sneakers·pam

Sneakers-PAM Security Review

Architecture, key custody and controls of a self-hosted privileged access vault, with the evidence behind each claim.

DOCUMENT	Draft 1
DATE	29 October 2026
PRODUCT VERSION	v0.1.0-rc.10 (prerelease)
CLASSIFICATION	Public

Sneakers-PAM holds no SOC 2 report, ISO 27001 certificate or other certification. This review describes only what is built and published, maps it to SOC 2 Trust Services Criteria, ISO/IEC 27001:2022 Annex A, NIST SP 800-63B and OWASP ASVS 4.0.3 so an independent reviewer can test it, and lists every known gap. Where a control depends on how you deploy Sneakers, it says so.

Contents

1. 1. Scope and how to read this review
2. 2. Architecture and data flows
3. 3. Key hierarchy, custody and encryption
4. 4. Identity, MFA, SSO and sessions
5. 5. Authorization, approvals and break-glass
6. 6. Reveal, check-out and rotation
7. 7. Tamper-evident audit trail
8. 8. MCP server and machine API
9. 9. Appliance: Secure Boot, TPM and signed updates
10. 10. Supply chain and releases
11. 11. Backups and recovery
12. 12. Logging, monitoring and health
13. 13. Secure development and vulnerability handling
14. Gaps and planned work
15. Framework crosswalk

1. Scope and how to read this review

Sneakers-PAM is an open-source privileged access manager: a vault for privileged credentials, with check-out, approvals, rotation, brokered SSH, an MCP server for agents, and a hash-chained audit trail. You run it yourself, as a signed appliance or as Helm charts on your own Kubernetes. The project operates no hosted service and never receives your data.

This review covers the v0.1.0 release train, currently at release candidate 10. Each control names the component tag it first ships in; the v0.1.0-rc.10 release builds every service from a pinned commit. A control marked shipped has its code on a published tag. Partial means it is built with a named gap. Planned means it isn't built.

FRAMEWORK	VERSION	HOW IT IS USED HERE
SOC 2 Trust Services Criteria	2017, revised points of focus 2022	Common Criteria (CC), Availability (A) and Confidentiality (C) references per control
ISO/IEC 27001 Annex A	2022	Annex A control references per control
NIST SP 800-63B	Revision 3 numbering	Authenticator and session requirements for sign-in and step-up
OWASP ASVS	4.0.3	Verification requirements for the web and API surface

> Sneakers-PAM is in prerelease. Several controls are partial, one area (encryption between services) is not built, and backups stay on the box with no offsite copy. They are described plainly in their sections and collected in Gaps and planned work.

2. Architecture and data flows

Sneakers is a set of small Go services, each with its own database in a shared PostgreSQL cluster, talking over gRPC. There is no message broker. Valkey (Redis-compatible) holds gateway sessions, the notification inbox and short-lived SSH tickets.

COMPONENT	ROLE	HOLDS
gateway	The only entry point. GraphQL for browser sessions and for tokens, sign-in, sessions, CSRF.	Session records in Valkey
identity	Users, groups, second factors, service accounts and tokens.	Encrypted TOTP seeds, hashed codes and tokens
vault	Secrets, folders and access rules, versions, break-glass, rotation schedules.	Envelope-encrypted values, the key ring
workflow	Check-out leases and approvals.	Lease and request state
audit	Append-only, hash-chained audit records.	Metadata only, never values
connector	Heartbeat and rotation against targets over LDAP/LDAPS and Kerberos.	Nothing at rest
sshbroker	In-browser SSH using a credential the user never sees.	Single-use tickets, no key material
notify	In-app notification inbox.	Action, subject and actor labels only
mcp	MCP server and CLIs for agents.	Nothing; forwards the caller's own token
web	Staff and admin web apps, server-rendered.	Nothing; the browser holds one HttpOnly cookie
Ory Kratos, Hydra, Polis	Passwords, OAuth for agents and CLIs, SAML sign-on.	Password hashes, OAuth clients

Figure 1 is the high-level design of the product. The same services run on the appliance and on your own Kubernetes.

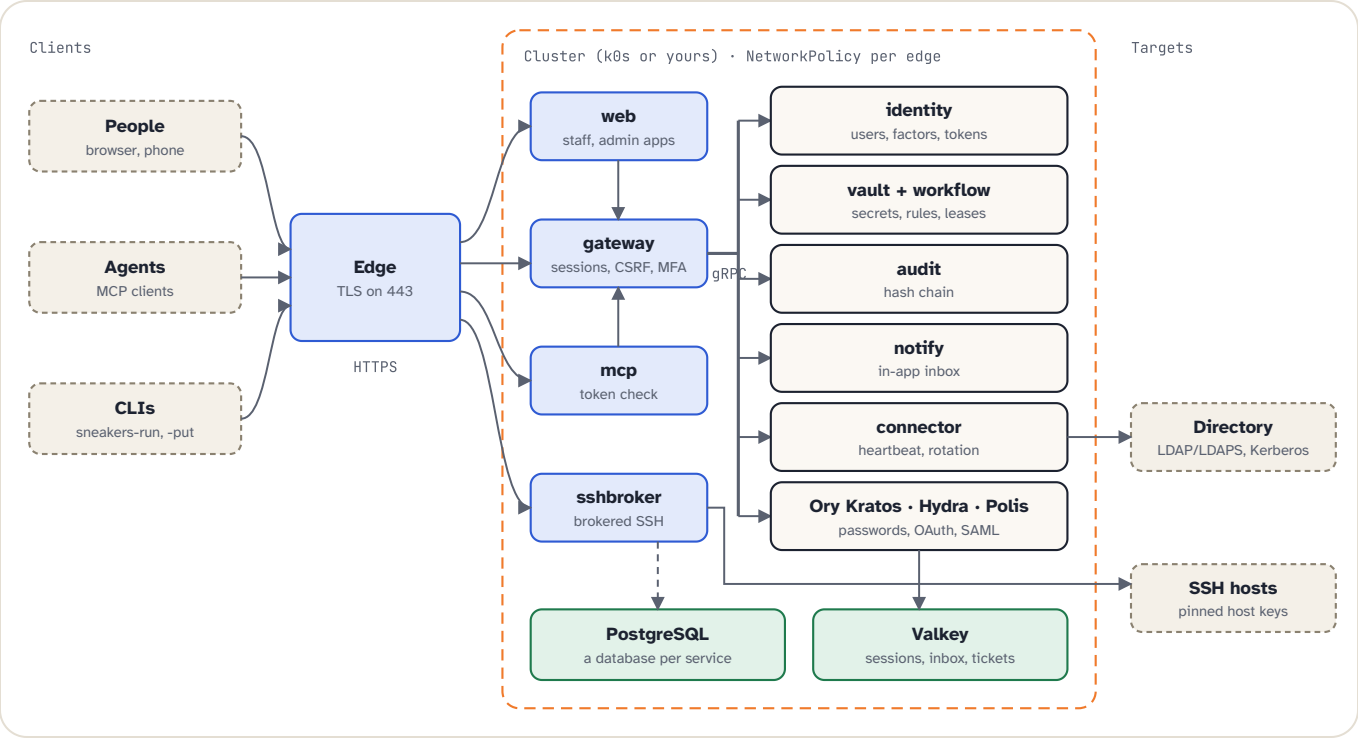


Figure 1. Architecture and trust boundaries. Every internal call is gRPC carrying a verified workload identity. Solid arrows are calls; the dashed arrow is the front tier's use of the database and Valkey.

Trust boundaries: browsers and agents reach only the gateway, the MCP server, the web apps and the SSH broker's WebSocket. Every internal call carries a verified workload identity and is limited by a per-method caller allow-list and a NetworkPolicy. Only the connector and the SSH broker reach target systems.

SIGN IN

1. The browser posts credentials to the web app, which calls the gateway server-side.
2. Kratos checks the password; identity checks the second factor (TOTP, emailed code or passkey).
3. The gateway stores the session in Valkey and returns an opaque, HttpOnly, SameSite=Strict cookie.
4. Every later request carries the cookie and a CSRF token; a session without a verified factor gets mfa_required.

REVEAL A VALUE

1. The gateway passes the user's identity and second-factor time to the vault.
2. The vault refuses a retired secret, then checks the field, the folder rules, any approval and step-up.
3. It unwraps the secret's data key, decrypts only the requested field and records secret.reveal in the audit chain.
4. The value goes back through the gateway to the web app, masked until the user reveals it.

AN AGENT USES A SECRET THROUGH MCP

1. The agent calls the MCP server with its own OAuth, service-account or personal token.
2. The MCP server checks the token (failing closed) and forwards it unchanged to the gateway's machine API.
3. The vault applies the same rules. An approval-gated secret returns an approval link, not a value.
4. A person approves in the browser with a second factor; the agent redeems the value once, within 60 seconds.

CHECK-OUT AND ROTATION

1. workflow checks access, the type and any existing lease, and issues one lease.
2. On check-in by the holder, the vault queues a rotation.
3. The connector pulls the job, changes the password on the target using the account's own current password, and reports back.
4. The vault stores the new value as a new version; every step is audited.

Controls: Architecture and trust boundaries

CONTROL	STATUS	EVIDENCE
ARC-1 Single entry point for people and machines Clients reach Sneakers only through the gateway (two GraphQL endpoints, one for browser sessions and one for tokens), the MCP bridge and the web apps. Backend services aren't exposed outside the cluster. SOC 2 CC6.1, CC6.6 · ISO 27001 A.8.20, A.8.22 · ASVS V1.4.1	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/README.md sneakers-release/charts/sneakers-lib/templates/_resources.tpl (per-service NetworkPolicy)
ARC-2 Workload identity on every internal call Each service-to-service gRPC call carries the caller's projected Kubernetes ServiceAccount token. The callee checks issuer, audience, expiry and a per-method caller allow-list, and refuses to start without an issuer configured. Only the gateway may pass an end user's identity. SOC 2 CC6.1, CC6.3 · ISO 27001 A.5.15, A.8.5 · ASVS V1.4.4, V4.1.3	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/server/workloadauth.go sneakers-vault/internal/server/workloadauth_tokens_test.go (TestWorkloadAuthAcceptsAndRefusesTokens) sneakers-audit/internal/grpcsvc/callers_test.go (TestCallerAuth_UnlistedPodIsRefusedEvenWithAValidToken) sneakers-gateway/cmd/gateway/backend_test.go (TestBackendDialOptionsSendTheWorkloadTokenOnEveryCall) Gap: sneakers-gateway#27 Workload identity for machine-API callers

CONTROL	STATUS	EVIDENCE
ARC-3 Network policy per call-graph edge The Helm charts install a NetworkPolicy for each service that admits only its declared callers, plus policies for the bundled identity provider and database. The install test checks that blocked pods are blocked. Outbound traffic isn't restricted by default. SOC 2 CC6.6 · ISO 27001 A.8.20, A.8.22	PARTIAL SINCE V0.1.0-RC.3	sneakers-release/charts/sneakers-lib/templates/_resources.tpl sneakers-release/.github/workflows/install-test.yml
ARC-4 Hardened containers Every service runs as non-root with a read-only root filesystem, no privilege escalation, all capabilities dropped and a seccomp profile. Each chart's values schema rejects an install that turns these off. SOC 2 CC6.8 · ISO 27001 A.8.9 · ASVS V14.1.1	SHIPPED IN V0.1.0-RC.3	sneakers-release/charts/sneakers-lib/templates/_pod.tpl

3. Key hierarchy, custody and encryption

Every secret field is envelope-encrypted. The hierarchy has three tiers, and only the top one ever leaves the database.

KEY	ALGORITHM	WHERE IT LIVES	ROTATION
Root key	256-bit, AES-256-GCM key wrap	Process memory, loaded from the environment. On the appliance, on the TPM-sealed state volume and in the owner's escrow.	Not rotatable today
Working keys (kek-v1, kek-v2, …)	256-bit, AES-256-GCM key wrap	Wrapped by the root key in the vault database; one active	Every 90 days by default, or on demand by a site admin
Data key, one per secret	256-bit, AES-256-GCM, 96-bit random nonce per field, field name as AAD	Wrapped by a working key, stored with each record and version	Re-wrapped on working-key rotation
TOTP seed key	AES-256-GCM	identity service environment; appliance escrow	Not rotatable without re-enrolment
Appliance disk key	LUKS2, AES-XTS, 256-bit	Sealed in the TPM to PCR 7 and 11; key-file fallback without a TPM	Resealed on each update

Rotating a working key re-wraps each data key and never decrypts a value. The job is idempotent and resumable, and retired keys stay available for decryption only. At start-up the root key must unwrap every working key before any migration runs, so a wrong key stops the vault with the schema untouched.

> Traffic between services is authenticated but not encrypted: gRPC clients dial without TLS and the gateway serves plain HTTP behind the ingress. On your own Kubernetes, run Sneakers on a private network or behind a service mesh with mTLS. Metadata (names, folders, rules) is stored unencrypted in the database; on the appliance it is protected by full-disk encryption.

Controls: Keys and encryption

CONTROL	STATUS	EVIDENCE
CRY-1 Envelope encryption of every secret field Each secret has its own 256-bit data key from the OS CSPRNG. Every field value is sealed with AES-256-GCM under a fresh 96-bit random nonce, with the field name bound as associated data, so a value can't be moved to another field undetected. SOC 2 CC6.1, C1.1 · ISO 27001 A.8.24 · ASVS V6.2.2, V6.3.1	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/crypto/envelope.go sneakers-vault/internal/vault/crypto/envelope_test.go (TestSealStoresNoPlaintext, TestTamperedCiphertextFailsAuth, TestWrongKEKCannotOpen)

CONTROL	STATUS	EVIDENCE
CRY-2 Three-tier key hierarchy with a rotating key ring A root key wraps versioned working keys, which wrap each secret's data key. Working keys rotate by re-wrapping data keys only, without decrypting any value; rotation is resumable, retired keys stay decrypt-only, and a scheduler rotates every 90 days by default. Only a site admin can rotate on demand. SOC 2 CC6.1 · ISO 27001 A.8.24 · ASVS V6.4.1, V6.4.2	SHIPPED IN V0.1.0-RC.1	sneakers-vault/docs/kek-keyring.md sneakers-vault/internal/vault/grpcsvc/kek_rotate_test.go (TestRotateKek_SiteAdmin_RewrapsEveryRecordAndAudits, TestRotateKek_NonAdmin_Denied) sneakers-vault/internal/vault/grpcsvc/kek_schedule_test.go (TestKekSchedulerTick_RotatesWhenDue) Gap: sneakers-vault#104 Key ring design note
CRY-3 Fail-closed key start-up Outside development the vault won't start without a root key, and the root key must unwrap every working key before any database migration runs. An unknown key reference fails closed. SOC 2 CC6.1 · ISO 27001 A.8.24 · ASVS V6.2.1	SHIPPED IN V0.1.0-RC.1	sneakers-vault/cmd/vault/boot.go sneakers-vault/cmd/vault/boot_test.go (TestMigrateAfterPreflight_CorrectKEK_MigratesAndBoots) sneakers-vault/internal/vault/crypto/keyring_test.go (TestKeyringKEK_UnknownRef_FailsClosed)
CRY-4 Root key custody The vault reads its root key from the environment at start and holds it in memory. On the appliance the key lives on the encrypted state volume, whose key is sealed in the TPM, and is copied into the owner's escrow. The root key itself can't be rotated, and there is no HSM or cloud KMS provider yet. SOC 2 CC6.1 · ISO 27001 A.8.24 · ASVS V6.4.1	PARTIAL SINCE V0.1.0-RC.3	sneakers-appliance/docs/key-custody.md (Product keys) sneakers-vault/cmd/vault/main.go Gap: sneakers-vault#41 HSM and cloud KMS key providers
CRY-5 Second-factor secrets and tokens at rest TOTP seeds are sealed with AES-256-GCM under a dedicated key. Email codes and API tokens are stored only as SHA-256 hashes. The TOTP key can't be rotated yet without re-enrolling every factor. SOC 2 CC6.1 · ISO 27001 A.5.17, A.8.24 · ASVS V2.9.1, V6.1.1	SHIPPED IN V0.1.0-RC.1	sneakers-identity/internal/secrets/secrets.go sneakers-identity/internal/secrets/secrets_test.go (TestOpenTamperFails, TestOpenWrongKeyFails)
CRY-6 Metadata at rest Only field values are application-encrypted. Secret names, folders, rules, targets and settings are stored as plain JSON in each service's database. On the appliance that database sits on the LUKS2-encrypted state volume; on your own Kubernetes, disk encryption is yours to provide. SOC 2 C1.1 · ISO 27001 A.8.24	PARTIAL SINCE V0.1.0-RC.1	sneakers-vault/migrations/vault/0001_baseline.up.sql

CONTROL	STATUS	EVIDENCE
CRY-7 Encryption in transit between services Internal gRPC calls are authenticated (ARC-2) but not encrypted: every client dials without TLS, and the gateway serves plain HTTP behind the cluster ingress. Confidentiality on the wire depends on the platform today (a private network or a service mesh with mTLS). SOC 2 CC6.7 · ISO 27001 A.8.24, A.8.20 · ASVS V9.2.2	PLANNED	sneakers-gateway/cmd/gateway/backend.go sneakers-gateway/README.md (Before production) Gap: sneakers-appliance#531 Issue certificates through ACME with cert-manager
CRY-8 Export-compliant, standard cryptography Sneakers uses only standard, published algorithms from the Go standard library and well-known libraries (AES-GCM, AES-XTS through LUKS2, SHA-256, HKDF, ECDSA, age). There is no FIPS 140-3 validated mode yet. ISO 27001 A.8.24, A.5.31 · ASVS V6.2.2	SHIPPED IN V0.1.0-RC.1	website/CRYPTO.md Gap: sneakers-appliance#216 FIPS 140-3 mode

4. Identity, MFA, SSO and sessions

Passwords are held by Ory Kratos and never touch a Sneakers service. Every session needs a second factor: TOTP, an emailed code or a passkey. Sensitive actions (changing factors, approving, consenting to an agent, revealing where policy requires, sensitive check-outs) need a factor verified within MFA_MAX_AGE, 30 minutes by default and 4 hours at most.

The browser holds an opaque session id only. The cookie is HttpOnly, SameSite=Strict and Secure by default; sessions slide on use within an admin-set lifetime of 15 to 60 minutes, and an admin cutoff ends every session issued before it. CSRF uses a 32-byte double-submit token compared in constant time, and CORS is an exact-origin allow-list.

> Against SP 800-63B: TOTP codes aren't yet refused on reuse within their period (5.1.4.2), and emailed codes don't prove possession of a device, so they don't count as an out-of-band authenticator (5.1.3.1). Passkeys work as a second factor; passwordless passkey sign-in isn't built.

Controls: Identity, MFA and sessions

CONTROL	STATUS	EVIDENCE
IDN-1 Passwords held by a dedicated identity provider Passwords are stored and checked by the bundled Ory Kratos, never by Sneakers services. Its hashing parameters are Kratos configuration and aren't asserted by this review. Self-service reset uses an emailed code and answers the same way whether or not the account exists. SOC 2 CC6.1 · ISO 27001 A.5.17, A.8.5 · 800-63B 5.1.1.2 · ASVS V2.4.1, V2.5.2	SHIPPED IN V0.1.0-RC.1	sneakers-identity/README.md sneakers-identity/internal/grpcsvc/reset.go
IDN-2 Second factor required for every session A session without a verified second factor gets mfa_required on the API. Factors are TOTP (RFC 6238), emailed six-digit codes (single use, five minutes, five guesses) and WebAuthn passkeys as a second factor with a strictly increasing sign counter. SOC 2 CC6.1 · ISO 27001 A.8.5 · 800-63B 4.2.1, 5.1.4, 5.1.8 · ASVS V2.7, V2.8, V4.3.1	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/internal/bff/http.go sneakers-identity/internal/grpcsvc/mfa.go sneakers-identity/internal/grpcsvc/mfa_pg_test.go (TestPGTotpVerifyAndReEnrollGuard)
IDN-3 TOTP replay and guessing limits A pending second-factor sign-in ends after five wrong codes. TOTP verification itself doesn't yet refuse a code reused inside its period, and email codes don't prove possession of a device, so they don't meet SP 800-63B's out-of-band rules. 800-63B 5.1.4.2, 5.2.2, 5.1.3.1 · ASVS V2.2.1, V2.8.4	PARTIAL SINCE V0.1.0-RC.1	sneakers-gateway/internal/bff/mfa.go sneakers-gateway/internal/bff/stepup_test.go (TestMfaStepUp_AttemptBudgetEndsTheSession)

CONTROL	STATUS	EVIDENCE
IDN-4 Server-side sessions in a hardened cookie The browser holds only an opaque session id in a cookie that is HttpOnly, SameSite=Strict and Secure by default; a malformed setting stops the gateway at start. Sessions slide on use within an admin-set lifetime of 15 to 60 minutes, and an admin cutoff ends every older session. SOC 2 CC6.1 · ISO 27001 A.8.5 · 800-63B 7.1, 4.2.3 · ASVS V3.2.1, V3.3.1, V3.4.1, V3.4.2, V3.4.3	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/docs/cookies.md sneakers-gateway/internal/bff/cookie_secure_test.go (TestEveryCookieHonoursSecure) sneakers-gateway/internal/bff/session_cutoff_test.go (TestASessionIssuedBeforeTheCutoffIsEnded) Gap: sneakers-gateway#31 Session tokens are stored unencrypted in Redis
IDN-5 Step-up for sensitive actions Changing factors, approving requests, consenting to an agent's sign-in, revealing where policy requires it, and sensitive check-outs need a second factor verified within MFA_MAX_AGE (30 minutes by default, 4 hours at most). SOC 2 CC6.1 · ISO 27001 A.8.5 · 800-63B 7.2 · ASVS V3.7.1	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/docs/configuration.md (MFA_MAX_AGE) sneakers-gateway/internal/bff/stepup_test.go (TestMfaStepUp_NeedsSessionAndCSRF)
IDN-6 CSRF and cross-origin protection Every state-changing browser request carries a 32-byte CSRF token checked by double submit in constant time. CORS is an exact-origin allow-list; a wildcard or malformed entry stops the gateway at start. ISO 27001 A.8.26 · ASVS V4.2.2, V13.2.3, V14.5.3	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/internal/bff/csrf.go sneakers-gateway/cmd/gateway/cors_test.go (TestCORSBadEntriesRefused)
IDN-7 Browser security headers The web apps send a nonce-based Content-Security-Policy with frame-ancestors 'none', plus nosniff, X-Frame-Options DENY and a same-origin referrer policy. Styles still allow 'unsafe-inline'. ISO 27001 A.8.26 · ASVS V14.4.3, V14.4.4, V14.4.7	SHIPPED IN V0.1.0-RC.2	sneakers-web/packages/shell/src/server/csp.server.ts sneakers-web/packages/shell/src/server/csp.server.test.ts
IDN-8 Single sign-on and passkey sign-in SAML sign-on runs through the bundled Ory Polis at the gateway. Passkeys work as a second factor, but passwordless passkey sign-in isn't built. SOC 2 CC6.1 · ISO 27001 A.8.5 · 800-63B 5.1.8, 5.2.5	PARTIAL SINCE V0.1.0-RC.1	sneakers-gateway/internal/bff/sso.go Gap: sneakers-identity#22 Passkey sign-in

5. Authorization, approvals and break-glass

Access is decided in the vault from rules on a folder tree, evaluated most specific first, with deny as the default. A rule grants read, informed, approve or manage independently. Owners get read, approve and manage on their own secrets.

APPROVAL LEVEL	AN OWNER	SOMEONE WITH READ ACCESS	SOMEONE WITHOUT READ ACCESS
Normal	No approval	No approval	Refused
Approval required	No approval	Any one owner approves	Refused
Always approve	Another owner or an approver decides	An owner or approver decides	Refused

Nobody approves their own request. On an install where nobody else can decide, the requester confirms once with a fresh second factor, and that confirmation is recorded; it is not two-person control. Break-glass is for site admins from their own web session with a TOTP code, lasts 15 minutes, and is refused if its audit record can't be written.

Controls: Authorization and approvals

CONTROL	STATUS	EVIDENCE
AUZ-1 Deny-by-default folder rules Access comes from rules on a folder tree, evaluated most specific first, first match wins, deny by default. Rules grant read, informed, approve and manage separately; approve and manage imply read, informed never does. Access decisions are made in the vault, not the gateway. SOC 2 CC6.1, CC6.3 · ISO 27001 A.5.15, A.5.18, A.8.3 · ASVS V4.1.3, V4.1.5	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/authz/resolve.go sneakers-vault/internal/vault/authz/resolve_test.go (TestResolve_ReadableUserActs, TestResolve_InformedIndependentOfRead)
AUZ-2 Approvals no one grants to themselves A secret can require approval or always require it. A requester never approves their own request. Where nobody else can decide (a single-user install, or the only approver), the requester confirms once with a fresh second factor; that is a recorded self-confirmation, not two-person control. SOC 2 CC6.3 · ISO 27001 A.5.3, A.8.2 · ASVS V4.1.3	SHIPPED IN V0.1.0-RC.1	sneakers-mcp/docs/api.md (Approvals) sneakers-vault/internal/vault/grpcsvc/secret_use_test.go (TestSecretUseIsReleasedOnceAfterAnOwnerApproves) Gap: sneakers-gateway#47 Approval queue, renewal, force check-in and quorum
AUZ-3 One task, one prompt An agent run's requests share one approval page with one second factor. Each item is still decided in the vault and audited on its own, the requester can't decide (the machine API has no decide call), and the stated purpose is shown as untrusted text. SOC 2 CC6.3 · ISO 27001 A.8.2	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/internal/resolvers/batch_approval_test.go

CONTROL	STATUS	EVIDENCE
AUZ-4 Break-glass for people only Break-glass is open to site admins only, from their own web session with a TOTP code, and lasts 15 minutes. Agents and tokens are refused, refusals are audited, and a break-glass reveal is refused if its audit record can't be written. SOC 2 CC6.1, CC6.3 · ISO 27001 A.8.2, A.8.18 · ASVS V4.3.1	SHIPPED IN V0.1.0-RC.1	sneakers-gateway/internal/resolvers/break_glass_people_only_test.go (TestPlainBreakGlassSecretIsRefusedToAgentsAndTokens) sneakers-vault/internal/vault/grpcsvc/breakglass_test.go (TestBreakGlassAuditFailClosedDeniesReveal) Gap: sneakers-vault#70 Dual control for break-glass Gap: sneakers-vault#72 Break-glass rate limits and frequency alerts Gap: sneakers-vault#151 Enforce people-only reveals in the service, not only by routing
AUZ-5 Recovery role for history Listing a secret's history needs read access; revealing or restoring an earlier version needs the separate recovery role plus a fresh second factor. Site admin doesn't imply recovery. SOC 2 CC6.3 · ISO 27001 A.8.2, A.8.3	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/grpcsvc/recovery_test.go (TestRevealingAPriorVersionNeedsRecoveryAndFreshMFA)

6. Reveal, check-out and rotation

A value leaves the vault in four ways: a person reveals one field; an approved use is redeemed once; a check-out lease hands over a credential that rotates on check-in; or the SSH broker uses a key on the user's behalf without showing it. Each one is audited.

Rotation runs on a schedule or on demand through the connector, which installs nothing on targets and rotates an account using its own current password. LDAP/LDAPS and Kerberos rotation are built; SSH, WinRM and Windows-local rotation are not.

Controls: Reveal, check-out and rotation

CONTROL	STATUS	EVIDENCE
ACC-1 Reveal one field, with checks in order A reveal is refused for a retired secret, then checks the field is sensitive, read access, approval and step-up, and only then decrypts the one requested field and records secret.reveal. Step-up on reveal is a global setting with per-folder overrides that only site admins set. SOC 2 CC6.1, C1.1 · ISO 27001 A.8.3, A.8.12 · ASVS V4.1.1, V8.3.4	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/grpcsvc/secrets.go sneakers-vault/internal/vault/grpcsvc/sensitive_settings_test.go (TestRevealStepUpFollowsTheGlobalSettingAndFolderOverrides)
ACC-2 Single-use, short-lived approved uses A prepared use expires in 10 minutes; once approved it must be redeemed within one minute, by the same token, once. Access is checked again at redemption, and the audit record never carries the value. Standing pre-approvals are capped at 24 hours. SOC 2 CC6.1 · ISO 27001 A.8.3 · ASVS V4.1.3	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/grpcsvc/secret_use_test.go (TestSecretUseOnlyTheSameTokenRedeems, TestSecretUseRechecksAccessAtRedeem, TestSecretUseAuditNeverCarriesTheValue) sneakers-vault/internal/vault/grpcsvc/secret_use_hardening_test.go (TestRedeemMustFollowApprovalWithinAMinute)
ACC-3 Check-out leases that rotate on check-in Check-out needs read access, a type that allows it, a fresh second factor for sensitive types (on by default) and no lease already held. Only the holder can check in, and check-in rotates the credential. Concurrent check-outs issue one lease. SOC 2 CC6.1, CC6.3 · ISO 27001 A.5.18, A.8.2	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/workflow/grpcsvc/checkout.go sneakers-vault/internal/workflow/grpcsvc/checkout_checks_test.go (TestCheckinByANonHolderIsRefusedAndRotatesNothing, TestConcurrentCheckoutsIssueOneLease) sneakers-vault/internal/workflow/grpcsvc/sensitive_checkout_test.go (TestSensitiveCheckoutNeedsAFreshMFAWhenOn)

CONTROL	STATUS	EVIDENCE
ACC-4 Rotation without agents on targets The connector pulls due heartbeat and rotation jobs and changes passwords over LDAP/LDAPS and Kerberos using the account's own current password, so it needs no admin rights. A heartbeat reports an untrusted certificate as its own result, with the reason. The directory's built-in Administrator is matched by SID and never rotated. SSH, WinRM and SAMR rotation aren't built. SOC 2 CC6.1 · ISO 27001 A.5.17	PARTIAL SINCE V0.1.0-RC.1	sneakers-connector/README.md sneakers-connector/internal/adapter/ldap_account_test.go (TestRotateADBuiltinAdministratorIsRefusedWithoutChange) Gap: sneakers-connector#16 SSH, WinRM and SAMR rotation Gap: sneakers-connector#33 Respect directory password rules when rotating Gap: sneakers-connector#75 Pin LDAPS certificates for an enterprise CA
ACC-5 Strong generated passwords Generated passwords come from the OS CSPRNG: 20 characters from an 86-character pool by default (about 128 bits), with per-policy length, required classes and excluded characters. ISO 27001 A.5.17 · ASVS V6.3.1	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/vault/grpcsvc/pwgen_test.go (TestGeneratePasswordRequiredClasses)
ACC-6 Brokered SSH without handing over the key In-browser SSH uses a single-use ticket that lives 30 seconds and holds no key material. The broker fetches the key as the user when the browser connects, wipes it after use, connects only to pinned host keys and cuts sessions at 8 hours. Sessions aren't recorded yet. SOC 2 CC6.1 · ISO 27001 A.8.2, A.8.15	PARTIAL SINCE V0.1.0-RC.2	sneakers-sshbroker/internal/session/store_test.go (TestCreateAndConsumeOnce, TestZeroizeWipesKey) sneakers-sshbroker/internal/wsproxy/hostkey_test.go (TestHandlerConnectsWhenHostKeyPinned) Gap: sneakers-sshbroker#21 Session recording Gap: sneakers-sshbroker#26 Reject agent principals
ACC-7 Values on screen The web UI masks values until revealed and tells the user each reveal is logged. Hiding is manual: there is no auto-hide timer and nothing clears the clipboard after a copy. ISO 27001 A.8.12 · ASVS V8.2.1	PARTIAL SINCE V0.1.0-RC.2	sneakers-web/apps/staff/app/features/secret/SensitiveValue.tsx

7. Tamper-evident audit trail

Every reveal, check-out, approval, break-glass and admin action is written to the audit service as a record whose SHA-256 hash covers its fields and the previous record's hash. VerifyChain walks the whole chain and reports the first gap, broken link or wrong hash. Records carry which field was revealed, never the value.

> The chain detects edits by anyone without database write access. It is keyless and unsigned and the table doesn't block UPDATE or DELETE, so someone with database write access could rewrite it consistently. Routine audit writes are best-effort; only break-glass fails closed. A signed, encrypted archive for offline keeping is planned.

Controls: Audit trail

CONTROL	STATUS	EVIDENCE
AUD-1 Hash-chained audit records Each record's SHA-256 covers its sequence, action, actor, subject, attributes, time and the previous record's hash. VerifyChain reports a gap, a wrong link or a wrong hash, walking the chain in pages so it scales. SOC 2 CC7.2, CC4.1 · ISO 27001 A.8.15, A.5.33 · ASVS V7.3.3	SHIPPED IN V0.1.0-RC.2	sneakers-audit/internal/chain/chain.go sneakers-audit/internal/grpcsvc/audit_test.go (TestChainLinksAndVerifies, TestTamperBreaksChain)
AUD-2 Who may write and read the trail Only listed services may record, only the gateway may read, the migration job may only record and verify, and the appliance may only write its own admin events. The service won't start without an issuer. SOC 2 CC6.1 · ISO 27001 A.8.15 · ASVS V7.3.4	SHIPPED IN V0.1.0-RC.2	sneakers-audit/internal/grpcsvc/callers_test.go (TestCallerAuth_ReadsAreForTheGatewayOnly)
AUD-3 Tamper evidence, not tamper resistance The chain is keyless and unsigned, and the database doesn't stop UPDATE or DELETE. It detects edits by anyone without database write access; someone with it could rewrite the chain and recompute every hash. Routine audit writes are best-effort (only break-glass fails closed), and there is no retention setting yet. SOC 2 CC7.2 · ISO 27001 A.8.15 · ASVS V7.3.3	PARTIAL SINCE V0.1.0-RC.2	sneakers-audit/migrations/0001_baseline.up.sql Gap: sneakers-audit#60 Signed audit archive Gap: sneakers-audit#17 Run more than one instance safely Gap: sneakers-audit#18 Record vendor sign-ins and denials
AUD-4 No secret values in logs or notices Audit records hold metadata only; reveals record which field, never its value. Notifications have no field that could carry a value. Services and the MCP never log tokens or values. The audit service relies on its callers for this and doesn't check it server-side. SOC 2 C1.1 · ISO 27001 A.8.11, A.8.15 · ASVS V7.1.1, V7.1.2	SHIPPED IN V0.1.0-RC.1	sneakers-notify/proto/sneakers/notify/v1/notify.proto sneakers-connector/internal/tokensource/tokensource_test.go (TestFileNeverLogsToken)

8. MCP server and machine API

Agents get no shortcuts. The MCP server holds no credentials and makes no access decisions: it checks the caller's token, fails closed when it can't, and forwards that token to the gateway's machine API, where the vault applies the same rules, approvals and audit as for a person. Agents and tokens can never use break-glass and can never decide an approval.

An agent with access does receive plaintext values; that is the point of giving it access. Approval-gated secrets return a link for a person to approve, super-sensitive fields are withheld from every token unless an admin allows it, and the MCP add-on is off by default.

Controls: MCP and machine API

CONTROL	STATUS	EVIDENCE
MCP-1 Agents sign in with their own token The MCP bridge accepts OAuth JWTs (signature, issuer, audience and expiry checked locally), service-account tokens and personal tokens. Opaque tokens are confirmed by the gateway and cached for at most 60 seconds by hash; any failure is a 401 and is never cached. If the gateway is down, it fails closed. SOC 2 CC6.1 · ISO 27001 A.8.5 · ASVS V3.5.3, V13.1.3	SHIPPED IN V0.1.0-RC.1	sneakers-mcp/docs/configuration.md sneakers-mcp/internal/authn/authn_test.go (TestAPITokenGatewayDownFailsClosed, TestAPITokenRevokedIsRejectedOnceCacheExpires)
MCP-2 Same rules and audit as a person The bridge holds no credentials and makes no access decisions; it forwards the caller's own token to the machine API, where the vault applies the same folder rules, approvals and audit. Consent to an agent's sign-in needs a recent second factor, using PKCE with S256. SOC 2 CC6.1, CC6.3 · ISO 27001 A.5.15 · ASVS V4.1.1	SHIPPED IN V0.1.0-RC.1	sneakers-mcp/README.md sneakers-gateway/internal/bff/oauth.go
MCP-3 What an agent can receive By design an agent with access receives plaintext values: reads of plain fields, approved reveals (once, within 60 seconds) and new values from rotation. Approval-gated secrets return an approval link, not a value. Super-sensitive fields are withheld from every token unless an admin turns on API access to sensitive secrets (off by default). MCP is an add-on, off by default. SOC 2 C1.1 · ISO 27001 A.8.3, A.8.12	SHIPPED IN V0.1.0-RC.1	sneakers-mcp/internal/tools/tools_reveal_test.go (TestGetSecretOfAnApprovalSecretReturnsTheApprovalLinkNotAValue) sneakers-vault/internal/vault/grpcsvc/sensitive_settings_test.go (TestAPIForSensitiveOffKeepsSuperSensitiveFieldsFromMachines) Gap: sneakers-mcp#17 Personal tokens of a disabled user

9. Appliance: Secure Boot, TPM and signed updates

The appliance is a single-node Kubernetes box built from a declared file list rather than a Linux distribution. Its root is a read-only, dm-verity-checked SquashFS whose hash is inside a signed kernel image, booted under Secure Boot with project-owned keys and no shim.

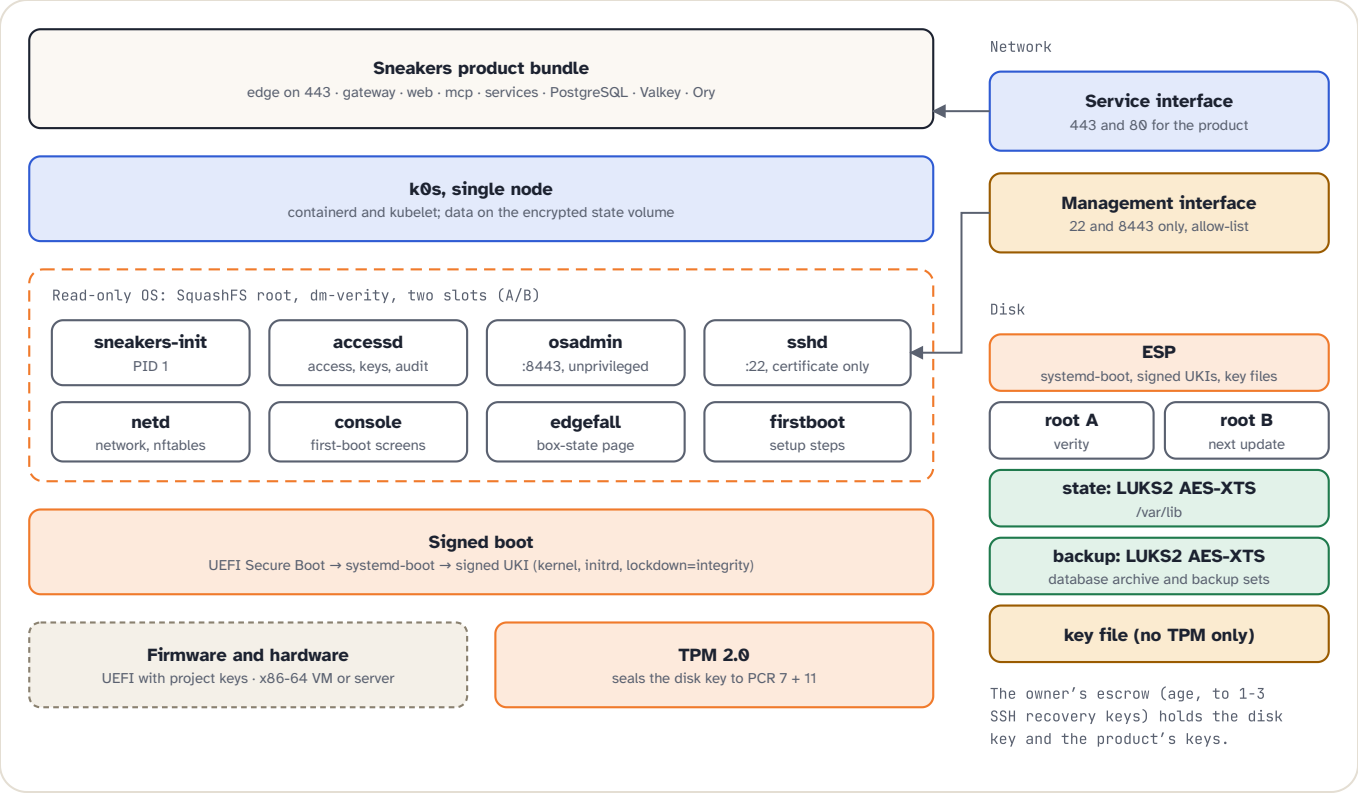


Figure 2. Appliance high-level design: the layers from firmware to product, the two network interfaces and the disk.

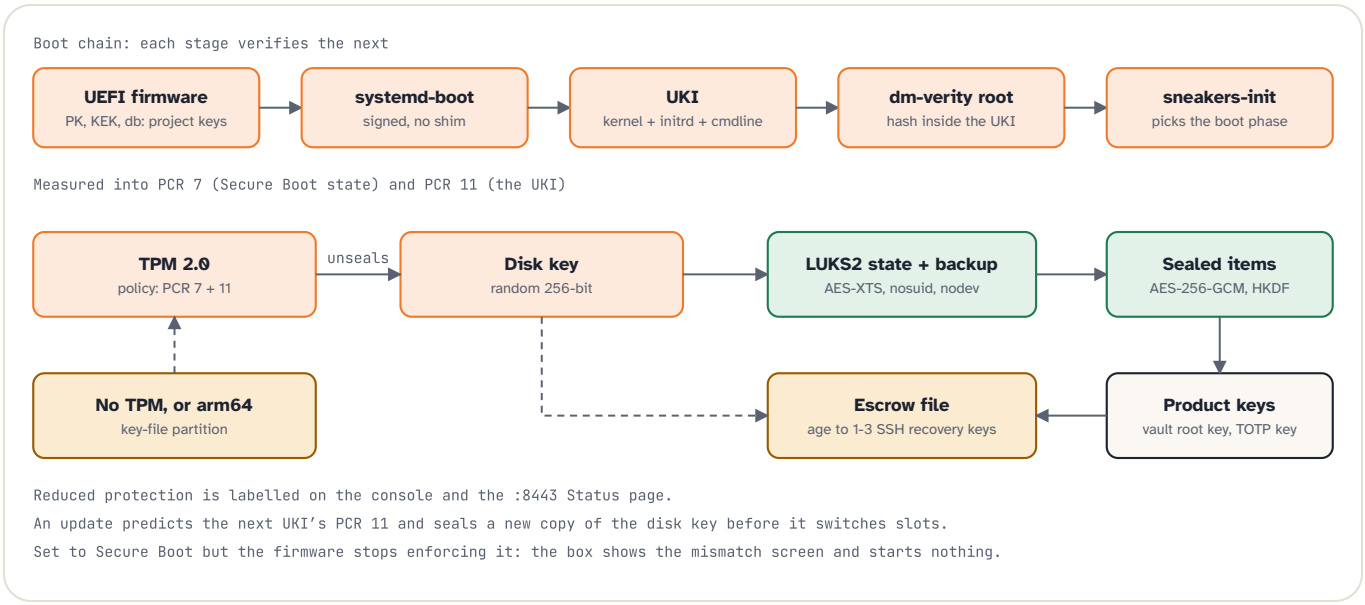


Figure 3. Appliance low-level design: the verified boot chain, the TPM-sealed disk key and what the escrow carries.

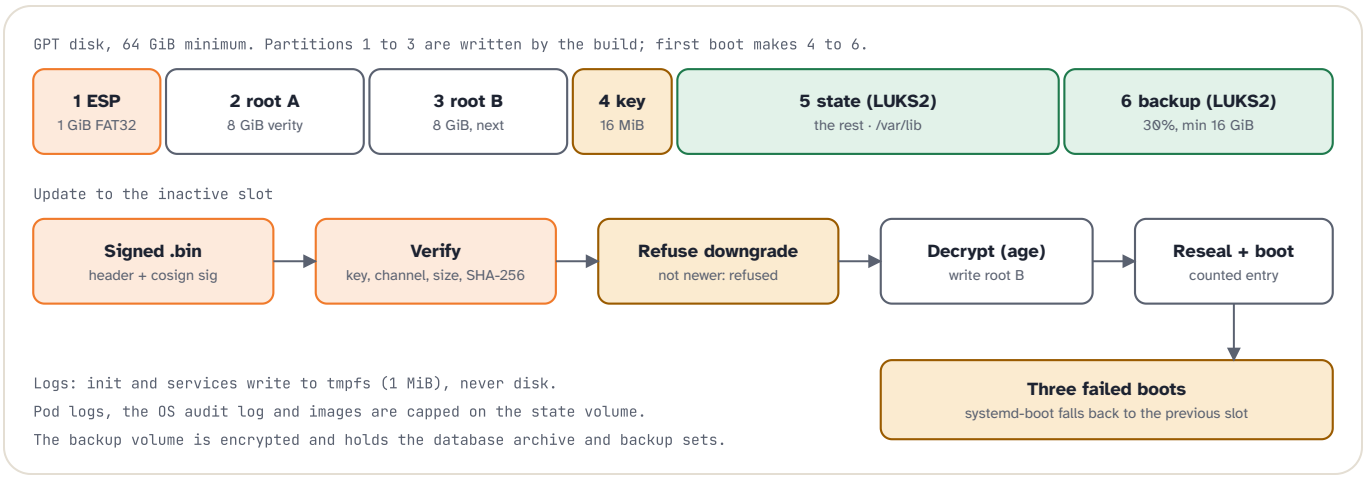


Figure 4. Appliance low-level design: the GPT layout and the A/B update path.

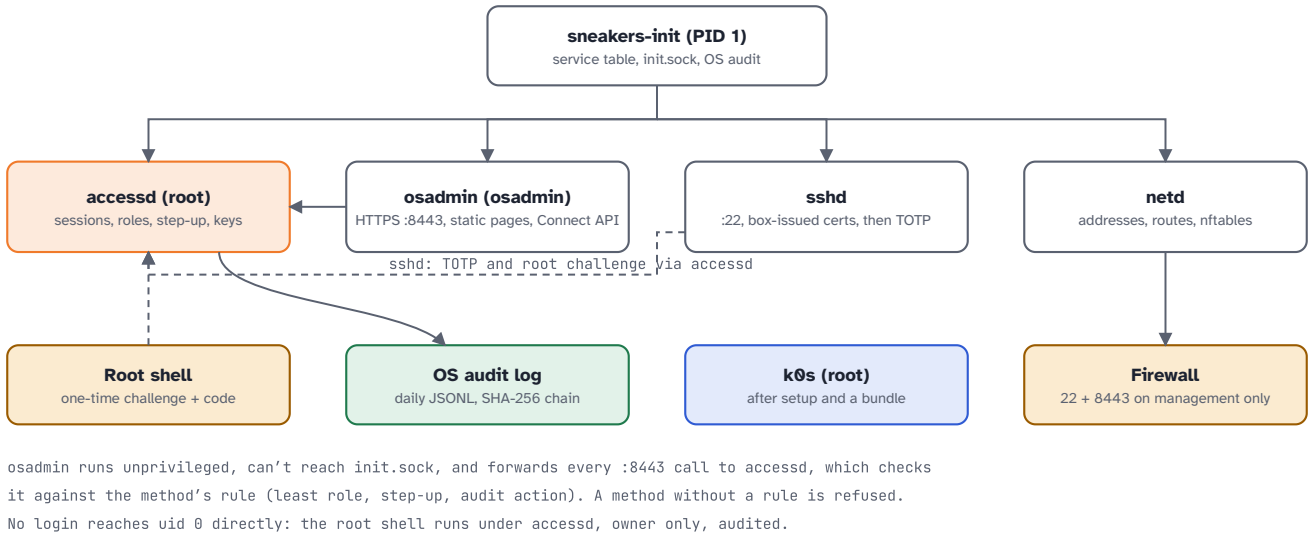


Figure 5. Appliance low-level design: OS processes, the users they run as, and where every privileged action is checked.

BOOT CHAIN

1. UEFI firmware, enrolled with the project's own PK, KEK and db keys, verifies the signed systemd-boot.
2. systemd-boot verifies one signed UKI holding the kernel, initrd and command line (lockdown=integrity).
3. Early boot unseals the disk key from the TPM against PCR 7 and 11 and opens the LUKS2 state volume.
4. dm-verity checks every block of the read-only root against the hash in the signed UKI; a changed block halts the box.

UPDATE

1. The box checks the update's cosign signature against the release key built into the running OS.
2. It checks the channel (lab or production), size and SHA-256, and refuses anything not newer than what runs.
3. It decrypts the payload, writes the inactive slot and seals a fresh disk-key copy to the new release's predicted PCR 11.
4. It boots the new slot; three failed boots fall back to the previous release.

> Without a TPM (including arm64 and Raspberry Pi), or with Secure Boot turned off, the box runs in a labelled reduced-protection mode. SSH is certificate-only with a TOTP code, and the firewall opens only SSH and the admin page, only on the management interface.

Controls: Appliance platform

CONTROL	STATUS	EVIDENCE
APL-1 Minimal, reproducible, read-only OS The appliance isn't a Linux distribution: it is built from a declared file list (a pinned kernel, busybox, OpenSSH and the project's own programs) with its own init as PID 1 and no setuid files. The root is a read-only SquashFS checked block by block with dm-verity, and the build is reproducible byte for byte. SOC 2 CC6.8, CC7.1 · ISO 27001 A.8.9, A.8.19	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/root-image.md sneakers-appliance/build/root/build_test.sh Gap: sneakers-appliance#45 Build the kernel and static tools from pinned sources
APL-2 Secure Boot with project-owned keys Firmware boots a project-signed systemd-boot and one signed kernel image (UKI) with no shim and no vendor keys. The kernel runs in lockdown, and the dm-verity root hash is inside the signed image. If the box is set to Secure Boot and the firmware stops enforcing it, the product won't start. An admin can choose reduced protection, which is labelled. SOC 2 CC6.8 · ISO 27001 A.8.9, A.8.19 · ASVS V10.3.2	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/secure-boot.md sneakers-appliance/.github/workflows/image-e2e.yml (QEMU with Secure Boot firmware and a software TPM) Gap: sneakers-appliance#413 Production Secure Boot enrolment without a manual key reset Gap: sneakers-appliance#62 Turn Secure Boot on or off later and reseal
APL-3 TPM-sealed disk encryption The state and backup partitions are LUKS2 (AES-XTS) under one random 256-bit key, sealed in the TPM to the boot measurements (PCR 7 and 11 with Secure Boot). Without a TPM, including on arm64, the key falls back to a key-file partition, shown as reduced protection. There is no remote attestation. SOC 2 CC6.1, C1.1 · ISO 27001 A.8.24, A.7.10	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/key-custody.md sneakers-appliance/internal/tpm
APL-4 Signed, encrypted, A/B updates An update is checked against the release key built into the running OS, then its channel, size and SHA-256, before it is decrypted. Anything not newer than the running release is refused. Updates install to the inactive slot; three failed boots fall back to the previous release. SOC 2 CC8.1, CC7.1 · ISO 27001 A.8.19, A.8.32 · ASVS V10.3.2	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/upgrades.md sneakers-appliance/internal/product/slots_test.go (downgrade refusal) Gap: sneakers-appliance#72 Rotate the update key

CONTROL	STATUS	EVIDENCE
APL-5 Certificate-only SSH and a closed firewall SSH accepts only box-issued keys with their certificates, then a TOTP code; no passwords and no root login. A root shell needs a one-time challenge from the admin page. The firewall opens SSH and the admin page only on the management interface, step by step during first boot, and can limit them to an allow-list. SOC 2 CC6.1, CC6.6 · ISO 27001 A.8.2, A.8.5, A.8.20	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/ssh-and-elevation.md sneakers-appliance/docs/network.md
APL-6 Hash-chained OS audit log Appliance administration is written to a daily JSONL log on the encrypted volume, each line carrying the previous line's SHA-256. It isn't forwarded off the box yet. SOC 2 CC7.2 · ISO 27001 A.8.15	PARTIAL SINCE V0.1.0-RC.3	sneakers-appliance/docs/os-audit.md Gap: sneakers-appliance#103 Forward the OS audit log

10. Supply chain and releases

Releases are built only from a tag the maintainer creates. Production signing keys are released to CI only for version tags, after the maintainer's approval. Every release asset is signed with cosign, and service images are published for linux/amd64 and linux/arm64.

- Verify a download: sha256sum -c SHA256SUMS, then cosign verify-blob --key with the published release key (see the appliance's release documentation).
- Signatures use a project-held key and aren't written to a public transparency log.
- No SBOM or SLSA provenance is published yet; the NIST SSDF self-assessment that will close this is tracked in sneakers-release#69, and a NIST SP 800-53 mapping in sneakers-release#70.

Controls: Supply chain and releases

CONTROL	STATUS	EVIDENCE
SUP-1 Signed release artifacts Every release asset (OS image, update headers, release manifest, checksums and a signature per pinned image digest) is signed with cosign using a project-held ECDSA P-256 key. Signatures aren't in a public transparency log, and image signatures are files on the release rather than attached in the registry. SOC 2 CC8.1 · ISO 27001 A.8.32 · ASVS V10.3.2, V14.2.4	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/.github/workflows/release.yml (sign job) sneakers-appliance/docs/release.md
SUP-2 Protected release keys Production signing keys live only in a protected CI environment that needs the maintainer's approval and accepts only version tags. CI refuses lab keys in a release, keys sit on tmpfs only for the step that uses them, and every other CI run uses fresh lab keys. SOC 2 CC6.1, CC8.1 · ISO 27001 A.8.24, A.8.31	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/runbooks/production-keys.md
SUP-3 Multi-arch service images Service images are built natively for linux/amd64 and linux/arm64 and published as one image index, so the same release runs on a Raspberry Pi. Appliance update and install images are amd64 only. ISO 27001 A.8.32	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/.github/workflows/release.yml (images job)
SUP-4 SBOM and build provenance No SBOM (SPDX or CycloneDX) or SLSA provenance is published yet. This is the main supply-chain gap. SOC 2 CC7.1, CC9.2 · ISO 27001 A.5.21, A.8.30 · ASVS V14.2.5	PLANNED	Gap: sneakers-release#69 NIST SSDF (SP 800-218) self-assessment

CONTROL	STATUS	EVIDENCE
SUP-5 A signed OVA is booted before it publishes The release build boots the production-signed OVA from fresh in QEMU, after signing and before anything publishes. If that boot fails, no files and no images go out. SOC 2 CC8.1 · ISO 27001 A.8.29, A.8.32	SHIPPED IN V0.1.0-RC.6	sneakers-appliance/.github/workflows/release.yml (Fresh OVA first boot) sneakers-appliance/test/image/ova/ova_test.go (TestFreshOVAFirstBoot)

11. Backups and recovery

Recovery today rests on two things: the key escrow, which lets a replacement box open restored data, and sneakers-migrate, which exports secrets, folders, users and settings as an age-encrypted archive and leaves the audit trail on the source. From v0.1.0-rc.7 the database also archives itself to the box's encrypted backup partition, takes a base backup nightly and can be restored to a named backup set on the same box. Offsite and scheduled copies off the box are not built.

> Without a downloaded escrow, losing the TPM or the box loses the data. Download the escrow after the product is installed, and again after anything that changes its keys.

Controls: Backup and recovery

CONTROL	STATUS	EVIDENCE
REC-1 Key escrow to the owner's recovery keys The owner downloads an escrow file holding the disk key and the product's keys (the vault root key and the TOTP key), encrypted with age to one to three SSH recovery keys. A replacement box imports it at first boot so restored data opens. SOC 2 A1.2 · ISO 27001 A.8.13, A.8.24	SHIPPED IN V0.1.0-RC.3	sneakers-appliance/docs/key-custody.md (Sealed items and the escrow)
REC-2 Encrypted migration bundles sneakers-migrate exports secrets, folders, users and settings as an age-encrypted archive. The audit trail never moves: it stays on the source, and the target's chain starts with one "imported from" record. Import refuses any stream that doesn't match its checksum or count, and the path is rehearsed in CI. SOC 2 A1.2, A1.3 · ISO 27001 A.8.13	SHIPPED IN V0.1.0-RC.3	sneakers-release/docs/migrate.md sneakers-release/.github/workflows/migrate-rehearsal.yml
REC-3 Scheduled backups The database archives its write-ahead log to the box's encrypted backup partition and takes a base backup nightly, keeping seven, with a nightly restore check. An admin cuts a named backup set over SSH, an owner can export it encrypted to the recovery keys, and an owner restores it with a new authenticator code. A set restores only on the box that made it. Nothing is pushed off the box on a schedule, so a lost box still relies on the escrow and a migration export. SOC 2 A1.2, A1.3 · ISO 27001 A.8.13, A.5.30	PARTIAL SINCE V0.1.0-RC.7	sneakers-appliance/docs/backups.md sneakers-appliance/docs/data-move.md (Moving the database to helm-postgres-ha) Gap: sneakers-appliance#462 Scheduled offsite backups Gap: sneakers-appliance#106 Backup lifecycle Gap: sneakers-appliance#537 Resources page for the PostgreSQL settings

12. Logging, monitoring and health

Each service reports readiness from its real dependencies and keeps liveness to the process. Logs are JSON in every cluster, at a level set per environment (error in production by default), and never include secrets, tokens or personal data. Traces, metrics and, if you turn them on, logs can go to your OpenTelemetry collector, set on the :8443 Telemetry page, with its own CA if it uses a private one.

Controls: Logging, monitoring and health

CONTROL	STATUS	EVIDENCE
OPS-1 Readiness that follows real dependencies Each service's gRPC health and /readyz fail while a required dependency (its database, gRPC peers, the workload-identity key set) is down; liveness checks the process only. SOC 2 A1.1, CC7.2 · ISO 27001 A.8.16	SHIPPED IN V0.1.0-RC.1	sneakers-vault/internal/server/health_test.go (TestHealth_ReadinessFollowsRequiredDependencyLivenessDoesNot)
OPS-2 Structured logs and telemetry Services log JSON at a level set per environment (production defaults to error) and export traces, metrics and, if you turn them on, logs over OTLP to a collector you set on the :8443 Telemetry page. A collector with a private CA can be given that CA, and no logs leave while its certificate isn't verified. The appliance doesn't forward its own logs and there is no built-in alerting yet. SOC 2 CC7.2, CC7.3 · ISO 27001 A.8.15, A.8.16	PARTIAL SINCE V0.1.0-RC.3	sneakers-release/docs/values.md Gap: sneakers-appliance#113 Forward logs to syslog or OTLP
OPS-3 Redacted support bundle An admin can take one diagnostic bundle over SSH or from the :8443 Logs page. The box runs a fixed set of read-only commands, redacts every line before it's written, shows secrets and environment variables as names only, and records each bundle in the appliance's audit log. Host names, addresses and account names stay in the bundle, so read it before sending it to anyone. SOC 2 CC7.2 · ISO 27001 A.8.15	SHIPPED IN V0.1.0-RC.5	sneakers-appliance/docs/support-bundle.md sneakers-appliance/internal/supportbundle/redact_test.go (TestRedactionRemovesEverySecretShape)

13. Secure development and vulnerability handling

Every change lands through a pull request that passes build and test, gosec, govulncheck, licence checks and a scrub for secrets and identifying details. Tests specify behaviour through public APIs, against real Postgres, real sshd and network namespaces, and the appliance is booted end to end in QEMU with Secure Boot and a software TPM.

Report vulnerabilities privately through GitHub's private vulnerability reporting on the affected repository. Fixes target the latest released minor version.

Controls: Secure development and vulnerability handling

CONTROL	STATUS	EVIDENCE
DEV-1 Gated changes on protected branches main accepts only squash-merged pull requests that pass the required checks (build and test, gosec, govulncheck, licence headers, dependency licences and the scrub). Force-push and deletion are blocked. No second reviewer is required: Sneakers has one maintainer today. SOC 2 CC8.1 · ISO 27001 A.8.25, A.8.32	SHIPPED IN V0.1.0-RC.1	.github/.github/workflows/scrub.yml Gap: sneakers-release#23 A required release-review group
DEV-2 Vulnerability and secret scanning govulncheck and gosec run on every pull request and weekly; gitleaks and pattern checks run in the scrub. There is no CodeQL, container image scan or automated dependency update yet, and actions are pinned by commit only in the appliance repo. SOC 2 CC7.1 · ISO 27001 A.8.8, A.8.28 · ASVS V14.2.1	PARTIAL SINCE V0.1.0-RC.1	sneakers-vault/.github/workflows/job-go-security.yaml Gap: sneakers-release#51 Excluded govulncheck finding awaiting an upstream fix
DEV-3 Test-first changes across layers Behaviour is specified by tests through public APIs, with real Postgres, real sshd and network namespaces where it matters, and the appliance is exercised end to end in QEMU after each merge and nightly. SOC 2 CC8.1 · ISO 27001 A.8.29	SHIPPED IN V0.1.0-RC.1	sneakers-appliance/docs/testing.md
DEV-4 Private vulnerability reporting Vulnerabilities are reported privately through GitHub on the affected repository, acknowledged within a few business days, fixed and disclosed on a timeline agreed with the reporter, and credited in the advisory. A signed advisory feed for appliances is planned. SOC 2 CC2.3, CC7.3, CC7.4 · ISO 27001 A.5.24, A.5.26, A.8.8	SHIPPED IN V0.1.0-RC.1	.github/.github/SECURITY.md Gap: sneakers-appliance#330 Signed vulnerability-advisory feed

Gaps and planned work

Every control that is partial or planned, with the open issue that tracks it. Nothing here is promised for a date; each issue carries its own milestone.

CONTROL	STATUS	EVIDENCE
ARC-3 Network policy per call-graph edge The Helm charts install a NetworkPolicy for each service that admits only its declared callers, plus policies for the bundled identity provider and database. The install test checks that blocked pods are blocked. Outbound traffic isn't restricted by default. SOC 2 CC6.6 · ISO 27001 A.8.20, A.8.22	PARTIAL SINCE V0.1.0-RC.3	sneakers-release/charts/sneakers-lib/templates/_resources.tpl sneakers-release/.github/workflows/install-test.yml
CRY-4 Root key custody The vault reads its root key from the environment at start and holds it in memory. On the appliance the key lives on the encrypted state volume, whose key is sealed in the TPM, and is copied into the owner's escrow. The root key itself can't be rotated, and there is no HSM or cloud KMS provider yet. SOC 2 CC6.1 · ISO 27001 A.8.24 · ASVS V6.4.1	PARTIAL SINCE V0.1.0-RC.3	sneakers-appliance/docs/key-custody.md (Product keys) sneakers-vault/cmd/vault/main.go Gap: sneakers-vault#41 HSM and cloud KMS key providers
CRY-6 Metadata at rest Only field values are application-encrypted. Secret names, folders, rules, targets and settings are stored as plain JSON in each service's database. On the appliance that database sits on the LUKS2-encrypted state volume; on your own Kubernetes, disk encryption is yours to provide. SOC 2 C1.1 · ISO 27001 A.8.24	PARTIAL SINCE V0.1.0-RC.1	sneakers-vault/migrations/vault/0001_baseline.up.sql
CRY-7 Encryption in transit between services Internal gRPC calls are authenticated (ARC-2) but not encrypted: every client dials without TLS, and the gateway serves plain HTTP behind the cluster ingress. Confidentiality on the wire depends on the platform today (a private network or a service mesh with mTLS). SOC 2 CC6.7 · ISO 27001 A.8.24, A.8.20 · ASVS V9.2.2	PLANNED	sneakers-gateway/cmd/gateway/backend.go sneakers-gateway/README.md (Before production) Gap: sneakers-appliance#531 Issue certificates through ACME with cert-manager
IDN-3 TOTP replay and guessing limits A pending second-factor sign-in ends after five wrong codes. TOTP verification itself doesn't yet refuse a code reused inside its period, and email codes don't prove possession of a device, so they don't meet SP 800-63B's out-of-band rules. 800-63B 5.1.4.2, 5.2.2, 5.1.3.1 · ASVS V2.2.1, V2.8.4	PARTIAL SINCE V0.1.0-RC.1	sneakers-gateway/internal/bff/mfa.go sneakers-gateway/internal/bff/stepup_test.go (TestMfaStepUp_AttemptBudgetEndsTheSession)

CONTROL	STATUS	EVIDENCE
IDN-8 Single sign-on and passkey sign-in SAML sign-on runs through the bundled Ory Polis at the gateway. Passkeys work as a second factor, but passwordless passkey sign-in isn't built. SOC 2 CC6.1 · ISO 27001 A.8.5 · 800-63B 5.1.8, 5.2.5	PARTIAL SINCE V0.1.0-RC.1	sneakers-gateway/internal/bff/sso.go Gap: sneakers-identity#22 Passkey sign-in
ACC-4 Rotation without agents on targets The connector pulls due heartbeat and rotation jobs and changes passwords over LDAP/LDAPS and Kerberos using the account's own current password, so it needs no admin rights. A heartbeat reports an untrusted certificate as its own result, with the reason. The directory's built-in Administrator is matched by SID and never rotated. SSH, WinRM and SAMR rotation aren't built. SOC 2 CC6.1 · ISO 27001 A.5.17	PARTIAL SINCE V0.1.0-RC.1	sneakers-connector/README.md sneakers-connector/internal/adapter/ldap_account_test.go (TestRotateADBuiltinAdministratorIsRefusedWithoutChange) Gap: sneakers-connector#16 SSH, WinRM and SAMR rotation Gap: sneakers-connector#33 Respect directory password rules when rotating Gap: sneakers-connector#75 Pin LDAPS certificates for an enterprise CA
ACC-6 Brokered SSH without handing over the key In-browser SSH uses a single-use ticket that lives 30 seconds and holds no key material. The broker fetches the key as the user when the browser connects, wipes it after use, connects only to pinned host keys and cuts sessions at 8 hours. Sessions aren't recorded yet. SOC 2 CC6.1 · ISO 27001 A.8.2, A.8.15	PARTIAL SINCE V0.1.0-RC.2	sneakers-sshbroker/internal/session/store_test.go (TestCreateAndConsumeOnce, TestZeroizeWipesKey) sneakers-sshbroker/internal/wsproxy/hostkey_test.go (TestHandlerConnectsWhenHostKeyPinned) Gap: sneakers-sshbroker#21 Session recording Gap: sneakers-sshbroker#26 Reject agent principals
ACC-7 Values on screen The web UI masks values until revealed and tells the user each reveal is logged. Hiding is manual: there is no auto-hide timer and nothing clears the clipboard after a copy. ISO 27001 A.8.12 · ASVS V8.2.1	PARTIAL SINCE V0.1.0-RC.2	sneakers-web/apps/staff/app/features/secret/SensitiveValue.tsx
AUD-3 Tamper evidence, not tamper resistance The chain is keyless and unsigned, and the database doesn't stop UPDATE or DELETE. It detects edits by anyone without database write access; someone with it could rewrite the chain and recompute every hash. Routine audit writes are best-effort (only break-glass fails closed), and there is no retention setting yet. SOC 2 CC7.2 · ISO 27001 A.8.15 · ASVS V7.3.3	PARTIAL SINCE V0.1.0-RC.2	sneakers-audit/migrations/0001_baseline.up.sql Gap: sneakers-audit#60 Signed audit archive Gap: sneakers-audit#17 Run more than one instance safely Gap: sneakers-audit#18 Record vendor sign-ins and denials

CONTROL	STATUS	EVIDENCE
APL-6 Hash-chained OS audit log Appliance administration is written to a daily JSONL log on the encrypted volume, each line carrying the previous line's SHA-256. It isn't forwarded off the box yet. SOC 2 CC7.2 · ISO 27001 A.8.15	PARTIAL SINCE V0.1.0-RC.3	sneakers-appliance/docs/os-audit.md Gap: sneakers-appliance#103 Forward the OS audit log
SUP-4 SBOM and build provenance No SBOM (SPDX or CycloneDX) or SLSA provenance is published yet. This is the main supply-chain gap. SOC 2 CC7.1, CC9.2 · ISO 27001 A.5.21, A.8.30 · ASVS V14.2.5	PLANNED	Gap: sneakers-release#69 NIST SSDF (SP 800-218) self-assessment
REC-3 Scheduled backups The database archives its write-ahead log to the box's encrypted backup partition and takes a base backup nightly, keeping seven, with a nightly restore check. An admin cuts a named backup set over SSH, an owner can export it encrypted to the recovery keys, and an owner restores it with a new authenticator code. A set restores only on the box that made it. Nothing is pushed off the box on a schedule, so a lost box still relies on the escrow and a migration export. SOC 2 A1.2, A1.3 · ISO 27001 A.8.13, A.5.30	PARTIAL SINCE V0.1.0-RC.7	sneakers-appliance/docs/backups.md sneakers-appliance/docs/data-move.md (Moving the database to helm-postgres-ha) Gap: sneakers-appliance#462 Scheduled offsite backups Gap: sneakers-appliance#106 Backup lifecycle Gap: sneakers-appliance#537 Resources page for the PostgreSQL settings
OPS-2 Structured logs and telemetry Services log JSON at a level set per environment (production defaults to error) and export traces, metrics and, if you turn them on, logs over OTLP to a collector you set on the :8443 Telemetry page. A collector with a private CA can be given that CA, and no logs leave while its certificate isn't verified. The appliance doesn't forward its own logs and there is no built-in alerting yet. SOC 2 CC7.2, CC7.3 · ISO 27001 A.8.15, A.8.16	PARTIAL SINCE V0.1.0-RC.3	sneakers-release/docs/values.md Gap: sneakers-appliance#113 Forward logs to syslog or OTLP
DEV-2 Vulnerability and secret scanning govulncheck and gosec run on every pull request and weekly; gitleaks and pattern checks run in the scrub. There is no CodeQL, container image scan or automated dependency update yet, and actions are pinned by commit only in the appliance repo. SOC 2 CC7.1 · ISO 27001 A.8.8, A.8.28 · ASVS V14.2.1	PARTIAL SINCE V0.1.0-RC.1	sneakers-vault/.github/workflows/job-go-security.yaml Gap: sneakers-release#51 Excluded govulncheck finding awaiting an upstream fix

Framework crosswalk

Each control against the framework references it supports. A reference means the control contributes evidence toward that criterion, not that the criterion is fully met. SP 800-63B references use Revision 3 section numbers.

CONTROL	SOC 2 TSC	ISO 27001:2022 ANNEX A	NIST SP 800-63B	OWASP ASVS 4.0.3
ARC-1 Single entry point for people and machines	CC6.1, CC6.6	A.8.20, A.8.22		V1.4.1
ARC-2 Workload identity on every internal call	CC6.1, CC6.3	A.5.15, A.8.5		V1.4.4, V4.1.3
ARC-3 Network policy per call-graph edge	CC6.6	A.8.20, A.8.22		
ARC-4 Hardened containers	CC6.8	A.8.9		V14.1.1
CRY-1 Envelope encryption of every secret field	CC6.1, C1.1	A.8.24		V6.2.2, V6.3.1
CRY-2 Three-tier key hierarchy with a rotating key ring	CC6.1	A.8.24		V6.4.1, V6.4.2
CRY-3 Fail-closed key start-up	CC6.1	A.8.24		V6.2.1
CRY-4 Root key custody	CC6.1	A.8.24		V6.4.1
CRY-5 Second-factor secrets and tokens at rest	CC6.1	A.5.17, A.8.24		V2.9.1, V6.1.1
CRY-6 Metadata at rest	C1.1	A.8.24		
CRY-7 Encryption in transit between services	CC6.7	A.8.24, A.8.20		V9.2.2
CRY-8 Export-compliant, standard cryptography		A.8.24, A.5.31		V6.2.2
IDN-1 Passwords held by a dedicated identity provider	CC6.1	A.5.17, A.8.5	5.1.1.2	V2.4.1, V2.5.2
IDN-2 Second factor required for every session	CC6.1	A.8.5	4.2.1, 5.1.4, 5.1.8	V2.7, V2.8, V4.3.1
IDN-3 TOTP replay and guessing limits			5.1.4.2, 5.2.2, 5.1.3.1	V2.2.1, V2.8.4
IDN-4 Server-side sessions in a hardened cookie	CC6.1	A.8.5	7.1, 4.2.3	V3.2.1, V3.3.1, V3.4.1, V3.4.2, V3.4.3
IDN-5 Step-up for sensitive actions	CC6.1	A.8.5	7.2	V3.7.1
IDN-6 CSRF and cross-origin protection		A.8.26		V4.2.2, V13.2.3, V14.5.3
IDN-7 Browser security headers		A.8.26		V14.4.3, V14.4.4, V14.4.7
IDN-8 Single sign-on and passkey sign-in	CC6.1	A.8.5	5.1.8, 5.2.5	
AUZ-1 Deny-by-default folder rules	CC6.1, CC6.3	A.5.15, A.5.18, A.8.3		V4.1.3, V4.1.5

CONTROL	SOC 2 TSC	ISO 27001:2022 ANNEX A	NIST SP 800-63B	OWASP ASVS 4.0.3
AUZ-2 Approvals no one grants to themselves	CC6.3	A.5.3, A.8.2		V4.1.3
AUZ-3 One task, one prompt	CC6.3	A.8.2		
AUZ-4 Break-glass for people only	CC6.1, CC6.3	A.8.2, A.8.18		V4.3.1
AUZ-5 Recovery role for history	CC6.3	A.8.2, A.8.3		
ACC-1 Reveal one field, with checks in order	CC6.1, C1.1	A.8.3, A.8.12		V4.1.1, V8.3.4
ACC-2 Single-use, short-lived approved uses	CC6.1	A.8.3		V4.1.3
ACC-3 Check-out leases that rotate on check-in	CC6.1, CC6.3	A.5.18, A.8.2		
ACC-4 Rotation without agents on targets	CC6.1	A.5.17		
ACC-5 Strong generated passwords		A.5.17		V6.3.1
ACC-6 Brokered SSH without handing over the key	CC6.1	A.8.2, A.8.15		
ACC-7 Values on screen		A.8.12		V8.2.1
AUD-1 Hash-chained audit records	CC7.2, CC4.1	A.8.15, A.5.33		V7.3.3
AUD-2 Who may write and read the trail	CC6.1	A.8.15		V7.3.4
AUD-3 Tamper evidence, not tamper resistance	CC7.2	A.8.15		V7.3.3
AUD-4 No secret values in logs or notices	C1.1	A.8.11, A.8.15		V7.1.1, V7.1.2
MCP-1 Agents sign in with their own token	CC6.1	A.8.5		V3.5.3, V13.1.3
MCP-2 Same rules and audit as a person	CC6.1, CC6.3	A.5.15		V4.1.1
MCP-3 What an agent can receive	C1.1	A.8.3, A.8.12		
APL-1 Minimal, reproducible, read-only OS	CC6.8, CC7.1	A.8.9, A.8.19		
APL-2 Secure Boot with project-owned keys	CC6.8	A.8.9, A.8.19		V10.3.2
APL-3 TPM-sealed disk encryption	CC6.1, C1.1	A.8.24, A.7.10		
APL-4 Signed, encrypted, A/B updates	CC8.1, CC7.1	A.8.19, A.8.32		V10.3.2
APL-5 Certificate-only SSH and a closed firewall	CC6.1, CC6.6	A.8.2, A.8.5, A.8.20		
APL-6 Hash-chained OS audit log	CC7.2	A.8.15		
SUP-1 Signed release artifacts	CC8.1	A.8.32		V10.3.2, V14.2.4
SUP-2 Protected release keys	CC6.1, CC8.1	A.8.24, A.8.31		
SUP-3 Multi-arch service images		A.8.32		

CONTROL	SOC 2 TSC	ISO 27001:2022 ANNEX A	NIST SP 800-63B	OWASP ASVS 4.0.3
SUP-4 SBOM and build provenance	CC7.1, CC9.2	A.5.21, A.8.30		V14.2.5
SUP-5 A signed OVA is booted before it publishes	CC8.1	A.8.29, A.8.32		
REC-1 Key escrow to the owner's recovery keys	A1.2	A.8.13, A.8.24		
REC-2 Encrypted migration bundles	A1.2, A1.3	A.8.13		
REC-3 Scheduled backups	A1.2, A1.3	A.8.13, A.5.30		
OPS-1 Readiness that follows real dependencies	A1.1, CC7.2	A.8.16		
OPS-2 Structured logs and telemetry	CC7.2, CC7.3	A.8.15, A.8.16		
OPS-3 Redacted support bundle	CC7.2	A.8.15		
DEV-1 Gated changes on protected branches	CC8.1	A.8.25, A.8.32		
DEV-2 Vulnerability and secret scanning	CC7.1	A.8.8, A.8.28		V14.2.1
DEV-3 Test-first changes across layers	CC8.1	A.8.29		
DEV-4 Private vulnerability reporting	CC2.3, CC7.3, CC7.4	A.5.24, A.5.26, A.8.8		