sie danach (Deduplizierung).
- **Backoff.** Nach einer fehlgeschlagenen Zustellung wartet das Ziel vor dem nächsten Versuch: die **erste Wartezeit** (Standard 30 Sekunden, `Clm:Siem:Delivery:InitialDelay`), verdoppelt mit jedem weiteren aufeinanderfolgenden Fehler bis zur **längsten Wartezeit** (Standard 15 Minuten, `Clm:Siem:Delivery:MaxDelay`), jeweils zufällig um ±20 % gestreut (`Clm:Siem:Delivery:Jitter`), damit gemeinsam fehlschlagende Ziele nicht gleichzeitig wiederholen. Antwortet ein Empfänger mit 429 oder 503 und `Retry-After` (Splunk HEC, HTTPS, Sentinel, OTLP), wird mindestens so lange gewartet, höchstens die längste Wartezeit. Die erste erfolgreiche Zustellung setzt den Backoff zurück.
- **Vorfall** `siem-delivery`**.** Ein Fehler löst diesen Vorfall für den Plattformbetreiber als Warnung aus; nach **Versuche bis zur kritischen Warnung** aufeinanderfolgenden Fehlern (Standard 10, `Clm:Siem:Delivery:MaxAttempts`) wird er kritisch. Er wird nach der nächsten erfolgreichen Zustellung aufgelöst.
- **Rückstand.** Warten mehr Ereignisse auf ein Ziel als seine Grenze für die **Rückstandswarnung ab (Ereignisse)** (Standard 100.000, `Clm:Siem:Delivery:BacklogLimit`), wird der Vorfall `siem-backlog` ausgelöst. Es geht nichts verloren (Überlaufrichtlinie *behalten und warnen*): Die Ereignisse bleiben im Journal und werden gesendet, sobald das Ziel sie wieder annimmt. Der Vorfall wird aufgelöst, wenn der Rückstand unter die Grenze fällt.

- **Stapelgröße.** Bis zu 200 Ereignisse pro Ziel und Durchlauf (`Clm:Siem:BatchSize`); ein in der Konsole angelegtes Ziel kann eine eigene **Stapelgröße (Ereignisse)** (1–5000) festlegen. Halten Sie bei Splunk HEC einen Stapel unter dem `max_content_length` des Empfängers (1.000.000 Bytes ist die kleinste übliche Einstellung).
- **Pro Ziel.** Versuche, erste Wartezeit (1–3600 s), längste Wartezeit (1–86400 s, nicht kürzer als die erste), Stapelgröße, Rückstandswarnung (1–100.000.000) und *Ein Ereignis überspringen, das der Empfänger nach den Versuchen oben weiter ablehnt* können im Zielformular festgelegt werden; leere Felder verwenden die Serverstandards. Für Dateiarchive gelten sie nicht.

20.1 Abgelehnte Ereignisse

Manche Antworten besagen, dass der Empfänger die Daten selbst ablehnt, unabhängig von Wiederholungen: HTTP 400, 413 und 422 sowie die Splunk-HEC-Codes 5 (keine Daten), 6 (ungültiges Datenformat), 12 (Feld event erforderlich), 13 (Feld event darf nicht leer sein) und 15 (Fehler bei der Verarbeitung indizierter Felder). Dann:

1. teilt CERTYX den Stapel rekursiv in Hälften, bis das einzelne abgelehnte Ereignis gefunden ist;
2. stellt CERTYX jedes Ereignis davor zu und hält die Position unmittelbar davor an;
3. nennt CERTYX dessen `event_id` im letzten Fehler und im Vorfall `siem-delivery` (*Der Empfänger lehnt das Ereignis … ab*).

Standard: Das abgelehnte Ereignis wird mit Backoff wiederholt, und der Vorfall bleibt offen, bis der Empfänger es annimmt (korrigieren Sie den Empfänger – zum Beispiel den Index, die Feldgrenzen oder die Größengrenze –, dann stellt der nächste Versuch es zu).

Optional *Ein Ereignis überspringen, das der Empfänger nach den Versuchen oben weiter ablehnt* (Zielformular oder `Clm:Siem:Delivery:SkipRejected` für jedes Ziel): Nach Versuche bis zur kritischen Warnung** Versuchen wird das abgelehnte Ereignis übersprungen, die Position rückt darüber hinaus vor, und das Überspringen wird als `siem.event.skipped` (Code `INTEGRATION-011`) mit der Ereignis-ID, der Sequenz, der Anzahl der Versuche und dem Grund im Journal erfasst. Das Ereignis bleibt im Journal und in Dateiarchiven; es kann nach der Korrektur des Empfängers mit Erneut senden noch einmal gesendet werden.

Jeder andere Fehler – nicht erreichbarer Empfänger, TLS-Fehler, HTTP 401, 403, 404, 429 oder 5xx, andere HEC-Codes – wiederholt den gesamten Stapel.

20.2 Grenzen

- **Takt.** Ein Durchlauf alle 30 Sekunden (`Clm:Siem:Interval`), 20 Sekunden Zeitüberschreitung pro Anfrage oder Verbindung (`Clm:Siem:HttpTimeout`); ein Ziel im Backoff wird übersprungen, bis seine Wartezeit abgelaufen ist.
- **Unabhängigkeit.** Jedes Ziel hat seine eigene Position, seinen eigenen Backoff und seinen eigenen Status; ein fehlschlagendes Ziel verzögert die anderen nicht.
- **Verlauf.** Ein neues Ziel beginnt am Anfang des aufbewahrten Journals seines Bereichs.
- **Nicht erfasst.** Anmeldungen, die vor CERTYX blockiert werden (TLS-Handshake, Firewall, Ratenbegrenzungen am Eingangspunkt), können nicht exportiert werden; hinter einem Proxy ist die Quell-IP die Adresse des Proxys.
- **Konfigurationsziele.** Ziele, die der Betreiber in der Serverkonfiguration festlegt, werden in der Ansicht ihres Bereichs schreibgeschützt aufgeführt; sie verwenden die Zustellstandards des Servers. Dateiarchive können nur in der Konsole eingerichtet werden.
- **Organisationskennzeichnung.** Jedes Format überträgt Kennung, Name und Art der Organisation sowie die Instanz (Mehrere Organisationen in einem SIEM).

21 Erneut senden ab einer Sequenz oder einem Zeitpunkt

Erneut senden ab auf der Karte eines Ziels sendet Ereignisse ab einer Journalsequenz oder einem Zeitpunkt noch einmal, zum Beispiel nachdem der Empfänger Daten verloren hat oder nachdem ein abgelehntes Ereignis übersprungen wurde.

1. Wählen Sie **Erneut senden ab** auf der Zielkarte.
2. Geben Sie entweder eine **Sequenz** im Journal (1 oder höher; der Teil einer `event_id` nach dem Doppelpunkt) oder einen **Zeitpunkt** ein – nicht beides.

3. Bestätigen Sie. Die Position des Ziels wird auf unmittelbar vor diesen Punkt zurückgesetzt; die nächsten Durchläufe senden jedes Ereignis ab dort der Reihe nach erneut. Backoff und der Zustand des abgelehnten Ereignisses werden zurückgesetzt.

Regeln:

- Die Position wird nur zurück-, nie vorgesetzt: Ein Punkt, den das Ziel noch nicht zugestellt hat, wird abgelehnt (*das Ziel hat noch nicht so weit zugestellt*).
- Ein Dateiarchiv kann nicht erneut senden: Es behält jedes Ereignis; laden Sie stattdessen seine Dateien herunter.
- Jedes erneute Senden wird als `siem.destination.replayed` (Code `INTEGRATION-012`) mit der bisherigen und der neuen Position im Journal erfasst.
- Der Empfänger erhält die Ereignisse ein zweites Mal mit derselben `event_id`; deduplizieren Sie danach.
- Filter gelten wie bei jedem Durchlauf: Ereignisse, die *Nur senden* des Ziels ausschließt, werden nicht gesendet.

Berechtigungen. Ein Admin der Organisation mit einer MFA-Sitzung sendet für die eigenen Ziele der Organisation erneut. Der Plattformbetreiber sendet für Ziele der Bereiche *Plattformbetreiber* und *Alle Ereignisse* erneut; dies ist eine Änderung im Kontrollraum und erfordert den Step-up-Code des Betreibers. Sitzungen der mobilen App können nicht erneut senden.

API: `POST /api/administration/siem/{id}/replay` (Organisation) oder `POST /platform/api/siem/{scope}/{id}/replay` (Plattform, Bereich `platform` oder `all`, Step-up-Code in `X-Step-Up-Code`) mit `{"fromSequence": 48000}` oder `{"fromTime": "2026-10-18T06:00:00Z"}`; die Antwort ist `{"position": …}`.

22 Deduplizierung nach event_id

Die Zustellung erfolgt mindestens einmal: Nach einer Zeitüberschreitung, einer abgebrochenen Verbindung, einem unbestätigten HEC-Acknowledgement oder einem erneuten Senden kann der Empfänger ein Ereignis zweimal erhalten. Beide Kopien haben dieselbe Ereignis-ID, die über Installationen hinweg eindeutig ist (`<installation id>:<journal sequence>`; eine UUID für Testereignisse).

Empfänger	Wo die Ereignis-ID steht	Wie dedupliziert wird
Splunk	indiziertes Feld `event_id` (jedes HEC-Format), `event_id` im kanonischen Ereignis	`dedup event_id` in Suchen; für Zusammenfassungen `stats latest(*) by event_id`
Elastic	`event.id` (ECS), `event_id` (native)	als `_id` des Dokuments verwenden (`document_id => "%{[event][id]}"` oder der Filter `fingerprint`); Duplikate werden mit 409 abgelehnt
QRadar	LEEF-Attribut `eventId`	benutzerdefinierte Ereigniseigenschaft `eventId`; QRadar dedupliziert nicht von selbst – verwenden Sie sie in Suchen und Regeln (zum Beispiel *group by eventId*)
ArcSight	CEF `externalId` (`cef`), `certyxEventId` (`cef-compat`)	ArcSight verwirft Duplikate nicht von selbst; gruppieren oder filtern Sie in Regeln und Berichten nach External ID
Sentinel	Spalte `EventId`	`summarize arg_max(TimeGenerated, *) by EventId`
OCSF-Verarbeiter	`metadata.uid`	als Schlüssel des gespeicherten Datensatzes verwenden
Webhook / eigener Collector	`EventId`, `event_id`, `event.id` oder `metadata.uid` je nach Profil	eindeutiger Schlüssel in Ihrem Speicher
Dateiarchiv	Spalte `event_id` (`csv`), `event_id` / `event.id` / `metadata.uid` (`ndjson`); `Sequence` (`csv-compat`)	Archive enthalten eine Zeile nie doppelt, außer nach einem Absturz bei einem Tageswechsel (Zustellung, Neustarts und Speicherplatz)

Der Teil mit der Installations-ID ändert sich nur, wenn der Betreiber `Clm:Siem:InstallationId` ändert; halten Sie ihn stabil, sonst erkennen Empfänger erneut gesendete Ereignisse nicht mehr als Duplikate.

23 Nur senden: weniger Daten an das SIEM

Externe SIEM- und Log-Dienste werden oft nach aufgenommenem Datenvolumen lizenziert. *Nur senden* verringert, was ein einzelnes Ziel erhält, während ein Dateiarchiv den vollständigen Bestand auf Ihrem eigenen Speicher behält.

- **Ergebnisse** (Kontrollkästchen): Das Ziel erhält die ausgewählten Ergebnisse; ausgewählte Optionen werden mit *oder* verknüpft. *Konfigurations- und Geschäftsänderungen* fügt jedes Ereignis hinzu, das eine Konfigurations- oder Geschäftsänderung enthält (zum Beispiel `siem.destination.updated`, Benutzer- und Rollenänderungen, Erneuerungsentscheidungen, `agent.health.changed`), unabhängig von seinem Ergebniskästchen. Beispiel für einen lizenzschonenden SIEM-Feed: *Zugriff verweigert*, *Serverfehler*, *Anmeldeereignisse* und *Konfigurations- und Geschäftsänderungen*.
- **Mindestschweregrad** (Liste): Nur Ereignisse ab diesem Schweregrad werden gesendet. Schweregrad (0–10): empfangen (`attempt`) 2, Erfolg und geschäftliche Änderung 3, abgelehnt 5, verweigert und fehlgeschlagene Anmeldung 7, Serverfehler und Sperre 8.
- **Nur senden (Kategorien)**: eine oder mehrere der zehn Kategorien.
- **Nur senden (Ereigniscodes)**: Codes aus dem Ereigniscode-Katalog, zum Beispiel `AUTHN-002`, höchstens 200.
- **Gesendete Felder** (unter *Form und Größe*, eingeklappt): jedes Feld des gewählten Formats mit einem Kontrollkästchen. Das Ziel erhält genau die angekreuzten Felder; ein nicht angekreuztes Feld wird in der Formung als `excludeFields` gespeichert. Das gilt für jeden Transport und jedes Format außer den festen Layouts `cef-compat` und `leef-compat` (stellen Sie dort `cef` oder `leef` ein). Immer gesendet werden: in kanonischen Formaten die vom Schema verlangten Felder (`schema_version`, `event_id`, `event_code`, `category`, `action`, `outcome`, `outcome_original`, `severity`, `severity_label`, `message`, die drei Zeitstempel, `application`, `service_name`, `organization_kind`), bei Sentinel `TimeGenerated`; im `json`-Datensatz kann jedes Feld entfernt werden. **Alle auswählen** und **Nur erforderliche** setzen alle Kästchen auf einmal. Das Ergebnis zeigt die *Vorschau des Testereignisses*.
- **Weitere Filter** (eingeklappt): **Nur diese Routen senden** und **Diese Routen nie senden** (eine pro Zeile; eine Route umfasst jede Route, die mit ihr beginnt, zum Beispiel `/agent/v1/discovery/telemetry` für die regelmäßigen Berichte der Agenten), **Ereignisse dieser Akteure nie senden** (zum Beispiel `pending`, Anfragen vor der Authentifizierung), **Diese Kategorien nie senden** und **Diese Ereigniscodes nie senden**. Ein Ausschluss gilt immer vorrangig.
- Zustandsprüfungen werden nie gesendet, es sei denn, `Clm:Siem:ExportHealthProbes` ist eingeschaltet.

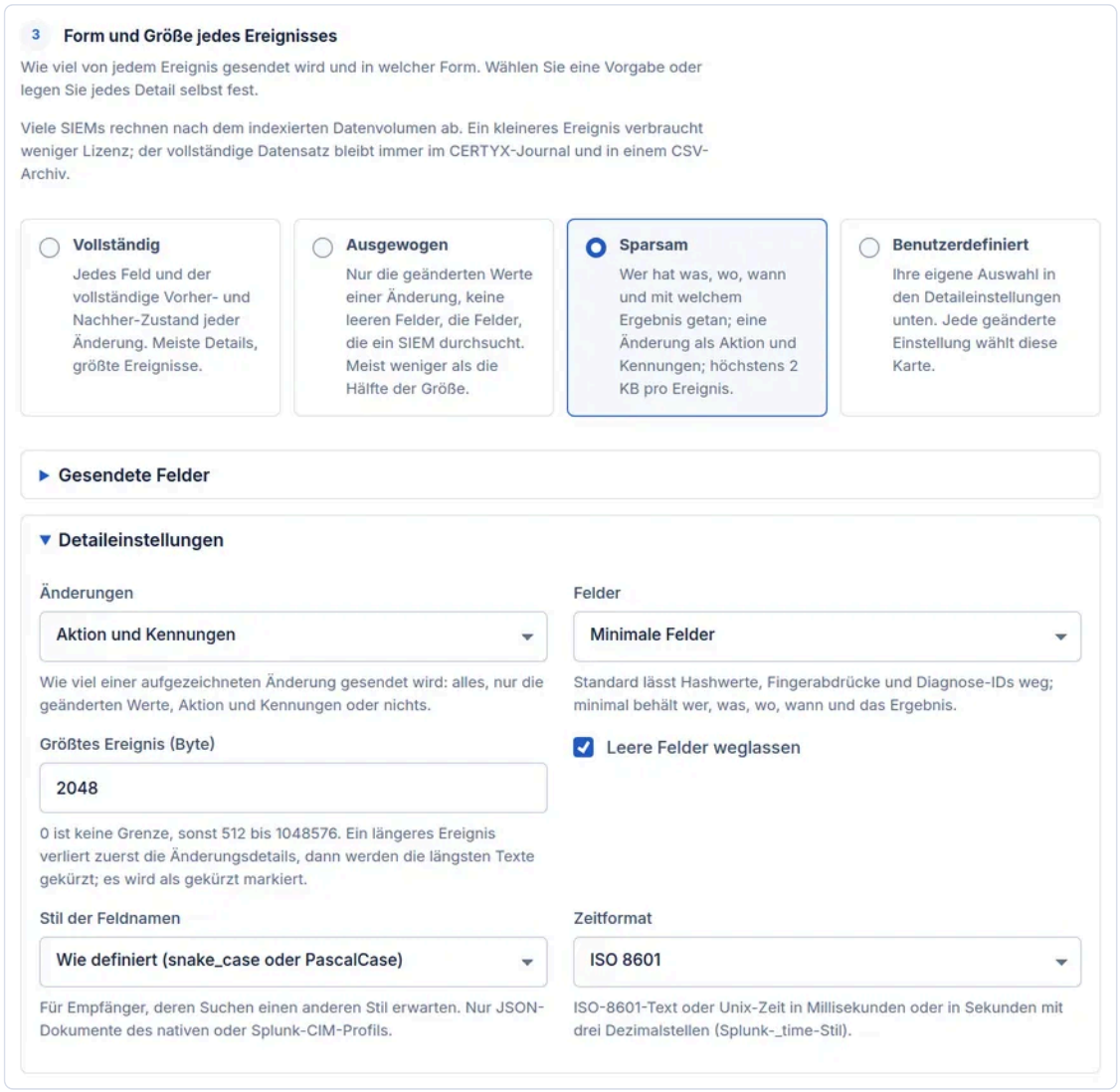
23.1 Wie die Filter zusammenwirken

1. Zustandsprüfungen werden entfernt (sofern sie nicht exportiert werden).
2. Ergebnisse (*oder* zwischen den ausgewählten) und Mindestschweregrad wählen die Ereignisse aus.
3. Kategorien und Ereigniscodes schränken dieses Ergebnis ein: Ein Ereignis passiert, wenn seine Kategorie **oder** sein Code aufgeführt ist. Ist keine der beiden Listen festgelegt, passiert jede Kategorie und jeder Code.
4. *Weitere Filter* entfernen den Rest: Routen außerhalb von *Nur diese Routen senden* sowie jede ausgeschlossene Route, jeden Akteur, jede Kategorie und jeden Ereigniscode.

Beispiel: Ergebnisse leer (alle), Mindestschweregrad 0, Kategorien `authentication`, Ereigniscodes `CONFIG-008` → jedes Anmeldeereignis und jede Änderung eines SIEM-Ziels.

Ausgelassene Ereignisse gehen nicht verloren: Die Position des Ziels rückt trotzdem über sie hinaus vor (das Journal behält sie), und ein Dateiarchiv desselben Bereichs behält sie. Eine Änderung des Filters gilt für Ereignisse, die ab dann exportiert werden; bereits exportierte Ereignisse werden nicht erneut gesendet, es sei denn, Sie senden erneut.

24 Ereignisgestaltung: Größe und Form



Ereignisgestaltung: die Vorgabe-Karten und die Detaileinstellungen eines Ziels (Beispieldaten)

Viele SIEM- und Log-Dienste rechnen nach dem indexierten Datenvolumen ab. *Nur senden* entscheidet, welche Ereignisse ein Ziel erhält; die **Ereignisgestaltung** entscheidet, wie groß und in welcher Form jedes davon ankommt. Jedes Ziel hat seine eigene Gestaltung, sodass ein SIEM kompakte Ereignisse und ein anderes alles erhalten kann. Der vollständige Datensatz bleibt immer im CERTYX-Journal und in einem Dateiarchiv; die Gestaltung ändert die Hash-Kette nie.

Vorgaben (Karten in Abschnitt 3 des Formulars):

Vorgabe	Änderungen	Felder	Leere Felder	Größtes Ereignis
Vollständig (wie vor der Gestaltung)	vollständiger Vorher- und Nachher-Zustand	alle	gesendet	keine Grenze
Ausgewogen	nur die geänderten Werte	Standard	weggelassen	keine Grenze
Sparsam	Aktion und Kennungen	minimal	weggelassen	2 KB
Benutzerdefiniert	Ihre Wahl	Ihre Wahl	Ihre Wahl	Ihre Wahl

Eine Karte füllt die **Detaileinstellungen** mit ihren Werten; jede Änderung daran wählt **Benutzerdefiniert**. Gemessen an einer Agentenbeobachtung von drei Zertifikaten im Format `json`: Vollständig 4.578 Byte, Ausgewogen 591 Byte (−87 %), Sparsam 378 Byte (−92 %); ein einfacher Agentenbericht: 706, 462 und 390 Byte. **Vorschau des Testereignisses** zeigt die Größe des Testereignisses mit und ohne Gestaltung.

Detaileinstellungen:

- **Änderungen** – *Vollständiger Vorher- und Nachher-Zustand*; *Nur die geänderten Werte* (`before` und `after` werden durch `changed` ersetzt, zum Beispiel `{"certificate.Owner":{"before":"Unassigned","after":"Team A"}}`; verschachtelte Objekte als Pfade mit Punkten); *Aktion und Kennungen* (die einfachen Werte einer Änderung, jedes Objekt und jede Liste

weggelassen); *Keine Änderungen*.

- **Felder** – *Alle Felder*; *Standardfelder* (lässt Hashwerte, den Agenten-Fingerabdruck, Span- und Request-IDs, Host-IP und Export-Verwaltung weg); *Minimale Felder* (wer hat was, wo, wann und mit welchem Ergebnis getan). Die Pflichtfelder des kanonischen Ereignisses (`schema_version`, `event_id`, `event_code`, `category`, `action`, `outcome`, `outcome_original`, `severity`, `severity_label`, `message`, die drei Zeiten, `application`, `service_name`, `organization_kind`) und `hostname` (es benennt die Instanz im Splunk-HEC-Umschlag, im Syslog-Header und in der OTLP-Ressource) werden immer gesendet, und ein Feld, das die Feldzuordnung behält oder hinzufügt, bleibt. Im `json`-Datensatz besteht die minimale Auswahl aus `Sequence`, `At`, `Category`, `EventCode`, `Tenant`, `Actor`, `SourceIp`, `Method`, `Route`, `Status`, `Outcome`, `EventId` und `SignIn`; Standard ergänzt `OrganizationName`, `Instance`, `TraceId`, `ErrorType` und `Changes`.
- **Leere Felder weglassen** – Felder ohne Wert (null, leerer Text, leere Liste oder leeres Objekt) werden nicht geschrieben.
- **Größtes Ereignis (Byte)** – 0 ist keine Grenze, sonst 512 bis 1.048.576. Ein längeres Ereignis behält zuerst nur Aktion und Kennungen seiner Änderungen, verliert dann seine Änderungen, dann werden seine längsten Texte gekürzt (endend mit `…`); es trägt dann `truncated: true` (`Truncated` im `json`-Datensatz). Die Größe wird am Ereignisdokument gemessen; Zeilenformate sind etwas kürzer oder länger.
- **Stil der Feldnamen** – *Wie definiert*, `snake_case`, `camelCase` oder `PascalCase` für die Felder der obersten Ebene (Werte in Änderungen und Anmeldedetails behalten ihre Namen). **Zeitformat** – ISO 8601, Unix-Zeit in Millisekunden oder in Sekunden mit drei Dezimalstellen (Splunk-`_time`-Stil). Beide gelten nur, wo ein JSON-Dokument des nativen oder Splunk-CIM-Profils geschrieben wird (`json`, `ndjson`, `rfc5424`); ECS, OCSF und die Zeilenformate behalten ihre eigenen Namen und Zeiten.
- **Syslog-MSGID** – *Ereigniskategorie* (Standard), *Ereigniscode*, *Eigener Text* (1–32 druckbare Zeichen ohne Leerzeichen, zum Beispiel `security-journal`, um vor 1.2.19 geschriebene Suchen zu behalten) oder *Keine* (`-`); **Syslog-APP-NAME** – leer sendet `clm`. Nur Syslog-Ziele haben sie.

Die Gestaltung gilt für jedes Netzwerkziel und jedes Format; ein Dateiarchiv behält immer den vollständigen Datensatz. Eine geänderte Gestaltung gilt für die danach exportierten Ereignisse. Die Karte des gespeicherten Ziels nennt die Gestaltung, zum Beispiel *Sparsam · Ausschlüsse: 1 · MSGID Ereigniscode*.

25 Feldzuordnung

Ein Netzwerkziel mit kanonischem Format (`ndjson`, `rfc5424`, `cef`, `leef-1.0`, `leef-2.0`, `kv`, `otlp-protobuf`, `otlp-json`) kann die Felder des kanonischen Ereignisses zuordnen, bevor ein Format sie schreibt. Die Regeln gehören zum Ziel: mit ihm gespeichert, mit ihm versioniert (ein Speichern mit veralteter Version wird abgelehnt) und mit dem Ziel vor und nach der Änderung protokolliert (`siem.destination.updated`).

Regel	Wirkung
Umbenennen (`rename`)	Verschiebt ein kanonisches Feld unter einen neuen Namen (ein eigenes Feld)
Entfernen (`drop`)	Entfernt ein kanonisches Feld (Negativliste)
Behalten (`keep`)	Positivliste: mit mindestens einer Behalten-Regel wird jedes optionale Feld entfernt, das weder behalten noch umbenannt wird
Konstante hinzufügen (`add`)	Fügt ein eigenes Feld mit festem Text von höchstens 256 Zeichen hinzu

Behalten-Regeln gelten zuerst, danach Umbenennen, Entfernen und Hinzufügen in ihrer Reihenfolge. Die Felder `schema_version`, `event_id`, `event_code`, `category`, `action`, `outcome`, `outcome_original`, `severity`, `severity_label`, `message`, `timestamp`, `observed_timestamp`, `export_timestamp`, `application`, `service_name` und `organization_kind` werden nie entfernt oder umbenannt, sodass Identifikation, Zeit, Kategorie und Ergebnis in jedem Ereignis bleiben. Ein neuer Name hat 1–64 Zeichen, beginnt mit einem Buchstaben und verwendet Buchstaben, Ziffern, Unterstriche und Punkte (CEF: nur Buchstaben und Ziffern; ECS und OCSF: keine Punkte); er darf nicht mit `certyx` beginnen, keinen anderen neuen Namen wiederholen und keinem kanonischen Feld oder Schlüssel entsprechen, den das Format selbst schreibt (zum Beispiel `dst` in CEF oder `server.address` in OTLP). Höchstens 32 Regeln; jedes Feld kommt in höchstens einer Umbenennen- oder Entfernen-Regel vor.

Wo das Ergebnis erscheint: entfernte Felder fehlen in jedem Format (KV lässt ihre Schlüssel weg); eigene Felder folgen den Feldern des Formats – inline im JSON-Dokument (NDJSON, RFC-5424-MSG, HEC-`event`), unter `labels` in ECS (als Text), unter `unmapped` in OCSF, als Erweiterungen in CEF und Attribute in LEEF, als Schlüssel in KV und als Attribute des Log-Eintrags in OTLP.

Die indizierten HEC-Felder und die Splunk-CIM-Felder behalten ihre Standardnamen. Der JSON-Datensatz, die Sentinel-Zeilen, die Layouts der ersten Version und Dateiarchive (die das vollständige Ereignis behalten) haben keine Feldzuordnung.

Vorschau. Vorschau des Testereignisses sendet das Formular im aktuellen, noch nicht gespeicherten Zustand an den Server, der das Testereignis des Bereichs nach den Regeln im gewählten Transport, Format und in der gewählten Kodierung erzeugt: Webhook-Inhalt, HEC-Umschlag, Syslog-Nachricht, Sentinel-Zeile, OTLP-Anfrage (Protobuf in seiner OTLP/JSON-Kodierung gezeigt, mit der Größe des binären Inhalts) oder Archivzeile. Es wird nichts an den Empfänger gesendet und nichts gespeichert. Eine ungültige Regel wird mit dem Grund abgelehnt.

API: Der Inhalt beim Speichern trägt die Regeln in `fieldMapping`, zum Beispiel

```
"fieldMapping": [
  {"op": "rename", "field": "user_id", "to": "account"},
  {"op": "drop", "field": "host_ip"},
  {"op": "add", "to": "region", "value": "eu-central"}
]
```

und `POST /api/administration/siem/preview` (Organisation) oder `POST /platform/api/siem/{scope}/preview` (Plattform; ohne Step-up-Code, es ändert nichts) nimmt denselben Inhalt wie ein Speichern und antwortet `{transport, format, profile, eventId, contentType, payload, bytes, canonical}`.

Format `kv` mit den Regeln oben (user_id in `account` umbenannt, `host_ip` entfernt, `region` hinzugefügt), Testereignis:

```
<86>1 2026-10-19T04:20:37.893Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM"
swVersion="1.0.0"] schema_version=certyx.event/1.0 event_id=f4293b79-ca2c-4fd2-a9d5-6490b28aee56 event_code=INTEGRATION-001
category=integration action=siem.test outcome=success outcome_original=business severity=3 severity_label=low message="siem.test
by siem-test" timestamp=2026-10-19T04:20:37.893Z observed_timestamp=2026-10-19T04:20:37.893Z export_timestamp=2026-10-
19T04:20:37.893Z application="CERTYX CLM" version=1.0.0 environment=production hostname=certyx-node-1 service_name=clm
component=siem organization_id=platform organization_name=- organization_kind=platform username=- user_role=- session_id=-
authentication_method=- user_verified=- user_agent=- src_ip=- src_port=- http_method=- http_route=- http_status=- protocol=-
correlation_id=f4293b79-ca2c-4fd2-a9d5-6490b28aee56 request_id=f4293b79-ca2c-4fd2-a9d5-6490b28aee56 trace_id=- resource_type=-
resource_id=- change_type=- reason=- error_type=- sequence=- event_hash=- agent_fingerprint=- changes="
[{\"action\":\"siem.test\",\"requestedBy\":\"admin\"}]" changes_redacted=- dst_ip=- dst_port=- span_id=- account=siem-test
region=eu-central
```

26 Anreicherungsfelder

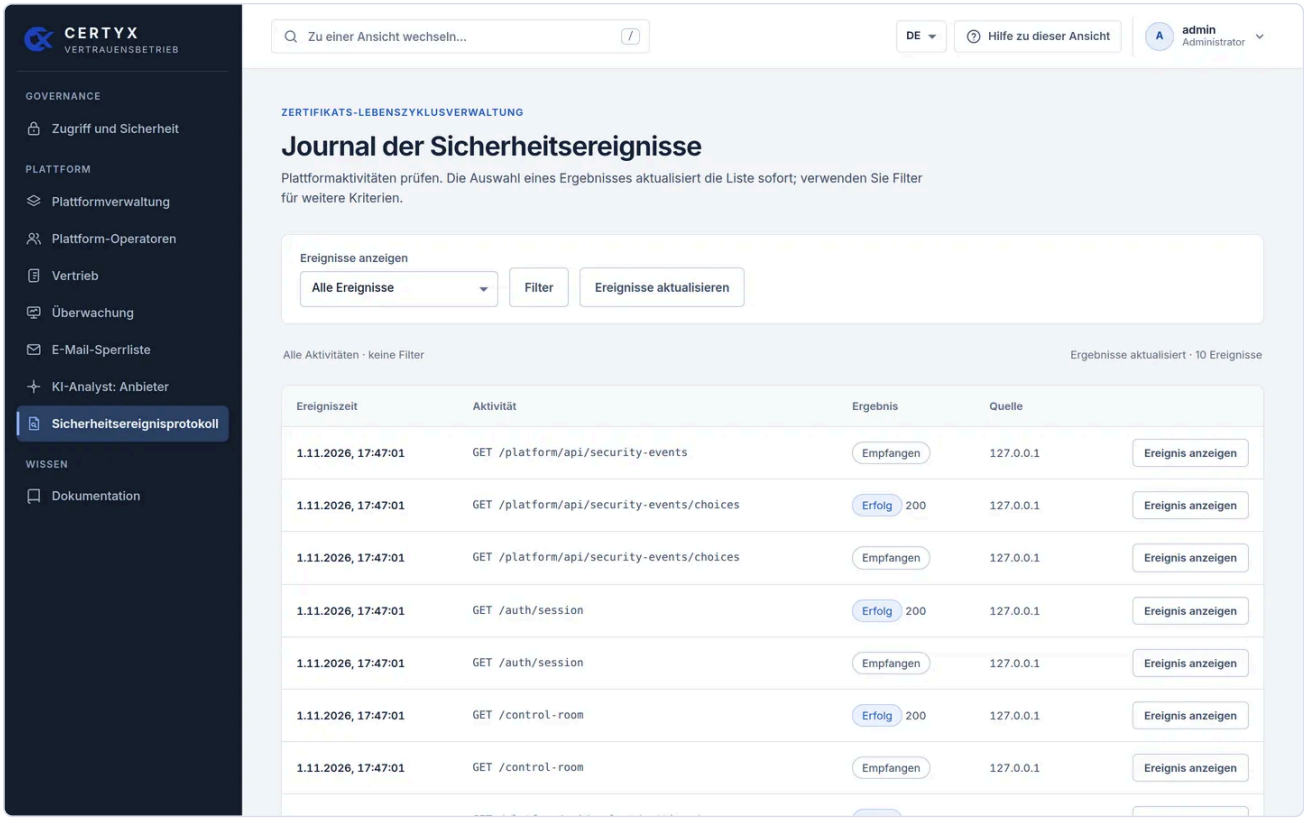
Seit diesem Release erfasst CERTYX zu jeder Anfrage diese Fakten neben dem Journaleintrag (in einer eigenen Tabelle mit der Sequenz als Schlüssel, außerhalb der Hashkette, sodass der Journal-Hash überprüfbar bleibt):

Feld	Inhalt	Hinweise
`trace_id`	W3C-Trace-ID der Anfrageaktivität	Ein eingehender `traceparent`-Header wird fortgesetzt, sodass die ID zum Trace des Aufrufers passt
`src_port`	Quellport des Clients	Hinter einem Proxy der Port des Proxys
`host_ip`	Serveradresse, die die Verbindung angenommen hat	Hinter einem Proxy oder in einem Container-Netzwerk die Adresse der lokalen Schnittstelle; RFC 5424 `origin ip`
`dst_port`	Serverport, der die Verbindung angenommen hat	`dst_ip` ist dieselbe Adresse wie `host_ip`; hinter einem Reverse Proxy oder in einem Container-Netzwerk der lokale Listener-Port (zum Beispiel 8080), nicht der öffentliche Port
`span_id`	W3C-Span-ID der Anfrageaktivität	Mit `trace_id` die TraceId und SpanId eines OTLP-Log-Eintrags
`user_role`	Rollen-Claim des angemeldeten Prinzipals	

Feld	Inhalt	Hinweise
`session_id`	Sitzungskorrelations-ID: bei einer serverseitigen Sitzung (Konsole oder mobile App) die ersten 32 hexadezimalen Zeichen von SHA-256 über `clm-session` und den SHA-256 der Sitzungskennung; bei einer älteren Browsersitzung von SHA-256 über `clm-session`, das Subjekt und den Anmeldezeitpunkt	Für jede Anfrage einer Sitzung gleich; nie das Sitzungscookie, die Sitzungskennung oder der gespeicherte Sitzungsschlüssel

Verfügbarkeit. Einträge, die vor diesem Release erfasst wurden, geschäftliche Einträge (Änderungen außerhalb einer Anfrage) und Testereignisse haben keine Anreicherung; ihre Felder sind unbekannt (in JSON ausgelassen, in CSV leer, in KV `-`). Anfragen von Agents und API-Schlüsseln sowie Sitzungen der mobilen App, die vor diesem Release begonnen wurden, tragen nur einen Teil davon (kein `session_id`). `dst_port` und `span_id` werden ab der OpenTelemetry-Version erfasst. Ein weitergeleiteter Host-Header wird für das Ziel nie verwendet, weil ein Client ihn setzen kann. Agenten- und Endpunktereignisse, die ein beobachtetes Ziel nennen (eine Discovery-Warnung nennt ihren Endpunkt `host:port`), übernehmen `dst_ip` und `dst_port` von diesem Ziel; ein Hostname ergibt keine `dst_ip`. Diese Felder erscheinen nur in den kanonischen Formaten und Profilen; die Layouts der ersten Version sind unverändert.

27 Mehrere Organisationen in einem SIEM



Journal der Sicherheitsereignisse im Kontrollraum, in dem die plattformweiten SIEM-Ziele verwaltet werden (Beispieldaten)

Jedes exportierte Ereignis weist seine Organisation aus, sodass sich mehrere Organisationen – oder der Betreiberbereich *Alle Ereignisse* – einen SIEM-Index teilen können. Die Felder werden beim Export ergänzt; der hashverkettete Journaleintrag selbst wird nicht verändert, sodass der Hash überprüfbar bleibt.

Bedeutung	Kanonisch	JSON-Datensatz / HEC `json`	CEF (`cef`)	LEEF	Sentinel	CSV (`csv`, `compat`)
Organisations-ID	`organization_id`	`Tenant`	`cs1`	`organization`	`Organization`	`OrganizationId`
Organisationsname	`organization_name`	`OrganizationName`	`cs2`	`organizationName`	`OrganizationName`	`OrganizationName`
Art: `organization`, `platform`, `unattributed`	`organization_kind`	`OrganizationKind`	—	`organizationKind`	`OrganizationKind`	`OrganizationKind`

Bedeutung	Kanonisch	JSON-Datensatz / HEC `json`	CEF (`cef`)	LEEF	Sentinel	CSV (`csv`, `csv-compat`)
CERTYX-Instanz	`hostname`	`Instance` (HEC auch `host`)	`dvchost`	`instance`	`Instance`	`Instance`

In `cef-compat` ist die Organisation `cs1`, und ihr Name und ihre Art sind `certyxOrganizationName` und `certyxOrganizationKind`. `platform` kennzeichnet Betreiberereignisse und fehlgeschlagene Plattformanmeldungen; `unattributed` kennzeichnet Anfragen ohne angemeldete Organisation (vor der Anmeldung, anonyme Aufrufe, Zustandsprüfungen). Der Name ist der Name der Organisation in der Kundenliste der Plattform; er ist leer (unbekannt), wenn die Installation keine Plattformverwaltung hat. Verwenden Sie für Zugriffsregeln und Deduplizierung die ID, nicht den Namen – ein Name kann sich ändern.

Splunk:

```
index=clm sourcetype=certyx:event organization_kind=organization
| stats count by organization_name, organization_id, category
```

Cribl, um Organisationen in einen eigenen Index zu leiten: ein *Eval* mit `index = 'clm_' + organization_id.substring(0, 8)` (oder ein Lookup von der Organisations-ID auf einen Index), platziert nach dem JSON-*Parser*. Die rollenbasierte Zugriffssteuerung von Splunk kann dann die Benutzer jeder Organisation auf deren Index beschränken.

28 Dateiarchive

Der Transport `file` schreibt jedes Ereignis seines Bereichs in Tagesdateien auf dem CERTYX-Server, statt es über das Netzwerk zu senden. Verwenden Sie ihn, um den vollständigen Sicherheitsbestand für eine Aufbewahrungsfrist außerhalb des SIEM zu behalten, und senden Sie dem SIEM nur, was es braucht (Nur senden).

Format × Profil	Dateiname	Inhalt
`csv-compat`	`certyx-security-YYYY-MM-DD.csv`	Die Spalten der ersten Version (unten)
`csv`	`certyx-events-YYYY-MM-DD.csv`	Kanonische Spalten in der festen Reihenfolge der Schemaversion, `schema_version` zuerst, `event_id` an zweiter Stelle
`ndjson` × `native` / `ecs` / `ocsf`	`certyx-events-YYYY-MM-DD.ndjson`	Ein Dokument des Profils pro Zeile

28.1 Einrichtung

1. **Organisation:** Ein mit MFA angemeldeter Admin öffnet **SIEM-Export**. **Plattform:** Der Betreiber öffnet **Journal der Sicherheitsereignisse** → **SIEM-Export** und wählt den Bereich *Plattformbetreiber* oder *Alle Ereignisse*.
2. Zielname und Kennung (zum Beispiel `csv-archive`), Transport **Dateiarchiv**, Format `csv-compat`, `csv` oder `ndjson` (mit seinem Profil).
3. **Aufbewahrung (Tage)**: 1–3650. Ältere Tagesdateien werden automatisch gelöscht.
4. Optional **Abgeschlossene Tage mit gzip komprimieren** (`.csv.gz`, `.ndjson.gz`).
5. Speichern, dann **Verbindung prüfen**: Die Prüfung stellt fest, ob das Archivverzeichnis beschreibbar ist, und zeigt die Anzahl der Dateien, ihren Platzbedarf und den freien Speicherplatz auf dem Volume.

Ein Archiv behält jedes Ereignis: Es hat keinen Ergebnis-, Schweregrad-, Kategorie- oder Codefilter, keine Zustellwerte, kein Clientzertifikat und kein Testereignis. Jedes Archiv zählt zum Ziellimit seines Bereichs.

Speicherort. Die Organisation kann keinen Pfad wählen: Ihre Dateien liegen unterhalb eines Verzeichnisses, das der Plattformbetreiber konfiguriert, `<Clm:Siem:ArchiveRoot>/organizations/<organization ID>/<archive identifier>/` (Standard für `ArchiveRoot`: `siem-archive` im CERTYX-Datenverzeichnis). Plattformweite Archive liegen in `<Clm:Siem:PlatformArchiveDirectory>/<platform|all>/<archive identifier>/` (Standard `<ArchiveRoot>/platform`). Verzeichnisse und Dateien werden so angelegt, dass nur das CERTYX-Dienstkonto sie lesen kann; ein Pfad, der einen symbolischen Link enthält, wird abgelehnt.

28.2 Dateien

- Eine Datei pro UTC-Tag, an dem Ereignisse geschrieben wurden. Die erste Datei eines neuen Archivs enthält den aufbewahrten Journalverlauf, danach folgen die Ereignisse jedes Tages.
- **CSV (`csv-compat`, `csv`)**: RFC 4180, UTF-8 ohne Byte Order Mark, eine Kopfzeile, Komma als Trennzeichen, CRLF-Zeilenenden, Anführungszeichen verdoppelt. Steuerzeichen innerhalb eines Werts werden zu Leerzeichen (`csv-compat`) oder bleiben innerhalb der Anführungszeichen (`csv`), sodass jeder Datensatz genau ein CSV-Datensatz ist. `csv-compat` setzt jede Zelle in Anführungszeichen. `csv` unterscheidet unbekannt von leer: Ein unbekannter Wert ist ein leeres Feld ohne Anführungszeichen, ein bekannter leerer Wert ist `""`.
- **Sicher gegen Formelinjektion:** Einer Zelle, die mit `=`, `+`, `-`, `@` (auch nach Leerzeichen), einem Tabulator oder einem Wagenrücklauf beginnt, wird `'` vorangestellt, sodass Tabellenkalkulationen sie als Text anzeigen.
- Spalten von `csv-compat`: `Sequence`, `At` (UTC), `OrganizationId`, `OrganizationName`, `OrganizationKind`, `Instance`, `Actor`, `SourceIp`, `AgentFingerprint`, `Method`, `Route`, `Status`, `Outcome`, `Severity`, `TraceId`, `ErrorType`, `SignInAction`, `SignInUser`, `SignInMethod`, `SignInResult`, `SignInReason`, `Changes` (JSON-Array), `PreviousHash`, `Hash`.
- Spalten von `csv`: die kanonischen Felder in der Reihenfolge der Feldreferenz; ein neues Feld wird nur mit einer neuen Schemaversion und nur am Ende angefügt.
- **NDJSON (`ndjson`)**: UTF-8 ohne Byte Order Mark, ein JSON-Dokument pro Zeile, mit LF abgeschlossen, keine Kopfzeile.
- **Versiegelung und Integrität.** Nach Ende des UTC-Tages wird die Datei des Tages beim nächsten Exportdurchlauf versiegelt: bei eingeschalteter Komprimierung mit gzip komprimiert, und ihr SHA-256 wird dem Manifest `SHA256SUMS` im selben Verzeichnis hinzugefügt. Prüfen Sie ein heruntergeladenes Archiv mit `sha256sum -c SHA256SUMS` (die Datei des laufenden Tages steht erst nach ihrer Versiegelung im Manifest). In `csv-compat` verknüpfen `Hash` und `PreviousHash` jede Zeile mit der Journalkette, die unter **Journal der Sicherheitsereignisse → Integrität des Journals prüfen** geprüft wird; die kanonischen Formate enthalten `event_hash`.
- **Aufbewahrung.** Bei jedem Exportdurchlauf werden Dateien von Tagen, die älter als die Aufbewahrungsfrist sind, samt ihren Manifestzeilen gelöscht; jede Löschung wird als `siem.archive.expired` (Code `INTEGRATION-009`) mit den Dateinamen im Journal erfasst (Akteur `system:siem-archive`).

28.3 Herunterladen

Auf der Karte des Archivs listet **Archivdateien** jede Datei mit ihrer Größe, der Angabe, ob sie versiegelt ist, und ihrem SHA-256 auf. **Herunterladen** lädt eine Datei, **Alle herunterladen (ZIP)** alle Dateien samt Manifest. Downloads erfordern eine mit MFA angemeldete Sitzung (wie Änderungen) und werden für Sitzungen der mobilen App abgelehnt; jeder Download wird als `siem.archive.downloaded` (Code `DATA-003`) mit den Dateinamen im Journal erfasst. Zum Auflisten der Dateien genügt die Rolle Admin (Organisation) bzw. die Plattformverwaltung (Betreiber). Der Betreiber kann die Dateien auch direkt vom Archiv-Volume kopieren.

28.4 Zustellung, Neustarts und Speicherplatz

- Das Archiv verwendet dieselbe dauerhafte Position wie die Netzwerkziele: Die Zeilen werden geschrieben und auf den Datenträger übertragen, danach rückt die Position vor (mindestens einmal, nichts übersprungen). Nach einem Absturz zwischen beiden Schritten liest der nächste Durchlauf zuerst die letzte vollständige Zeile der offenen Datei (ihre Sequenz, aus `Sequence` oder aus der Ereignis-ID), entfernt eine durch den Absturz abgeschnittene Zeile und setzt danach fort, sodass eine Datei nie eine Zeile doppelt enthält. Eine Zeile kann nur dann in zwei Dateien vorkommen, wenn während eines solchen Absturzes der Tag gewechselt hat.
- Bis zu `Clm:Siem:BatchSize` Ereignisse (standardmäßig 200) pro Durchlauf, ein Durchlauf alle `Clm:Siem:Interval` (standardmäßig 30 Sekunden); ein neues Archiv holt den Verlauf in diesem Takt nach.
- **Das CSV-Archivverzeichnis ist nicht beschreibbar (…)** Planen Sie für `csv-compat` unkomprimiert etwa 0,5–1 KB pro Ereignis, für `csv` und `ndjson` mehr (mehr Felder); gzip verringert dies typischerweise auf ein Zehntel. Überwachen Sie das Volume – ein voller Datenträger oder ein fehlendes Volume lässt das Archiv mit *Das CSV-Archivverzeichnis ist nicht beschreibbar (…)* fehlschlagen, löst den Vorfall `siem-delivery` aus und hält die Position, sodass das Archiv fortsetzt, sobald wieder Speicherplatz frei ist.
- In einer Installation mit mehreren Instanzen muss jede Instanz dasselbe Archivverzeichnis einbinden (der Export läuft jeweils auf einer Instanz, aber nicht immer auf derselben).
- Das Entfernen des Archivziels beendet das Schreiben und löscht seine Position; die Dateien bleiben im Verzeichnis, bis der Betreiber sie löscht. Ein erneutes Hinzufügen mit derselben Kennung setzt im selben Verzeichnis fort.

29 Berechtigungen

Aktion	Ziele der Organisation	Plattformbereiche (*Plattformbetreiber*, *Alle Ereignisse*)
Ziele, Status und Liste der Archivdateien anzeigen	Rolle Admin	Plattformverwaltung
Verbindung prüfen, Testereignis senden	Admin mit einer MFA-Sitzung	Plattformbetreiber mit einer MFA-Sitzung
Speichern, bearbeiten, entfernen, neues Geheimnis	Admin mit einer MFA-Sitzung	Plattformbetreiber mit einer MFA-Sitzung und dem Step-up-Code des Betreibers
Erneut senden ab einer Sequenz oder einem Zeitpunkt	Admin mit einer MFA-Sitzung	Plattformbetreiber mit einer MFA-Sitzung und dem Step-up-Code des Betreibers
Archivdateien herunterladen	Admin mit einer MFA-Sitzung	Plattformbetreiber mit einer MFA-Sitzung

Sitzungen der mobilen App können nicht ändern, testen, erneut senden oder herunterladen. Jede Änderung, Prüfung, jeder Test, jedes erneute Senden, Überspringen und Herunterladen wird in das Sicherheitsjournal geschrieben (Codes `CONFIG-008`, `INTEGRATION-002`, `INTEGRATION-003`, `INTEGRATION-012`, `INTEGRATION-011`, `DATA-003`).

30 Fehlerbehebung

30.1 Vorfälle

Vorfall	Schweregrad	Bedeutung	Maßnahme
`siem-delivery`	Warnung; kritisch nach *Versuche bis zur kritischen Warnung* aufeinanderfolgenden Fehlern	Die letzte Zustellung eines Ziels ist fehlgeschlagen; der Grund steht auf der Karte (und nennt bei einem abgelehnten Ereignis die Ereignis-ID)	Führen Sie *Verbindung prüfen* aus; korrigieren Sie den Empfänger; bei einem abgelehnten Ereignis siehe Abgelehnte Ereignisse
`siem-backlog`	Warnung	Es warten mehr Ereignisse als die Grenze der Rückstandswarnung des Ziels; es geht nichts verloren	Beheben Sie den Zustellfehler oder erhöhen Sie Stapelgröße oder Grenze, wenn der Empfänger nur langsam ist

Beide Vorfälle lösen sich von selbst auf: `siem-delivery` nach der nächsten erfolgreichen Zustellung, `siem-backlog`, wenn der Rückstand unter die Grenze fällt.

30.2 Ergebnisse von Prüfung und Test

Code	Bedeutung	Maßnahme
`dns-failed`	Der CERTYX-Server kann den Hostnamen nicht auflösen	Korrigieren Sie den Namen oder tragen Sie ihn in dem DNS ein, das der CERTYX-Server verwendet; interne Namen müssen auf dem Server auflösbar sein, nicht nur auf Ihrem Rechner
`dns-private`	Der Host wird zu einer privaten, Loopback- oder Link-Local-Adresse aufgelöst, und der Betreiber hat Ziele von Organisationen auf öffentliche Adressen beschränkt	Verwenden Sie eine öffentliche Adresse oder bitten Sie den Betreiber, private Ziele zu erlauben (`Clm:Siem:AllowPrivateDestinations`)
`tcp-refused`	Auf dem Port lauscht nichts, oder eine Firewall weist die Verbindung ab	Aktivieren Sie den empfangenden Eingang auf diesem Port; prüfen Sie die Portnummer
`tcp-unreachable`	Keine Netzwerkroute vom CERTYX-Server zur Adresse	Prüfen Sie Routing und Firewalls zwischen Server und Empfänger

Code	Bedeutung	Maßnahme
`tcp-timeout`	Keine Antwort innerhalb der Zeitüberschreitung (standardmäßig 10 Sekunden)	Vermutlich verwirft eine Firewall die Verbindung; erlauben Sie TCP vom CERTYX-Server
`tls-timeout`	TCP verbunden, aber kein TLS-Handshake	Der Eingang ist reines TCP; aktivieren Sie TLS daran (Cribl: TLS-Einstellungen der Quelle; Splunk: `[tcp-ssl:<port>]` oder HEC mit Enable SSL)
`tls-handshake-failed`	Der Empfänger hat die Verbindung während des Handshakes geschlossen	Aktivieren Sie TLS am Eingang; bieten Sie TLS 1.2 oder 1.3 an; verlangt der Empfänger ein Clientzertifikat, fügen Sie eines hinzu (mTLS) oder prüfen Sie, ob es von einer CA ausgestellt ist, der der Empfänger vertraut
`tls-name-mismatch`	Das Zertifikat nennt den eingegebenen Host nicht	Geben Sie einen im Zertifikat genannten Hostnamen ein oder installieren Sie ein Zertifikat für diesen Namen (`SplunkServerDefaultCert` von Splunk passt nie)
`tls-expired`	Das Zertifikat des Empfängers liegt außerhalb seiner Gültigkeitsdauer	Erneuern Sie das Zertifikat des Empfängers
`tls-untrusted`	Das Zertifikat führt zu keinem Stammzertifikat im Zertifikatspeicher des Systems, oder es ist gesperrt (bei Hard-Fail: sein Sperrstatus ist unbekannt)	Fügen Sie das Stamm-CA-Zertifikat (PEM) unter Vertrauenswürdige Stammzertifikate ein; ersetzen Sie ein gesperrtes Zertifikat; machen Sie bei Hard-Fail die CRL-/OCSP-Adresse erreichbar oder wählen Sie Soft-Fail
`tls-untrusted-roots`	Das Zertifikat führt zu keinem der für dieses Ziel eingefügten Stammzertifikate oder besteht die Sperrprüfung nicht	Fügen Sie das Stammzertifikat ein, das das Zertifikat des Empfängers tatsächlich ausgestellt hat; zur Sperrung wie oben
`hec-refused`	Splunk hat das Ereignis mit einem HEC-Code abgelehnt	Siehe die Tabelle der HEC-Codes unten
`hec-status`	Splunk hat mit einem HTTP-Fehler ohne HEC-Code geantwortet (oft ein Proxy oder ein falscher Pfad)	Verwenden Sie `/services/collector/event` ; prüfen Sie dazwischenliegende Proxys
`hec-not-acknowledged`	Splunk hat den Stapel angenommen, die Indizierung aber nicht innerhalb von 30 Sekunden bestätigt	Prüfen Sie den Zustand des Indexers und Acknowledgement am Token; der Stapel wird erneut gesendet
`entra-refused`	Microsoft Entra hat die Anmeldedaten der Anwendung abgelehnt	Prüfen Sie Mandanten-ID, Anwendungs-ID (Client) und den Value des Geheimnisses (nicht seine ID); erneuern Sie ein abgelaufenes Geheimnis
`http-status`	Der Empfänger hat mit einem HTTP-Fehler geantwortet	400/422: Der Empfänger lehnt die Daten ab (siehe Abgelehnte Ereignisse); 401/403: Geben Sie Token oder Geheimnis erneut ein, prüfen Sie bei Sentinel die Rollenzuweisung; 404: falscher Pfad, falsche unveränderliche DCR-ID oder falscher Stream; 413: Der Stapel ist für den Empfänger zu groß – verringern Sie die Stapelgröße; 429/503: ausgelastet, wird mit Backoff wiederholt; 5xx: Prüfen Sie die Protokolle des Empfängers
`delivery-failed`	Die Verbindung oder das Schreiben ist nach der TLS-Stufe fehlgeschlagen	Siehe den angezeigten Grund (zum Beispiel *Das Ziel hat nicht rechtzeitig geantwortet.*)
`archive-unwritable`	Das Archivverzeichnis ist nicht beschreibbar	Bitten Sie den Betreiber, das Archiv-Volume, seine Berechtigungen und den freien Speicherplatz zu prüfen (`Clm:Siem:ArchiveRoot`)
`archive-link`	Der Archivpfad enthält einen symbolischen Link	Der Betreiber muss einen Archivpfad ohne symbolische Links konfigurieren

30.3 Empfänger zeigt Fragmente oder binäre Daten

Was Sie im SIEM sehen (`FRAG`)	Ursache	Maßnahme
Datensätze, die mit einer Zahl und einem Leerzeichen beginnen (`412 <84>1 2026-…`), eine an das Ende eines Datensatzes angehängte Zahl (`… }4`), in der Mitte abgeschnittene Datensätze (`<86>1 2026-10-11T1`)	CERTYX verwendet Oktettzählung, der Empfänger trennt an Zeilenvorschüben (Cribl-Syslog-Quelle mit *Octet count framing* No, dem Standard)	Stellen Sie an der Quelle *Octet count framing* auf Yes oder wählen Sie in CERTYX das Framing *Zeilenumbruch* – auf beiden Seiten dasselbe

Was Sie im SIEM sehen (`_raw`)	Ursache	Maßnahme
Binäre Ereignisse, die mit `\x16\x03` beginnen (oft als `\x16\x03\x01…` angezeigt) und den Hostnamen des Empfängers enthalten (zum Beispiel `cribl-worker1…`)	Am empfangenden Port ist TLS ausgeschaltet; er indiziert den TLS-ClientHello von CERTYX	Aktivieren Sie TLS an der Quelle bzw. am Eingang (Cribl: *TLS Settings → Enabled*); CERTYX sendet nie reines TCP
`` , `ï»¿` oder `` vor dem JSON einer `rfc5424` -Nachricht	Der Empfänger zeigt das UTF-8-Byte-Order-Mark (RFC 5424 MSG-UTF8) als Text an	Entfernen Sie es im Parser des Empfängers oder bitten Sie den Betreiber, `Clm:Siem:Syslog:ByteOrderMark=false` zu setzen
Ereignisse von `/health/ready` mit `Tenant unattributed` alle paar Sekunden	Ein Load Balancer prüft die Bereitschaft; Versionen vor diesem Release haben die Prüfungen exportiert	Aktuelle Versionen senden keine Zustandsprüfungen (`Clm:Siem:ExportHealthProbes` , standardmäßig aus)
Jede Anfrage erscheint zweimal (`attempt` , dann `success` / `denied` /…)	Unter *Nur senden* ist *Empfangene Anfragen* angekreuzt	Heben Sie die Auswahl auf; ohne angekreuztes Feld erhält das Ziel einen Datensatz pro Anfrage
Dasselbe Ereignis erscheint zweimal mit derselben Ereignis-ID	Zustellung mindestens einmal nach einer Zeitüberschreitung oder einem erneuten Senden	Deduplizieren Sie nach der Ereignis-ID (Deduplizierung)
Werte für `host` oder `source` wie `http:clm` oder die Adresse des Cribl-Workers	Von der weiterleitenden Pipeline gesetzt (zum Beispiel vom HEC-Output von Cribl), nicht von CERTYX	Verwenden Sie `hostname` / `Instance` und die Organisationsfelder aus dem Ereignis oder setzen Sie `host` mit einem Eval in Cribl
Splunk-Felder mit zwei identischen Werten (`event_id` , `action`)	Das Feld ist sowohl ein indiziertes HEC-Feld als auch Teil des JSON-Ereignisses	Siehe props.conf und Splunk CIM

30.4 Splunk-HEC-Codes

Code	Splunk-Text	Maßnahme
1	Token disabled	Aktivieren Sie das Token unter *HTTP Event Collector*
2	Token is required	Geben Sie das HEC-Token ein
3	Invalid authorization	Geben Sie den Tokenwert genau so ein, wie Splunk ihn anzeigt
4	Invalid token	Das gespeicherte Token ist Splunk unbekannt; geben Sie ein gültiges Token ein
5	No data	Wird als Ablehnung der Daten behandelt (Abgelehnte Ereignisse); prüfen Sie Proxys, die den Inhalt entfernen könnten
6	Invalid data format	Verwenden Sie `/services/collector/event` (nicht `/raw`); wird als Ablehnung der Daten behandelt
7	Incorrect index	Fügen Sie den Index zu den Allowed indexes des Tokens hinzu oder korrigieren Sie den Indexnamen
8	Internal server error	Prüfen Sie den Zustand des Splunk-Indexers
9	Server is busy	Splunk ist überlastet; die Zustellung wird mit Backoff wiederholt
10	Data channel is missing	Vermutlich entfernt ein Proxy den Header `X-Splunk-Request-Channel`
11	Invalid data channel	Deaktivieren und reaktivieren Sie Indexer Acknowledgement am Token
12, 13	Event field is required / cannot be blank	Verwenden Sie `/services/collector/event` ; wird als Ablehnung der Daten behandelt
14	ACK is disabled	Keine Maßnahme nötig; die Zustellung ist nicht betroffen
15	Error in handling indexed fields	Wird als Ablehnung der Daten behandelt; prüfen Sie die Grenzen für indizierte Felder in Splunk
16	Query string authorization is not enabled	Keine Maßnahme nötig; CERTYX verwendet Header-Autorisierung
18, 19, 20	HEC is unhealthy	Warteschlangen voll oder Acknowledgement nicht verfügbar; die Zustellung wird mit Backoff wiederholt

30.5 Zustellstatus auf der Zielkarte

Letzter Fehler	Maßnahme
Das Ziel war nicht erreichbar.	Führen Sie Verbindung prüfen aus; die Prüfung nennt die fehlschlagende Stufe
Die TLS-Prüfung des Ziels ist fehlgeschlagen.	Führen Sie Verbindung prüfen aus; siehe die TLS-Codes oben
Das Ziel hat nicht rechtzeitig geantwortet.	Der Empfänger hat nicht innerhalb von 20 Sekunden geantwortet; prüfen Sie seine Auslastung oder das Netzwerk
HTTP <status>	Siehe `http-status` oben
Splunk HEC antwortete mit HTTP <status> und Code <n> (…)	Siehe die Tabelle der HEC-Codes
Der Empfänger lehnt das Ereignis <event ID> ab: …	Siehe Abgelehnte Ereignisse
Microsoft Entra hat die Anmeldedaten der Anwendung abgelehnt.	Siehe `entra-refused`
Das gespeicherte Geheimnis kann nicht entschlüsselt werden; geben Sie es erneut ein.	Bearbeiten Sie das Ziel und geben Sie das Geheimnis erneut ein (der Schlüsselbund des Servers hat sich geändert)
Das Clientzertifikat oder sein privater Schlüssel kann nicht gelesen werden; geben Sie beide erneut ein.	Bearbeiten Sie das Ziel und fügen Sie das Clientzertifikat und seinen Schlüssel erneut ein
Das Ziel wird zu einer privaten, Loopback- oder Link-Local-Adresse aufgelöst; …	Siehe `dns-private`
Der Empfänger hat die TCP-Verbindung angenommen, aber keinen TLS-Handshake abgeschlossen …	Am empfangenden Eingang ist TLS ausgeschaltet; siehe Empfänger zeigt Fragmente oder binäre Daten
Das CSV-Archivverzeichnis ist nicht beschreibbar (…)	Bitten Sie den Betreiber, das Archiv-Volume, seine Berechtigungen und den freien Speicherplatz zu prüfen (`Clm:Siem:ArchiveRoot`)
Der CSV-Archivpfad enthält einen symbolischen Link; …	Der Betreiber muss einen Archivpfad ohne symbolische Links konfigurieren

Ein fehlgeschlagener Stapel wird nach der Backoff-Wartezeit erneut versucht und löst den Vorfall `siem-delivery` aus (Zustellung). Die Position rückt erst vor, wenn der Empfänger den Stapel angenommen hat.

30.6 Formular- und Berechtigungsmeldungen

- Jedes abgelehnte Feld nennt das erwartete Format, zum Beispiel *Syslog-Host: Geben Sie nur den Hostnamen ein und tragen Sie den Port (zum Beispiel 10514) im Feld Port ein.*
- *Format, Schemaprofil und Transport: Wählen Sie einen Transport und dann eines der dafür angebotenen Formate und Profile.* – die Kombination ist nicht zulässig (Zulässige Kombinationen).
- *LEEF-Trennzeichen: …* – nur `leef-2.0` deklariert ein Trennzeichen; TAB oder ein Satzzeichen außer `=`, dem senkrechten Strich und dem Backslash, oder sein hexadezimaler Code.
- *Nur senden (Kategorien)* / *Nur senden (Ereigniscodes)* – nur Kategorien aus der Liste; Codes aus dem Katalog, höchstens 200.
- *Versuche bis zur kritischen Warnung*, *Erste Wartezeit (Sekunden)*, *Längste Wartezeit (Sekunden)*, *Stapelgröße (Ereignisse)*, *Rückstandswarnung ab (Ereignisse)* – ganze Zahlen in den Bereichen unter Zustellung; die längste Wartezeit ist nicht kürzer als die erste.
- *Clientzertifikat: …* – ein PEM-Zertifikat mit einem passenden unverschlüsselten privaten PEM-Schlüssel, jeweils höchstens 64 KiB; ein neues Zertifikat benötigt seinen Schlüssel.
- *Ein Dateiarchiv behält jedes Ereignis: Zustellwiederholungen, das Überspringen abgelehnter Ereignisse und das Clientzertifikat gelten nicht.*
- *Erneut senden ab: …* – eine Sequenz (1 oder höher) oder ein Zeitpunkt, nicht beides, an oder vor der Position des Ziels; nicht für Dateiarchive.
- *Geben Sie das Geheimnis erneut ein, wenn Sie das Format ändern.* – ein gespeichertes Geheimnis gehört zu seinem Transport.
- *Das Ziel wurde zwischenzeitlich geändert, existiert bereits oder das Ziellimit ist erreicht.* – laden Sie die Ansicht neu; entfernen Sie ein ungenutztes Ziel, wenn das Limit erreicht ist.

- *Geben Sie das Geheimnis ein, bevor Sie ein Testereignis senden.* – Webhook, Splunk HEC und Sentinel benötigen zuerst ihr Geheimnis.
- Der MFA-Hinweis nennt den Grund: kein bestätigter Authenticator, Einrichtung nicht bestätigt, Sitzung vor MFA begonnen, Firmenanmeldung ohne gemeldete MFA oder reine Passwortsitzung. Folgen Sie der angezeigten Anweisung und melden Sie sich dann erneut an.

31 Beispielereignisse

Jedes Beispiel ist das Testereignis eines laufenden CERTYX-Servers in diesem Format und Profil (Ereigniscode `INTEGRATION-001`); installationsspezifische Werte weichen in Ihrer Installation ab.

Die Beispiele wurden am 19.10.2026 auf einem CERTYX-Server mit den Standard-Syslog-Einstellungen (Facility authpriv, APP-NAME `clm`, keine Enterprise-Nummer, daher kein Element `certyx@…`) und `Clm:Siem:Environment=production` aufgezeichnet. Syslog-Beispiele zeigen eine Nachricht ohne ihr Oktettzähler-Präfix; `<BOM>` steht für die UTF-8-Bytereihenfolgemarke. Ein Dateiarchiv erhält kein Testereignis, daher sind seine Beispiele die ersten Journaleinträge, die der Export geschrieben hat.

31.1 HTTPS-Webhook

Format `json` (JSON-Array von Datensätzen):


```
[{"Sequence":0,"At":"2026-10-19T04:20:31.2998647+00:00","Category":"integration","EventCode":"INTEGRATION-001","Tenant":"platform","OrganizationName":"","OrganizationKind":"platform","Instance":"certyx-node-1","Actor":"siem-test","SourceIp":"system","AgentFingerprint":"","Method":"INTERNAL","Route":"business/siem.test","Status":200,"Outcome":"business","TraceId":"ecb63792-0af6-4902-b5c1-4a8b409b82ce","ErrorType":"","Changes":[{"action":"siem.test","requestedBy":"admin"}],"Hash":"","EventId":"ecb63792-0af6-4902-b5c1-4a8b409b82ce"}]
```


Format `ndjson`, Profil `native`:


```
{"schema_version":"certyx.event/1.0","event_id":"c6030d2e-e437-4a96-a118-688cc908e6f9","event_code":"INTEGRATION-001","category":"integration","action":"siem.test","outcome":"success","outcome_original":"business","severity":3,"severity_label":"low","message":"siem.test by siem-test","timestamp":"2026-10-19T04:20:31.764Z","observed_timestamp":"2026-10-19T04:20:31.764Z","export_timestamp":"2026-10-19T04:20:31.764Z","application":"CERTYX CLM","version":"1.0.0","environment":"production","hostname":"certyx-node-1","service_name":"clm","component":"siem","organization_id":"platform","organization_kind":"platform","user_id":"siem-test","correlation_id":"c6030d2e-e437-4a96-a118-688cc908e6f9","request_id":"c6030d2e-e437-4a96-a118-688cc908e6f9","changes":[{"action":"siem.test","requestedBy":"admin"}]}
```


Format `ndjson`, Profil `ecs`:


```
{"@timestamp":"2026-10-19T04:20:32.164Z","message":"siem.test by siem-test","ecs":{"version":"8.11.0"},"event":{"id":"81e697f9-214e-46c7-9b22-785f48e900a6","code":"INTEGRATION-001","kind":"event","category":["network"],"type":["connection"],"action":"siem.test","outcome":"success","severity":3,"created":"2026-10-19T04:20:32.164Z","dataset":"certyx.security","provider":"siem"},"user":{"id":"siem-test"},"host":{"hostname":"certyx-node-1"},"service":{"name":"clm","version":"1.0.0","environment":"production"},"organization":{"id":"platform"},"labels":{"schema_version":"certyx.event/1.0","certyx_category":"integration"}}
```


Format `ndjson`, Profil `ocsf`:


```
{"class_uid":0,"category_uid":0,"activity_id":99,"type_uid":99,"severity_id":1,"status_id":1,"time":1792383632558,"message":"siem.test by siem-test","metadata":{"version":"1.1.0","uid":"d2bbe2f2-be7a-4963-903a-94a2c7790742","event_code":"INTEGRATION-001","log_name":"security-journal","product":{"name":"CLM","vendor_name":"CERTYX","version":"1.0.0"},"correlation_uid":"d2bbe2f2-be7a-4963-903a-94a2c7790742"},"unmapped":{"certyx_schema_version":"certyx.event/1.0","certyx_category":"integration","certyx_action":"siem.test"}}
```

31.2 Splunk HEC

Format `json` (Datensatz der ersten Version, indiziertes Feld `event_id`):


```
{"time":1792383633.006,"host":"certyx-node-1","source":"certyx:security-journal","sourcetype":"certyx:security","event":
{"Sequence":0,"At":"2026-10-19T04:20:33.0060384+00:00","Category":"integration","EventCode":"INTEGRATION-
001","Tenant":"platform","OrganizationName":"","OrganizationKind":"platform","Instance":"certyx-node-1","Actor":"siem-
test","SourceIp":"system","AgentFingerprint":"","Method":"INTERNAL","Route":"business/siem.test","Status":200,"Outcome":"business
","TraceId":"94ac6f46-f534-4fdc-95a5-ff0199cbc888","ErrorType":"","Changes":
[{"action":"siem.test","requestedBy":"admin"}],"Hash":"","EventId":"94ac6f46-f534-4fdc-95a5-ff0199cbc888"},"fields":
{"event_id":"94ac6f46-f534-4fdc-95a5-ff0199cbc888"}}
```


Format `ndjson`, Profil `native`:


```
{"time":1792383633.454,"host":"certyx-node-1","source":"certyx:security-journal","sourcetype":"certyx:event","event":
{"schema_version":"certyx.event/1.0","event_id":"cbd516bc-2c12-497d-909d-66417ae5b605","event_code":"INTEGRATION-
001","category":"integration","action":"siem.test","outcome":"success","outcome_original":"business","severity":3,"severity_label
":"low","message":"siem.test by siem-test","timestamp":"2026-10-19T04:20:33.454Z","observed_timestamp":"2026-10-
19T04:20:33.454Z","export_timestamp":"2026-10-19T04:20:33.454Z","application":"CERTYX
CLM","version":"1.0.0","environment":"production","hostname":"certyx-node-
1","service_name":"clm","component":"siem","organization_id":"platform","organization_kind":"platform","user_id":"siem-
test","correlation_id":"cbd516bc-2c12-497d-909d-66417ae5b605","request_id":"cbd516bc-2c12-497d-909d-66417ae5b605","changes":
[{"action":"siem.test","requestedBy":"admin"}]},"fields":{"event_id":"cbd516bc-2c12-497d-909d-
66417ae5b605","event_code":"INTEGRATION-001","category":"integration"}}
```


Format `ndjson`, Profil `ecs`:


```
{"time":1792383633.853,"host":"certyx-node-1","source":"certyx:security-journal","sourcetype":"certyx:event","event":
{"@timestamp":"2026-10-19T04:20:33.853Z","message":"siem.test by siem-test","ecs":{"version":"8.11.0"},"event":{"id":"4bdad006-
45f0-494e-a088-1857180b973c","code":"INTEGRATION-001","kind":"event","category":["network"],"type":
["connection"],"action":"siem.test","outcome":"success","severity":3,"created":"2026-10-
19T04:20:33.853Z","dataset":"certyx.security","provider":"siem"},"user":{"id":"siem-test"},"host":{"hostname":"certyx-node-
1"},"service":{"name":"clm","version":"1.0.0","environment":"production"},"organization":{"id":"platform"},"labels":
{"schema_version":"certyx.event/1.0","certyx_category":"integration"}},"fields":{"event_id":"4bdad006-45f0-494e-a088-
1857180b973c","event_code":"INTEGRATION-001","category":"integration"}}
```


Format `ndjson`, Profil `ocsf`:


```
{"time":1792383634.227,"host":"certyx-node-1","source":"certyx:security-journal","sourcetype":"certyx:event","event":
{"class_uid":0,"category_uid":0,"activity_id":99,"type_uid":99,"severity_id":1,"status_id":1,"time":1792383634227,"message":"siem
.test by siem-test","metadata":{"version":"1.1.0","uid":"05b4a4e8-c15b-439f-8051-d00980aca411","event_code":"INTEGRATION-
001","log_name":"security-journal","product":{"name":"CLM","vendor_name":"CERTYX","version":"1.0.0"},"correlation_uid":"05b4a4e8-
c15b-439f-8051-d00980aca411"},"unmapped":
{"certyx_schema_version":"certyx.event/1.0","certyx_category":"integration","certyx_action":"siem.test"}},"fields":
{"event_id":"05b4a4e8-c15b-439f-8051-d00980aca411","event_code":"INTEGRATION-001","category":"integration"}}
```


Format `ndjson`, Profil `splunk-cim`:


```
{"time":1792383634.591,"host":"certyx-node-1","source":"certyx:security-journal","sourcetype":"certyx:event","event":
{"schema_version":"certyx.event/1.0","event_id":"38cd7f4b-5f1b-44c3-909c-695c0c9caf3c","event_code":"INTEGRATION-
001","category":"integration","action":"siem.test","outcome":"success","outcome_original":"business","severity":3,"severity_label
":"low","message":"siem.test by siem-test","timestamp":"2026-10-19T04:20:34.591Z","observed_timestamp":"2026-10-
19T04:20:34.591Z","export_timestamp":"2026-10-19T04:20:34.591Z","application":"CERTYX
CLM","version":"1.0.0","environment":"production","hostname":"certyx-node-
1","service_name":"clm","component":"siem","organization_id":"platform","organization_kind":"platform","user_id":"siem-
test","correlation_id":"38cd7f4b-5f1b-44c3-909c-695c0c9caf3c","request_id":"38cd7f4b-5f1b-44c3-909c-695c0c9caf3c","changes":
```

[{"action":"siem.test","requestedBy":"admin"}]},"fields":{"event_id":"38cd7f4b-5f1b-44c3-909c-695c0c9caf3c","event_code":"INTEGRATION-001","category":"integration","vendor_product":"CERTYX CLM","app":"clm","signature":"siem.test by siem-test","signature_id":"INTEGRATION-001","severity":"informational","severity_id":"3","dest":"certyx-node-1","user":"siem-test","user_id":"siem-test","action":"siem.test"}}

31.3 Syslog über TLS

Format `json` (erste Version, fester Header):

<86>1 2026-10-19T04:20:34.960Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"] {"Sequence":0,"At":"2026-10-19T04:20:34.9600107+00:00","Category":"integration","EventCode":"INTEGRATION-001","Tenant":"platform","OrganizationName":"","OrganizationKind":"platform","Instance":"certyx-node-1","Actor":"siem-test","SourceIp":"system","AgentFingerprint":"","Method":"INTERNAL","Route":"business/siem.test","Status":200,"Outcome":"business","TraceId":"e4828ba9-d4aa-46ef-bff7-82465c8f9598","ErrorType":"","Changes":[{"action":"siem.test","requestedBy":"admin"}],"Hash":"","EventId":"e4828ba9-d4aa-46ef-bff7-82465c8f9598"}

Format `rfc5424`, Profil `native`:

<86>1 2026-10-19T04:20:35.330Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM" swVersion="1.0.0"] <BOM>{"schema_version":"certyx.event/1.0","event_id":"75620e9f-9fe1-4f59-84d4-5c1263f6c1fe","event_code":"INTEGRATION-001","category":"integration","action":"siem.test","outcome":"success","outcome_original":"business","severity":3,"severity_label":"low","message":"siem.test by siem-test","timestamp":"2026-10-19T04:20:35.330Z","observed_timestamp":"2026-10-19T04:20:35.330Z","export_timestamp":"2026-10-19T04:20:35.330Z","application":"CERTYX CLM","version":"1.0.0","environment":"production","hostname":"certyx-node-1","service_name":"clm","component":"siem","organization_id":"platform","organization_kind":"platform","user_id":"siem-test","correlation_id":"75620e9f-9fe1-4f59-84d4-5c1263f6c1fe","request_id":"75620e9f-9fe1-4f59-84d4-5c1263f6c1fe","changes":[{"action":"siem.test","requestedBy":"admin"}]}

Format `rfc5424`, Profil `ecs`:

<86>1 2026-10-19T04:20:35.699Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM" swVersion="1.0.0"] <BOM>{"@timestamp":"2026-10-19T04:20:35.699Z","message":"siem.test by siem-test","ecs":{"version":"8.11.0"},"event":{"id":"ae9bc3c0-bc81-4d1e-9540-11fc3940029b","code":"INTEGRATION-001","kind":"event","category":["network"],"type":["connection"],"action":"siem.test","outcome":"success","severity":3,"created":"2026-10-19T04:20:35.699Z","dataset":"certyx.security","provider":"siem"},"user":{"id":"siem-test"},"host":{"hostname":"certyx-node-1"},"service":{"name":"clm","version":"1.0.0","environment":"production"},"organization":{"id":"platform"},"labels":{"schema_version":"certyx.event/1.0","certyx_category":"integration"}}

Format `rfc5424`, Profil `ocsf`:

<86>1 2026-10-19T04:20:36.066Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM" swVersion="1.0.0"] <BOM>{"class_uid":0,"category_uid":0,"activity_id":99,"type_uid":99,"severity_id":1,"status_id":1,"time":1792383636066,"message":"siem.test by siem-test","metadata":{"version":"1.1.0","uid":"f4dbd1be-ac9d-49b1-b952-07226fedb56e","event_code":"INTEGRATION-001","log_name":"security-journal","product":{"name":"CLM","vendor_name":"CERTYX","version":"1.0.0"},"correlation_uid":"f4dbd1be-ac9d-49b1-b952-07226fedb56e"},"unmapped":{"certyx_schema_version":"certyx.event/1.0","certyx_category":"integration","certyx_action":"siem.test"}}

Format `cef`:

<86>1 2026-10-19T04:20:36.426Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM" swVersion="1.0.0"] CEF:0|CERTYX|CLM|1.0.0|INTEGRATION-001|siem.test by siem-test|3|rt=1792383636426 externalId=1ff31932-0574-411b-ae54-e5198a04aea0 act=siem.test cat=integration outcome=success msg=siem.test by siem-test suser=siem-test suid=siem-test

```
dvchost=certyx-node-1 cs1Label=organizationId cs1=platform cs3Label=correlationId cs3=1ff31932-0574-411b-ae54-e5198a04aea0
cs6Label=schemaVersion cs6=certyx.event/1.0
```

Format `leef-1.0`:

```
<86>1 2026-10-19T04:20:36.792Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM"
swVersion="1.0.0"] LEEF:1.0|CERTYX|CLM|1.0.0|INTEGRATION-001|devTime=2026-10-19T04:20:36.792+0000 devTimeFormat=yyyy-MM-
dd'T'HH:mm:ss.SSSZ cat=integration sev=3 usrName=siem-test eventId=cc2733fd-b537-4cec-98f2-80c0acf4e3c6
schemaVersion=certyx.event/1.0 action=siem.test outcome=success message=siem.test by siem-test userId=siem-test
organization=platform organizationKind=platform correlationId=cc2733fd-b537-4cec-98f2-80c0acf4e3c6
instance=certyx-node-1
```

Format `leef-2.0` mit dem Zieltrennzeichen `^` (ohne Trennzeichen wird TAB verwendet, deklariert als `x09`):

```
<86>1 2026-10-19T04:20:37.152Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM"
swVersion="1.0.0"] LEEF:2.0|CERTYX|CLM|1.0.0|INTEGRATION-001|^|devTime=2026-10-19T04:20:37.152+0000^devTimeFormat=yyyy-MM-
dd'T'HH:mm:ss.SSSZ^cat=integration^sev=3^usrName=siem-test^eventId=4ead979d-6b67-496c-931a-
d0c0ff1fc02a^schemaVersion=certyx.event/1.0^action=siem.test^outcome=success^message=siem.test by siem-test^userId=siem-
test^organization=platform^organizationKind=platform^correlationId=4ead979d-6b67-496c-931a-d0c0ff1fc02a^instance=certyx-node-1
```

Format `kv` (`certyx-kv/1.0`):

```
<86>1 2026-10-19T04:20:37.523Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"][origin software="CERTYX CLM"
swVersion="1.0.0"] schema_version=certyx.event/1.0 event_id=acbe1e97-a048-43f7-b350-ae2dba0f2dfb event_code=INTEGRATION-001
category=integration action=siem.test outcome=success outcome_original=business severity=3 severity_label=low message="siem.test
by siem-test" timestamp=2026-10-19T04:20:37.523Z observed_timestamp=2026-10-19T04:20:37.523Z export_timestamp=2026-10-
19T04:20:37.523Z application="CERTYX CLM" version=1.0.0 environment=production hostname=certyx-node-1 host_ip=- service_name=clm
component=siem organization_id=platform organization_name=- organization_kind=platform user_id=siem-test username=- user_role=-
session_id=- authentication_method=- user_verified=- user_agent=- src_ip=- src_port=- http_method=- http_route=- http_status=-
protocol=- correlation_id=acbe1e97-a048-43f7-b350-ae2dba0f2dfb request_id=acbe1e97-a048-43f7-b350-ae2dba0f2dfb trace_id=-
resource_type=- resource_id=- change_type=- reason=- error_type=- sequence=- event_hash=- agent_fingerprint=- changes="
[{\"action\":\"siem.test\",\"requestedBy\":\"admin\"}]" changes_redacted=- dst_ip=- dst_port=- span_id=-
```

Format `cef-compat` (erste Version):

```
<86>1 2026-10-19T04:20:38.263Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"]
CEF:0|CERTYX|CLM|1.0.0|certyx:business|business INTERNAL business/siem.test|3|rt=1792383638263 suser=siem-test
requestMethod=INTERNAL request=business/siem.test outcome=business externalId=0 dvchost=certyx-node-1 cn1Label=httpStatus cn1=200
cs1Label=organization cs1=platform certyxOrganizationName= certyxOrganizationKind=platform cs2Label=traceId cs2=0d0b9f13-888d-
4f45-bca4-ec48d749f919 cs3Label=eventHash cs3= cs4Label=agentFingerprint cs4= cs5Label=errorType cs5= cs6Label=changes cs6=
[{"action":"siem.test","requestedBy":"admin"}] certyxEventId=0d0b9f13-888d-4f45-bca4-ec48d749f919
```

Format `leef-compat` (erste Version):

```
<86>1 2026-10-19T04:20:38.733Z certyx-node-1 clm 1 integration [timeQuality tzKnown="1"]
LEEF:2.0|CERTYX|CLM|1.0.0|certyx:business|^|devTime=2026-10-19T04:20:38.733+0000^devTimeFormat=yyyy-MM-
dd'T'HH:mm:ss.SSSZ^usrName=siem-
test^cat=business^sev=3^method=INTERNAL^resource=business/siem.test^httpStatus=200^organization=platform^organizationName=^organi
zationKind=platform^traceId=2938c9ab-5183-4e4f-81fa-
deab9119266a^sequence=0^eventHash=^agentFingerprint=^errorType=^instance=certyx-node-1^changes=
[{"action":"siem.test","requestedBy":"admin"}]^eventId=2938c9ab-5183-4e4f-81fa-deab9119266a
```

31.4 Microsoft Sentinel

Format `sentinel` (Zeile für Logs Ingestion):

```
[{"TimeGenerated":"2026-10-
19T04:20:39.0967775Z","Sequence":0,"Organization":"platform","OrganizationName":"","OrganizationKind":"platform","Actor":"siem-
test","SourceIp":"system","AgentFingerprint":"","Method":"INTERNAL","Route":"business/siem.test","Status":200,"Outcome":"business
","TraceId":"70b73efa-1bea-4dfe-9e68-d72ba95b82a7","ErrorType":"","Instance":"certyx-node-1","EventHash":"","Changes":
[{"action":"siem.test","requestedBy":"admin"}],"User":"","UserVerified":false,"UserOrganization":"","AuthAction":"","AuthMethod":
"","AuthResult":"","Reason":"","UserAgent":"","EventId":"70b73efa-1bea-4dfe-9e68-d72ba95b82a7"}]
```


31.5 OpenTelemetry Logs

Format `otlp-json`, Inhalt des Log-Eintrags *Nachricht* (Testereignis):


```
{"resourceLogs":[{"resource":{"attributes":[{"key":"service.name","value":{"stringValue":"certyx"}},
{"key":"service.version","value":{"stringValue":"1.0.0"}},{"key":"deployment.environment.name","value":
{"stringValue":"production"}},{"key":"host.name","value":{"stringValue":"certyx-node-1"}},
{"key":"certyx.organization.id","value":{"stringValue":"platform"}},{"key":"certyx.organization.kind","value":
{"stringValue":"platform"}}]},"scopeLogs":[{"scope":{"name":"certyx.security-journal","version":"1.0.0"},"logRecords":
[{"timeUnixNano":"1792838639470357800","observedTimeUnixNano":"1792838639470357800","severityNumber":9,"severityText":"INFO","bod
y":{"stringValue":"siem.test by siem-test"},"attributes":[{"key":"certyx.schema_version","value":
{"stringValue":"certyx.event/1.0"}},{"key":"certyx.event.id","value":{"stringValue":"eaf0e504-9ff5-4214-8758-5eebb3e5b907"}},
{"key":"certyx.event.code","value":{"stringValue":"INTEGRATION-001"}},{"key":"certyx.event.category","value":
{"stringValue":"integration"}},{"key":"event.name","value":{"stringValue":"siem.test"}},{"key":"certyx.event.outcome","value":
{"stringValue":"success"}},{"key":"certyx.event.outcome_original","value":{"stringValue":"business"}},
{"key":"certyx.event.severity","value":{"intValue":"3"}},{"key":"certyx.event.severity_label","value":{"stringValue":"low"}},
{"key":"certyx.export_timestamp","value":{"stringValue":"2026-10-19T04:20:39.470Z"}},{"key":"certyx.application","value":
{"stringValue":"CERTYX CLM"}},{"key":"certyx.service_name","value":{"stringValue":"clm"}},{"key":"certyx.component","value":
{"stringValue":"siem"}},{"key":"user.id","value":{"stringValue":"siem-test"}},{"key":"certyx.correlation_id","value":
{"stringValue":"eaf0e504-9ff5-4214-8758-5eebb3e5b907"}},{"key":"certyx.request_id","value":{"stringValue":"eaf0e504-9ff5-4214-
8758-5eebb3e5b907"}},{"key":"certyx.changes","value":{"stringValue":"
[{\"action\":\"siem.test\",\"requestedBy\":\"admin\"}]"}}]}]}]}]}
```


Ein Anfrageeintrag des Journals (der `POST` eines Tests, mit Trace- und Span-ID, Client- und Serveradresse), vom Exportdurchlauf über `otlp-json` exportiert:


```
{"resourceLogs":[{"resource":{"attributes":[{"key":"service.name","value":{"stringValue":"certyx"}},
{"key":"service.version","value":{"stringValue":"1.0.0"}},{"key":"deployment.environment.name","value":
{"stringValue":"production"}},{"key":"host.name","value":{"stringValue":"certyx-node-1"}},{"key":"host.ip","value":{"arrayValue":
{"values":[{"stringValue":"127.0.0.1"}]}}},{"key":"certyx.organization.id","value":{"stringValue":"default"}},
{"key":"certyx.organization.kind","value":{"stringValue":"organization"}}]},"scopeLogs":[{"scope":{"name":"certyx.security-
journal","version":"1.0.0"},"logRecords":
[{"timeUnixNano":"1792838631369104500","observedTimeUnixNano":"1792838631369104500","severityNumber":9,"severityText":"INFO","bod
y":{"stringValue":"POST /platform/api/siem/{scope}/{id}/test 200 success"},"attributes":[{"key":"certyx.schema_version","value":
{"stringValue":"certyx.event/1.0"}},{"key":"certyx.event.id","value":{"stringValue":"e828c2f0-1089-48fe-838a-9c5267fb560b:31"}},
{"key":"certyx.event.code","value":{"stringValue":"INTEGRATION-002"}},{"key":"certyx.event.category","value":
{"stringValue":"integration"}},{"key":"event.name","value":{"stringValue":"siem.destination.tested"}},
{"key":"certyx.event.outcome","value":{"stringValue":"success"}},{"key":"certyx.event.outcome_original","value":
{"stringValue":"success"}},{"key":"certyx.event.severity","value":{"intValue":"3"}},{"key":"certyx.event.severity_label","value":
{"stringValue":"low"}},{"key":"certyx.export_timestamp","value":{"stringValue":"2026-10-19T04:20:41.828Z"}},
{"key":"certyx.application","value":{"stringValue":"CERTYX CLM"}},{"key":"certyx.service_name","value":{"stringValue":"clm"}},
{"key":"certyx.component","value":{"stringValue":"platform"}},{"key":"user.id","value":
{"stringValue":"7e636f39d1f54da2a7b53c583a2c16ab"}},{"key":"user.roles","value":{"arrayValue":{"values":
[{"stringValue":"Admin"}]}}},{"key":"session.id","value":{"stringValue":"2387475c1fe53858e83262b02ab8c789"}},
{"key":"client.address","value":{"stringValue":"127.0.0.1"}},{"key":"client.port","value":{"intValue":"38350"}},
{"key":"server.address","value":{"stringValue":"127.0.0.1"}},{"key":"server.port","value":{"intValue":"5291"}},
{"key":"http.request.method","value":{"stringValue":"POST"}},{"key":"http.route","value":
{"stringValue":"/platform/api/siem/{scope}/{id}/test"}},{"key":"http.response.status_code","value":{"intValue":"200"}},
{"key":"network.protocol.name","value":{"stringValue":"http"}},{"key":"certyx.correlation_id","value":
```

{"stringValue":"0HNPDJ0V49OUH:00000002"}},{"key":"certyx.request_id","value":{"stringValue":"0HNPDJ0V49OUH:00000002"}},
{"key":"certyx.sequence","value":{"intValue":"31"}},{"key":"certyx.event_hash","value":
{"stringValue":"3C7261A625BABD9DCFC9918736DBE1825E73B63BCAF264DAF54D419EE4C6FECC"}},{"key":"certyx.changes","value":
{"stringValue":"[{\"action\":\"siem.destination.tested\",\"before\":{\"scope\":\"platform\",\"id\":\"https-json\"},\"after\":
{\"scope\":\"platform\",\"id\":\"https-json\",\"Ok\":true,\"Error\":null,\"steps\":[{\"Step\":\"dns\",\"Ok\":true,\"Code\":\"dns-
literal\"},{\"Step\":\"tcp\",\"Ok\":true,\"Code\":\"tcp-ok\"},{\"Step\":\"tls\",\"Ok\":true,\"Code\":\"tls-ok\"},
{\"Step\":\"delivery\",\"Ok\":true,\"Code\":\"http-accepted\"}]}},{\"action\":\"request.resource\",\"values\":
{\"scope\":\"platform\",\"id\":\"https-
json\"}}]"}}],"traceId":"5484ed97229f3acd5acff7a0bce7b454","spanId":"ae1a1295fae542f9"}]}]}]}

Das Format `otlp-protobuf` sendet dieselbe Anfrage im binären Protobuf; das Testereignis oben hatte 1030 Byte.

31.6 Dateiarchiv

Das Testereignis wird nicht in Archive geschrieben; dies sind Zeilen von Journalereignissen. Format `csv` (Kopfzeile und eine Zeile):

"schema_version","event_id","event_code","category","action","outcome","outcome_original","severity","severity_label","message","
timestamp","observed_timestamp","export_timestamp","application","version","environment","hostname","host_ip","service_name","com
ponent","organization_id","organization_name","organization_kind","user_id","username","user_role","session_id","authentication_m
ethod","user_verified","user_agent","src_ip","src_port","http_method","http_route","http_status","protocol","correlation_id","req
uest_id","trace_id","resource_type","resource_id","change_type","reason","error_type","sequence","event_hash","agent_fingerprint"
,"changes","changes_redacted","dst_ip","dst_port","span_id"
"certyx.event/1.0","e828c2f0-1089-48fe-838a-9c5267fb560b:1","CONFIG-028","configuration-
change","sales.mail.seeded","success","business","3","low","sales.mail.seeded mail by system:configuration","2026-10-
19T04:19:41.297Z","2026-10-19T04:19:41.297Z","2026-10-19T04:20:43.904Z","CERTYX CLM","1.0.0","production","certyx-node-
1",,"clm","sales","platform",,"platform","system:configuration",,,,,,,,,,,,,"9cf31c6f3be3445ba7e988dc7146cfd4","9cf31c6f3be3445ba
7e988dc7146cfd4",,"sales.mail","mail","create",,,"1","C6E68A3969E5557A0CD49A6969FE6824130603A162AB84DDD0B9D4B1E03FE932",,"
[{""action"":""sales.mail.seeded"",""resource"":""mail"",""before"":{},""after"":
{""Enabled"":false,""Host"":""",""Port"":587,""From"":""",""Username"":null,""OperatorCopy"":null,""InquiryRecipient"":null}}]"
,"true",,,

Format `csv-compat` (Kopfzeile und eine Zeile):

"Sequence","At","OrganizationId","OrganizationName","OrganizationKind","Instance","Actor","SourceIp","AgentFingerprint","Method",
"Route","Status","Outcome","Severity","TraceId","ErrorType","SignInAction","SignInUser","SignInMethod","SignInResult","SignInReas
on","Changes","PreviousHash","Hash"
"1","2026-10-19T04:19:41.2973423Z","platform","","platform","certyx-node-
1","system:configuration","system","","INTERNAL","business/sales.mail.seeded","200","business","3","9cf31c6f3be3445ba7e988dc7146c
fd4","","","","","","","[{""action"":""sales.mail.seeded"",""resource"":""mail"",""before"":{},""after"":
{""Enabled"":false,""Host"":""",""Port"":587,""From"":""",""Username"":null,""password"":
{""Set"":false,""Reference"":null},""OperatorCopy"":null,""InquiryRecipient"":null}}]","","C6E68A3969E5557A0CD49A6969FE6824130603
A162AB84DDD0B9D4B1E03FE932"

Format `ndjson`, Profil `native` (eine Zeile):

{"schema_version":"certyx.event/1.0","event_id":"e828c2f0-1089-48fe-838a-9c5267fb560b:1","event_code":"CONFIG-
028","category":"configuration-
change","action":"sales.mail.seeded","outcome":"success","outcome_original":"business","severity":3,"severity_label":"low","messa
ge":"sales.mail.seeded mail by system:configuration","timestamp":"2026-10-19T04:19:41.297Z","observed_timestamp":"2026-10-
19T04:19:41.297Z","export_timestamp":"2026-10-19T04:20:43.904Z","application":"CERTYX
CLM","version":"1.0.0","environment":"production","hostname":"certyx-node-
1","service_name":"clm","component":"sales","organization_id":"platform","organization_kind":"platform","user_id":"system:configu
ration","correlation_id":"9cf31c6f3be3445ba7e988dc7146cfd4","request_id":"9cf31c6f3be3445ba7e988dc7146cfd4","resource_type":"sale
s.mail","resource_id":"mail","change_type":"create","sequence":1,"event_hash":"C6E68A3969E5557A0CD49A6969FE6824130603A162AB84DDD0
B9D4B1E03FE932","changes":[{"action":"sales.mail.seeded","resource":"mail","before":{},"after":

{"Enabled":false,"Host":"","Port":587,"From":"","Username":null,"OperatorCopy":null,"InquiryRecipient":null}}],"changes_redacted":true}

32 Feldreferenz

Die vollständige Referenz – jedes kanonische Feld mit seiner Quelle und der Zuordnung zu CEF, LEEF, KV, RFC 5424, ECS, OCSF und Splunk CIM – ist das Feldreferenzdokument `docs/SIEM-EXPORT-FIELDS.md` der CERTYX-Dokumentation. Dieser Abschnitt fasst es zusammen.

Unbekannt und leer. Ein Feld, das das Journal nicht kennt oder das nicht zutrifft (zum Beispiel `http_status` einer geschäftlichen Änderung), ist *unbekannt*. JSON-Formate (NDJSON, RFC-5424-MSG, ECS, OCSF, HEC) lassen unbekannte Felder aus; CSV schreibt unbekannt als leeres Feld ohne Anführungszeichen und einen bekannten leeren Wert als `""`; KV schreibt unbekannt als einzelnes `-` und einen bekannten leeren Wert als `""`; CEF und LEEF lassen unbekannte und leere Werte aus. Alle Zeitangaben sind ISO 8601 in UTC mit Millisekunden und `Z` (CEF `rt`, OCSF `time` und das HEC-`time` bezeichnen denselben Zeitpunkt als Epochenwert).

32.1 Kanonisches Ereignis

Schemaversion `certyx.event/1.0`. Die Reihenfolge ist die feste Spaltenreihenfolge von `csv` und `kv`.

Feld	Inhalt
`schema_version`	`certyx.event/1.0`
`event_id`	`<installation id>:<journal sequence>`; eine UUID für Testereignisse
`event_code`, `category`	Aus dem Ereigniscode-Katalog
`action`	Name des Anmeldeereignisses, erste geschäftliche Änderung oder die Pseudoaktion des Ergebnisses (`http.read`, `http.write`, `http.denied`, …)
`outcome`, `outcome_original`	`success`, `failure` oder `unknown`; das Journalergebnis
`severity`, `severity_label`	0–10 (Versuch 2, Erfolg/geschäftlich 3, abgelehnt 5, verweigert 7, Fehler 8; erfolgreiche Anmeldung 3, Fehlschlag 7, Sperre 8); low, medium, high, very-high
`message`	Kurzer englischer Satz
`timestamp`, `observed_timestamp`, `export_timestamp`	Wann das Ereignis eingetreten ist, beobachtet wurde (gleich) und für den Export serialisiert wurde
`application`, `version`, `environment`	`CERTYX CLM`, Produktversion, Umgebung (`Clm:Siem:Environment`, andernfalls die Hosting-Umgebung)
`hostname`, `host_ip`	Erfassende Instanz; Serveradresse (Anreicherung)
`service_name`, `component`	`clm`; erstes Routensegment nach `api` oder erstes Segment einer geschäftlichen Aktion
`organization_id`, `organization_name`, `organization_kind`	Siehe Mehrere Organisationen
`user_id`, `username`, `user_role`, `session_id`	Akteur; Benutzer der Anmeldung; Rolle und Sitzungskorrelations-ID (Anreicherung)
`authentication_method`, `user_verified`, `user_agent`	Anmeldeereignisse
`src_ip`, `src_port`	Quelladresse (nur eine gültige Adresse); Quellport (Anreicherung)
`dst_ip`, `dst_port`	Ziel: Serveradresse und Port, die eine Anfrage angenommen haben (Anreicherung), oder das beobachtete Ziel eines Discovery-Ereignisses
`span_id`	W3C-Span-ID der Anfrageaktivität (Anreicherung)
`http_method`, `http_route`, `http_status`, `protocol`	Anfrageeinträge; Routenvorlage, nie der unveränderte Pfad; `http`
`correlation_id`, `request_id`, `trace_id`	Verbindet Versuch und Ergebnis; ASP.NET-Anfrage-ID; W3C-Trace-ID (Anreicherung)

Feld	Inhalt
`resource_type`, `resource_id`, `change_type`	Geänderte Ressource; `create`, `update`, `delete`
`reason`, `error_type`	Grundcode der Anmeldung oder Fehlertyp; Name des Ausnahmetyps
`sequence`, `event_hash`, `agent_fingerprint`	Journalsequenz (unbekannt bei Testereignissen); Hash der Journalkette; Fingerabdruck des Agent-Zertifikats
`changes`, `changes_redacted`	Aufgezeichnete Änderungen nach der Schwärzung (JSON-Array; kompakter JSON-Text in CSV und KV); `true`, wenn etwas entfernt wurde

32.2 JSON-Datensatz

Verwendet vom Format `json` beim Webhook (Array von Datensätzen), bei Syslog (ein Datensatz pro Nachricht) und bei Splunk HEC (`event`).

Feld	Typ	Inhalt
`Sequence`	Zahl	Position im CERTYX-Journal; 0 für das Testereignis
`EventId`	Zeichenfolge	Ereignis-ID `<installation id>:<journal sequence>` (Deduplizierungsschlüssel); eine UUID für das Testereignis
`At`	Zeichenfolge	Ereigniszeit, ISO 8601 mit Offset
`Tenant`	Zeichenfolge	Kennung der Organisation, `platform` für Betreiberereignisse, `unattributed` für Anfragen ohne angemeldete Organisation
`OrganizationName`	Zeichenfolge	Name der Organisation, sofern die Plattform ihn kennt; leer für `platform`, `unattributed` und Installationen ohne Plattformverwaltung (beim Export ergänzt)
`OrganizationKind`	Zeichenfolge	`organization`, `platform` oder `unattributed` (beim Export ergänzt)
`Instance`	Zeichenfolge	ID der CERTYX-Instanz, die das Ereignis erfasst hat (`Clm:Instance:Id`)
`Actor`	Zeichenfolge	Konto, Identität der Firmenanmeldung, Agent oder Systemakteur; `anonymous` vor der Anmeldung
`SourceIp`	Zeichenfolge	Adresse des Socket-Peers (`system` für Hintergrundereignisse)
`AgentFingerprint`	Zeichenfolge	SHA-256-Fingerabdruck des Clientzertifikats des Agents (Agent-Anfragen)
`Method`	Zeichenfolge	HTTP-Methode, `INTERNAL` für Hintergrundereignisse
`Route`	Zeichenfolge	Routenvorlage (nie der unveränderte Pfad), `business/<action>` für Hintergrundänderungen
`Status`	Zahl	HTTP-Status (200 oder 401 für Anmeldeereignisse)
`Outcome`	Zeichenfolge	`attempt`, `success`, `rejected`, `denied`, `error`, `business`, `authentication`
`TraceId`	Zeichenfolge	Korrelations-ID der Anfrage
`ErrorType`	Zeichenfolge	Ausnahmetyp bei Serverfehlern
`Changes`	Array von Objekten	Jedes Element ist ein JSON-Objekt `{action, before, after}` (verschachtelt, ohne maskierte Anführungszeichen; seit Version 1.2.12, davor war jedes Element eine Zeichenfolge mit dem Objekt); Änderungen außerhalb einer Anfrage und Anmeldeereignisse ergänzen `resource`
`Hash`	Zeichenfolge	SHA-256 des Eintrags in der CERTYX-Journalkette (hexadezimal, Großbuchstaben)
`SignIn`	Objekt	Nur Anmeldeereignisse: `Action`, `User`, `UserVerified`, `Organization`, `Method`, `Result`, `Reason`, `UserAgent`

32.3 CEF-Felder (erste Version)

Format `cef-compat`. Das kanonische Format `cef` ist unter ArcSight beschrieben.

CEF-Schlüssel	Inhalt
Signature ID	`certyx:<outcome>`; Anmeldeereignisse `certyx:<event name>`
Name	`<outcome> <method> <route>`; Anmeldeereignisse `<event name> <method>`
`rt`	Ereigniszeit, Epochenmillisekunden

CEF-Schlüssel	Inhalt
`suser`	Akteur; Anmeldeereignisse: die E-Mail-Adresse des Benutzers
`suid`	Anmeldeereignisse: der Akteur
`src`	Quell-IP (nur eine gültige Adresse)
`requestMethod`, `request`	HTTP-Methode, Routenvorlage
`outcome`	Ergebnis; Anmeldeereignisse `success` oder `failure`
`externalId`	Journalsequenz
`certyxEventId`	Ereignis-ID `<installation id>:<journal sequence>`
`dvchost`	CERTYX-Instanz
`cn1` (`httpStatus`)	HTTP-Status
`cs1` … `cs6`	`organization`, `traceId`, `eventHash`, `agentFingerprint`, `errorType`, `changes` (JSON)
`certyxOrganizationName`, `certyxOrganizationKind`	Name und Art der Organisation (herstellerspezifische Erweiterungsschlüssel; ArcSight zeigt sie als zusätzliche Daten)
`cat`, `act`, `reason`	Anmeldeereignisse: `authentication`, Ereignisname, Grundcode
`requestClientApplication`	Anmeldeereignisse: User-Agent
`flexString1` (`authMethod`), `flexString2` (`userVerified`)	Anmeldeereignisse: Methode, verifizierte Identität

Header-Felder maskieren `\` und den senkrechten Strich; Erweiterungswerte maskieren `\`, `=` und Zeilenumbrüche (`\n`, `\r`).

32.4 LEEF-Felder (erste Version)

Format `leef-compat` (LEEF 2.0, Trennzeichen `^`). Die kanonischen Formate `leef-1.0` und `leef-2.0` sind unter QRadar beschrieben.

LEEF-Attribut	Inhalt
Event ID	`certyx:<outcome>`; Anmeldeereignisse `certyx:<event name>`
`devTime`, `devTimeFormat`	Ereigniszeit (`yyyy-MM-dd'T'HH:mm:ss.SSSZ`)
`usrName`	Akteur; Anmeldeereignisse: die E-Mail-Adresse des Benutzers
`src`	Quell-IP (nur eine gültige Adresse)
`cat`, `sev`	Ergebnis, Schweregrad 1–10
`method`, `resource`, `httpStatus`	HTTP-Methode, Routenvorlage, Status
`organization`, `organizationName`, `organizationKind`, `traceId`, `sequence`, `eventHash`, `agentFingerprint`, `errorType`, `instance`, `changes`	Wie im JSON-Datensatz (`organization` = `Tenant`)
`eventId`	Ereignis-ID `<installation id>:<journal sequence>`
`actor`, `action`, `authMethod`, `result`, `reason`, `userAgent`, `userVerified`	Nur Anmeldeereignisse

Werte maskieren `\` und `^`; Zeilenumbrüche und Tabulatoren werden zu Leerzeichen.

32.5 Sentinel-Spalten

Spalte	Typ
`TimeGenerated`	datetime
`EventId`	string (`<installation id>:<journal sequence>`)

Spalte	Typ
`Sequence`	long
`Organization`, `OrganizationName`, `OrganizationKind`, `Actor`, `SourceIp`, `AgentFingerprint`, `Method`, `Route`, `Outcome`, `TraceId`, `ErrorType`, `Instance`, `EventHash`	string
`Status`	int
`Changes`	dynamic (Array von Objekten)
`User`, `UserOrganization`, `AuthAction`, `AuthMethod`, `AuthResult`, `Reason`, `UserAgent`	string (leer bei anderen Ereignissen als Anmeldeereignissen)
`UserVerified`	boolean

Eine Regel, die angelegt wurde, bevor es `EventId`, `OrganizationName` und `OrganizationKind` gab, nimmt die übrigen Spalten weiterhin auf; fügen Sie sie (string) der Tabelle und der Stream-Deklaration hinzu, um sie zu speichern.

33 Ereigniscode-Katalog

Jede Journalaktion hat einen stabilen Ereigniscode; ein exakter Eintrag hat Vorrang vor einer Familie, und unter Familien gewinnt das längste Präfix. Eine Familienzeile (Präfix) deckt zur Laufzeit gebildete Aktionsnamen ab, etwa `renewal.profile.<decision>`. Codes werden nie neu nummeriert; neue Aktionen erhalten neue Codes.

Kategorie	Codepräfix	Bedeutung
`authentication`	AUTHN	Anmeldung, Abmeldung, Sperre, MFA-Schritte
`authorization`	AUTHZ	abgewiesene Anfragen (401/403) und abgewiesene Anmeldenetzwerke
`user-management`	USER	Konten, SCIM-Benutzer und -Gruppen, Geräte, MFA-Zurücksetzungen
`configuration-change`	CONFIG	Einstellungen für SSO, Netzwerke, Benachrichtigungen, SIEM, Agents, Erkennung, Erneuerung und Überwachung
`data-access`	DATA	Exporte, Archiv-Downloads, angezeigte Dokumentpasswörter, Lesezugriffe
`administrative`	ADMIN	Aktionen des Plattformbetreibers, Kunden, Vertriebsdatensätze, Freigaben und betriebliche Entscheidungen
`security`	SECURITY	Agent-Widerruf und -Identität, vorgelegte Zertifikate, Zertifikats- und Erkennungswarnungen
`application-error`	ERROR	fehlgeschlagene und abgelehnte Anfragen, Bereitstellungsfehler
`system`	SYSTEM	Zustand, Scans, Aufträge, Bereitstellungen, Anfrageversuche und Anmerkungen
`integration`	INTEGRATION	Verbindungsprüfungen und Testnachrichten von SIEM-, Benachrichtigungs-, Warn- und Analyst-Verbindungen, Importe, übersprungene und erneut gesendete SIEM-Ereignisse

Die Kategorie ist zugleich die RFC-5424-MSGID.

33.1 Nach Ergebnis gewählt

Einträge ohne geschäftliche Aktion erhalten eine Pseudoaktion und den Code ihres Ergebnisses. Eine erfolgreiche Anfrage verwendet `http.read` für GET und HEAD, sonst `http.write`; eine nicht katalogisierte geschäftliche Aktion behält ihren Namen und erhält SYSTEM-001.

Code	Kategorie	Aktion	Art
AUTHN-000	authentication	`authentication`	Anmeldeeintrag ohne lesbare Details
AUTHZ-000	authorization	`http.denied`	Ergebnis denied
DATA-000	data-access	`http.read`	Ergebnis success, GET/HEAD
ADMIN-000	administrative	`http.write`	Ergebnis success, andere Methoden
ERROR-000	application-error	`http.error`	Ergebnis error

Code	Kategorie	Aktion	Art
ERROR-001	application-error	`http.rejected`	Ergebnis rejected
SYSTEM-000	system	`http.attempt`	Versuchseintrag
SYSTEM-001	system	`business`	geschäftlicher Eintrag mit nicht katalogisierter Aktion

33.2 Journalaktionen

Code	Kategorie	Aktion	Art
AUTHN-001	authentication	`authentication.sign-in.succeeded`	exakt
AUTHN-002	authentication	`authentication.sign-in.failed`	exakt
AUTHN-003	authentication	`authentication.sign-out`	exakt
AUTHN-004	authentication	`authentication.locked`	exakt
AUTHN-005	authentication	`authentication.repeated-failures`	exakt
AUTHN-006	authentication	`authentication.mfa.enrolled`	exakt
AUTHN-007	authentication	`authentication.mfa.confirmed`	exakt
AUTHN-008	authentication	`authentication.mfa.recovery-codes-regenerated`	exakt
AUTHN-009	authentication	`authentication.claimed-identity`	exakt
AUTHN-010	authentication	`mobile.signin.failed`	exakt
AUTHN-011	authentication	`mobile.session.created`	exakt
AUTHN-012	authentication	`account.locked`	exakt
AUTHN-013	authentication	`account.mfa.recovery-code-used`	exakt
AUTHN-014	authentication	`account.passkey.used`	exakt
AUTHN-015	authentication	`session.revoked`	exakt
AUTHZ-001	authorization	`identity.signin-networks.refused`	exakt
USER-001	user-management	`identity.scim.user-created`	exakt
USER-002	user-management	`identity.scim.user-replaced`	exakt
USER-003	user-management	`identity.scim.user-updated`	exakt
USER-004	user-management	`identity.scim.user-deactivated`	exakt
USER-005	user-management	`identity.scim.user-deleted`	exakt
USER-006	user-management	`identity.scim.group-created`	exakt
USER-007	user-management	`identity.scim.group-replaced`	exakt
USER-008	user-management	`identity.scim.group-updated`	exakt
USER-009	user-management	`identity.scim.group-deleted`	exakt
USER-010	user-management	`identity.scim.role-changed`	exakt
USER-011	user-management	`account.created`	exakt
USER-012	user-management	`account.updated`	exakt
USER-013	user-management	`account.mfa.reset`	exakt
USER-014	user-management	`account.mfa.recovery-codes-regenerated`	exakt
USER-015	user-management	`authentication.mfa.reset`	exakt
USER-016	user-management	`mfa-setup.mail.`	Familie (Präfix)
USER-017	user-management	`customer.account.`	Familie (Präfix)

Code	Kategorie	Aktion	Art
USER-018	user-management	`mobile.device.`	Familie (Präfix)
USER-019	user-management	`mobile.enrollment.`	Familie (Präfix)
USER-020	user-management	`account.passkey.registered`	exakt
USER-021	user-management	`account.passkey.removed`	exakt
USER-022	user-management	`account.second-factor.changed`	exakt
USER-023	user-management	`identity.scim.role-raised`	exakt
USER-024	user-management	`account.create-refused`	exakt
USER-025	user-management	`account.email-refused`	exakt
CONFIG-001	configuration-change	`identity.sso.`	Familie (Präfix)
CONFIG-002	configuration-change	`identity.signin-networks.changed`	exakt
CONFIG-003	configuration-change	`identity.signin-networks.cleared-by-operator`	exakt
CONFIG-004	configuration-change	`platform.signin-networks.changed`	exakt
CONFIG-005	configuration-change	`customer.signin-networks.cleared`	exakt
CONFIG-006	configuration-change	`notification.channel.`	Familie (Präfix)
CONFIG-007	configuration-change	`notification.smtp.`	Familie (Präfix)
CONFIG-008	configuration-change	`siem.destination.`	Familie (Präfix)
CONFIG-009	configuration-change	`agent.configuration.`	Familie (Präfix)
CONFIG-010	configuration-change	`agent.capabilities.updated`	exakt
CONFIG-011	configuration-change	`agent.discovery.profile.updated`	exakt
CONFIG-012	configuration-change	`discovery.scope.`	Familie (Präfix)
CONFIG-013	configuration-change	`renewal.automatic.`	Familie (Präfix)
CONFIG-014	configuration-change	`auto-renewal.`	Familie (Präfix)
CONFIG-015	configuration-change	`platform.cloud-source.`	Familie (Präfix)
CONFIG-016	configuration-change	`cloud-sources.seeded`	exakt
CONFIG-017	configuration-change	`monitoring.alert-channel.`	Familie (Präfix)
CONFIG-018	configuration-change	`monitoring.alert-relay.`	Familie (Präfix)
CONFIG-019	configuration-change	`monitoring.database-log-source.`	Familie (Präfix)
CONFIG-020	configuration-change	`monitoring.settings.`	Familie (Präfix)
CONFIG-021	configuration-change	`analyst.settings.`	Familie (Präfix)
CONFIG-022	configuration-change	`analyst.organization.`	Familie (Präfix)
CONFIG-023	configuration-change	`remediation.policy.`	Familie (Präfix)
CONFIG-024	configuration-change	`recipient-group.updated`	exakt
CONFIG-025	configuration-change	`policy.update.v`	Familie (Präfix)
CONFIG-026	configuration-change	`mail.suppression.`	Familie (Präfix)
CONFIG-027	configuration-change	`mail.events.subscription-confirmed`	exakt
CONFIG-028	configuration-change	`sales.mail.`	Familie (Präfix)
CONFIG-029	configuration-change	`execution.private-verification.`	Familie (Präfix)
CONFIG-030	configuration-change	`execution.target.`	Familie (Präfix)
CONFIG-031	configuration-change	`family.`	Familie (Präfix)

Code	Kategorie	Aktion	Art
CONFIG-032	configuration-change	`certificate.metadata.update`	exakt
CONFIG-033	configuration-change	`ssh.metadata.updated`	exakt
CONFIG-034	configuration-change	`trust-graph.updated`	exakt
CONFIG-035	configuration-change	`endpoint-health.monitor.`	Familie (Präfix)
DATA-001	data-access	`audit.exported`	exakt
DATA-002	data-access	`security-events.exported`	exakt
DATA-003	data-access	`siem.archive.downloaded`	exakt
DATA-004	data-access	`sales.document.password-revealed`	exakt
DATA-005	data-access	`analyst.request`	exakt
DATA-006	data-access	`analyst.conversation.deleted`	exakt
ADMIN-001	administrative	`platform.operator.`	Familie (Präfix)
ADMIN-002	administrative	`platform.customer.changed`	exakt
ADMIN-003	administrative	`platform.license.contract-override`	exakt
ADMIN-004	administrative	`platform.subscription.contract-override`	exakt
ADMIN-005	administrative	`customer.`	Familie (Präfix)
ADMIN-006	administrative	`sales.`	Familie (Präfix)
ADMIN-007	administrative	`certificate.import`	exakt
ADMIN-008	administrative	`renewal.approved`	exakt
ADMIN-009	administrative	`renewal.cancelled`	exakt
ADMIN-010	administrative	`renewal.plan`	exakt
ADMIN-011	administrative	`renewal.profile.`	Familie (Präfix)
ADMIN-012	administrative	`alert.acknowledged`	exakt
ADMIN-013	administrative	`monitoring.incident.`	Familie (Präfix)
ADMIN-014	administrative	`agent.enrollment.ticket.created`	exakt
ADMIN-015	administrative	`agent.discovery.warning.acknowledged`	exakt
ADMIN-016	administrative	`remediation.decision`	exakt
ADMIN-017	administrative	`remediation.notice`	exakt
ADMIN-018	administrative	`ssh.profile.`	Familie (Präfix)
ADMIN-019	administrative	`ssh.revocation.`	Familie (Präfix)
ADMIN-020	administrative	`ssh.certificate.imported`	exakt
ADMIN-021	administrative	`profile.`	Familie (Präfix)
SECURITY-001	security	`agent.revoked`	exakt
SECURITY-002	security	`agent.enrolled`	exakt
SECURITY-003	security	`agent.identity.`	Familie (Präfix)
SECURITY-004	security	`agent.presented-certificate`	exakt
SECURITY-005	security	`certificate.alert`	exakt
SECURITY-006	security	`certificate.alert.escalated`	exakt
SECURITY-007	security	`discovery.alert.raised`	exakt
SECURITY-008	security	`data-protection.secret-purpose-migrated`	exakt

Code	Kategorie	Aktion	Art
ERROR-002	application-error	`customer.provisioning.step-failed`	exakt
ERROR-003	application-error	`sales.customer.provisioning-failed`	exakt
SYSTEM-002	system	`agent.health.changed`	exakt
SYSTEM-003	system	`agent.discovery.baseline.completed`	exakt
SYSTEM-004	system	`agent.discovery.networks.reported`	exakt
SYSTEM-005	system	`discovery.scan.`	Familie (Präfix)
SYSTEM-006	system	`discovery.started`	exakt
SYSTEM-007	system	`discovery.resumed`	exakt
SYSTEM-008	system	`endpoint.observe`	exakt
SYSTEM-009	system	`ssh.job.`	Familie (Präfix)
SYSTEM-010	system	`ssh.deployment.`	Familie (Präfix)
SYSTEM-011	system	`relay.deployment.`	Familie (Präfix)
SYSTEM-012	system	`lease.expired`	exakt
SYSTEM-013	system	`inventory.projected`	exakt
SYSTEM-014	system	`execution.result.imported`	exakt
SYSTEM-015	system	`request.resource`	exakt
SYSTEM-016	system	`unmatched.path`	exakt
SYSTEM-017	system	`certificate.expiry`	exakt
SYSTEM-018	system	`certificate.renewed`	exakt
INTEGRATION-001	integration	`siem.test`	exakt
INTEGRATION-002	integration	`siem.destination.tested`	exakt
INTEGRATION-003	integration	`siem.destination.checked`	exakt
INTEGRATION-004	integration	`notification.channel.tested`	exakt
INTEGRATION-005	integration	`notification.smtp.tested`	exakt
INTEGRATION-006	integration	`monitoring.alert-channel.tested`	exakt
INTEGRATION-007	integration	`analyst.connection.tested`	exakt
INTEGRATION-008	integration	`sales.mail.tested`	exakt
INTEGRATION-009	integration	`siem.archive.expired`	exakt
INTEGRATION-010	integration	`cloud.snapshot.imported`	exakt
INTEGRATION-011	integration	`siem.event.skipped`	exakt
INTEGRATION-012	integration	`siem.destination.replayed`	exakt
INTEGRATION-013	integration	`siem.event.rejected`	exakt