Harpocrypt Threat Model

Stand: 2026-06-15

Status: Pilot-Arbeitsdokument. Kein Ersatz fuer externes Security-/Crypto-Review.

Dieses Dokument beschreibt die wichtigsten Bedrohungen fuer Harpocrypt im aktuellen MVP/Pilotstand. Es ist bewusst praktisch formuliert: Welche Angreifer nehmen wir an, welche Assets muessen geschuetzt werden, welche Kontrollen existieren bereits und welche Punkte bleiben offen.

Scope

Im Scope:

- Mobile App fuer Android, iOS vorbereitet.
- AdminWeb und oeffentlicher Onepager.
- DirectoryService, MessagingService, PolicyService, RelayService, Gateway, AuditService.
- PostgreSQL, Blob/Object Storage, Azure App Service Hosting in West Europe.
- Firebase Cloud Messaging fuer Android Push.
- Organisationen mit OrgAdmins, PlatformAdmins, Nutzern und registrierten Geraeten.

Nicht vollstaendig im Scope:

- Physische Sicherheit von Endgeraeten.
- Betriebssystem-Exploits auf kompromittierten Geraeten.
- Vollstaendige Lieferkettenpruefung aller Dependencies.
- Formales kryptographisches Proof-Modell.
- On-Premise-/Customer-hosted-Deployment.

Sicherheitsziele

- Server und Betreiber sollen Chat-Inhalte, Medien, Voice und Passphrases nicht im Klartext lesen koennen.
- Nur berechtigte aktive Geraete sollen neue Nachrichten entschluesseln koennen.
- Widerrufene Geraete sollen keine neuen Envelopes fuer neue Nachrichten erhalten.
- Push-Benachrichtigungen sollen keine sensiblen Inhalte enthalten.
- Admins sollen Organisationen, Nutzer, Policies und Devices verwalten, aber keine Chat-Klartexte lesen koennen.
- Mandantengrenzen ueber OrgId muessen auf Datenbank- und API-Ebene durchgesetzt werden.
- Security- und Betriebsereignisse sollen nachvollziehbar sein, ohne Inhalte zu protokollieren.

Assets

Asset	Schutzbedarf	Bemerkung
Nachrichteninhalte	sehr hoch	Text, Medien, Voice, Attachments
Private Device Keys	sehr hoch	duerfen Endgeraet nicht verlassen
Data Encryption Keys	sehr hoch	duerfen serverseitig nicht im Klartext vorliegen

Asset	Schutzbedarf	Bemerkung
Passphrase / lokale Secrets	sehr hoch	keine Server-Wiederherstellung
Push Tokens	hoch	personenbezogen, Drittanbieterbezug
User-/Device-Metadaten	hoch	Kommunikationsbeziehungen sichtbar
Admin-Zugriffe	hoch	koennen Org, Devices, Policies beeinflussen
Audit Events	mittel bis hoch	duerfen keine Inhalte enthalten
Produktions-Secrets	sehr hoch	Signing Keys, DB, Storage, FCM

Akteure und Angreifer

Akteur	Annahme
Externer Angreifer	versucht APIs, Tokens, Secrets, Storage oder Push zu missbrauchen
Boeswilliger Nutzer	ist Mitglied einer Org und versucht Zugriff ausserhalb seiner Rechte
Boeswilliger OrgAdmin	hat OrgAdmin-Rechte, soll aber keine Chat-Klartexte lesen koennen
Boeswilliger PlatformAdmin	hat weitreichende Betriebsrechte; Risiko wird ueber Prozesse, MFA, Logging reduziert
Kompromittiertes Geraet	kann Klartext vor/nach Verschluesselung lesen, wenn App/OS kompromittiert ist
Verlorenes Geraet	kann Risiko sein, bis Revoke/Wipe greift
Cloud-/Subprozessor	sieht Infrastruktur- und Zustellmetadaten, aber keine Chat-Klartexte

Trust Boundaries

- Endgeraet zu Harpocrypt APIs.
- Mobile App Secure Storage zu App Runtime.
- Service-zu-Service-Kommunikation.
- Harpocrypt Services zu PostgreSQL und Blob Storage.
- MessagingService zu Firebase Cloud Messaging.
- AdminWeb zu Directory/Policy/Messaging/Audit APIs.
- PlatformAdmin/OrgAdmin zu AdminWeb.

Bedrohungen nach STRIDE

Spoofing

Risiken:

- Angreifer nutzt gestohlene Tokens.
- Geraet wird als anderer User registriert.
- Public Keys werden falsch zugeordnet.

Bestehende Kontrollen:

- Einladungsbasierte Registrierung.
- OrgAdmin-/PlatformAdmin-MFA.
- Device-gebundene Public Keys.
- Org-Code aus Einladung mit Ablauf.

Offen:

- Safety Numbers / Key Verification fuer Nutzer.
- Token-Rotation und Session-Review formal dokumentieren.

Tampering

Risiken:

- Manipulation von Envelopes, Policies oder Message-Metadaten.
- Manipulierte App-Versionen ausserhalb offizieller Distribution.
- Veraenderte Release-Artefakte.

Bestehende Kontrollen:

- MessageRecipients-Modell mit Empfaenger-Envelopes.
- Policy-Versionierung und Policy-Zuordnung pro Message.
- Signierte Android APK fuer Pilotverteilung.
- SHA256-Hash fuer Mobile Releases im AdminWeb.

Offen:

- Vollstaendige Release-Provenance und SBOM in CI.
- App-Update-Pinning gegen erwartete Signatur/Hash weiter haerten.

Repudiation

Risiken:

- Admin- oder Supportaktionen sind spaeter nicht nachvollziehbar.
- Nutzer bestreitet Device-Revoke, Invite oder Policy-Aenderung.

Bestehende Kontrollen:

- AuditService fuer Admin-/Security-Ereignisse.
- Audit-Retention und CSV-Export vorbereitet.
- Incident-Runbooks fuer Dokumentation.

Offen:

- Produktive Audit-Retention je Kunde festlegen.
- Access Reviews fuer PlatformAdmins formalisieren.

Information Disclosure

Risiken:

- Chat-Inhalte werden serverseitig oder in Logs sichtbar.

- Push enthaelt sensitive Inhalte.
- Metadaten zeigen Kommunikationsmuster.
- Secrets liegen in AppSettings, Logs oder Repo.

Bestehende Kontrollen:

- Ende-zu-Ende-Verschluesselung fuer Inhalte.
- Push ohne Nachrichtentext, Absendernamen oder Dateinamen.
- Logging-Leitlinie ohne Passphrases, private Keys, Tokens und Klartextnachrichten.
- Production-Config erzwingt sichere Runtime-Werte.

Offen:

- Retention technisch fuer alle Log-/Auditklassen erzwingen.
- Externes AppSec-/Crypto-Review.
- Datenflussdiagramm fuer DPA/DSFA-Anlage grafisch ausarbeiten.

Denial of Service

Risiken:

- Service-Ausfall durch API-Missbrauch.
- Push-/Messaging-Ausfall.
- Migration blockiert Service-Start.

Bestehende Kontrollen:

- Health-/Readiness-Endpunkte.
- Azure Alerts und Readiness-Anzeige.
- Backup-/Restore-Runbook.
- Migration-Checks in Readiness.

Offen:

- Rate Limits und Abuse Controls pruefen.
- Last-/Smoke-Tests fuer Pilotgrenzen definieren.
- RTO/RPO je Pilotvertrag verbindlich festlegen.

Elevation of Privilege

Risiken:

- OrgAdmin greift auf andere Organisation zu.
- User erhaelt Admin-Scopes.
- PlatformAdmin-Zugriff wird missbraucht.

Bestehende Kontrollen:

- OrgId als zentrale Mandantengrenze.
- Rollen/Scopes fuer Admin- und Useraktionen.

- MFA fuer AdminWeb.
- OrgAdmin-Scope auf eigene Organisation.

Offen:

- Regelmaessige Access Reviews.
- Separate Break-glass-Prozesse fuer PlatformAdmin.
- Security Tests fuer mandantenuebergreifende API-Zugriffe.

Missbrauchsfaelle

Fall	Erwartete Reaktion
Teilnehmer ohne aktives Device wird ausgewaehlt	UX-Hinweis, keine stille Zustellung, keine Envelopes ohne Key
Geraet wird verloren	Device revoke, Wipe anstossen, neue Nachrichten nicht mehr an altes Device
Organisation wird gesperrt	Registrierung, Token-Erneuerung und Senden blockieren, klare Nutzerinfo
Push-Token ungueltig	Token deaktivieren, Fehler auditieren, kein Chatinhalt im Push
Passphrase vergessen	Neue Registrierung; alte Inhalte ohne passende Keys nicht wiederherstellbar
Admin will Chatinhalt lesen	Nicht moeglich, solange E2E und Client-Keys korrekt sind

Wichtigste Restrisiken

- Kompromittierte oder entsperrte Endgeraete koennen Inhalte offenlegen.
- Metadaten bleiben sichtbar und muessen datenschutzrechtlich behandelt werden.
- PlatformAdmins haben hohen Einfluss auf Betrieb und Verfuegbarkeit.
- Ohne externe Pruefung bleiben Implementierungs- und Kryptographiefehler moeglich.
- Firebase/APNs sehen technische Push-Zustellmetadaten.

Naechste Security-Arbeitspakete

- Safety Numbers / Key Verification.
- Externes Security-/Crypto-Review.
- SBOM und Dependency Scanning in CI.
- Rate Limiting und Abuse Controls.
- Formaler Access-Review-Prozess.
- Vollstaendige Retention technisch erzwingen.
- Kundenfaehiges Datenflussdiagramm fuer DSFA/DPA.

Verweise

- Security/Privacy Whitepaper: docs/security-privacy-whitepaper.md
- TOMs: docs/technical-organisational-measures.md

- AVV-Arbeitsentwurf: docs/data-processing-agreement-working-draft.md
- Incident Response: docs/incident-response-runbook.md
- Security-Unterlagen-Index: docs/security-documentation-index.md