SECURITY EVIDENCE PACKET

Test the controls. Do not trust the checklist.

A practical validation guide for MSP owners, security leads, and IT administrators evaluating Nizlo on a controlled Windows endpoint.

14-day trial	Start with one lab endpoint	Challenge every boundary

Current vendor disclosures, concrete pilot tests, and explicit non-claims. Applying or starting a trial does not reserve a lifetime-discount position; successful paid-upgrade order determines position.

Identity and access controls

A trial should prove who can act, which tenant and endpoint they can reach, and how the endpoint decides whether work is authorized.

CONTROL	CURRENT NIZLO DISCLOSURE	PILOT EVIDENCE TO RETAIN
Tenant isolation	Tenant ownership is enforced in backend queries and database relationships, not only in browser filtering.	Use two test organizations and a limited technician. Check lists, direct URLs, searches, reports, jobs, and remote targets for cross-tenant leakage.
Technician authorization	Roles and action-specific permissions separate viewing from remote access, scripts, files, patching, and administration.	Attempt one allowed and one denied operation. Retain the visible result and audit evidence for both.
MFA and passkeys	TOTP MFA, recovery handling, WebAuthn/passkey registration, and session controls are implemented for the permanent application origin.	Enable MFA for all pilot technicians, register a passkey where appropriate, and review and revoke a test session.
Device identity	A single-purpose, hashed, revocable, expiring placement token creates installation-specific credentials. Windows uses machine-scoped DPAPI for the device secret.	Enroll one disposable endpoint, revoke or replace its placement token, and verify that token reuse does not become device authentication.
Transport	Enrollment and agent/API traffic use TLS. Wake messages contain bounded identifiers; authoritative work is fetched over authenticated HTTPS.	Observe a lab endpoint's destinations and confirm ordinary agent use requires no inbound firewall opening.
Signed jobs	Finite work is target-bound, server-signed, and includes authorization context, nonce, issue time, and expiry. The endpoint verifies it before execution.	Run one harmless job and retain target, timestamps, states, result, and audit entry. Let another harmless job expire and inspect the outcome.

Test principle: a polished interface is not an authorization boundary. Exercise the API-backed behavior through the product and keep the evidence.

Operational and data controls

The most useful proof is an explainable lifecycle: authorized request, bounded delivery, endpoint verification, visible result, and clean exit.

CONTROL	CURRENT NIZLO DISCLOSURE	PILOT EVIDENCE TO RETAIN
Auditability	Sensitive actions create tenant-scoped records with actor, target, source, result, and bounded metadata.	Perform one successful and one denied or failed operation. Confirm both can be reconstructed without exposing secrets.
Delivery durability	PostgreSQL is authoritative. Redis and WebSockets are disposable wake paths; polling provides recovery.	Disconnect a lab endpoint, queue a harmless expiring job, reconnect, and verify the final state is visible and explainable.
File operations	Browsing and transfers are permission separated, tenant/device bound, size limited, and constrained to validated Windows paths. Uploads are not auto-executed.	Browse a lab folder read-only, test a denied mutation, and inspect size/digest evidence for a non-sensitive test file.
Patch operations	Installation is permission gated, bound to exact Windows Update identities and a recent scan, constrained by policy, and does not auto-restart.	Install one approved test update in a maintenance window and retain per-update result and reboot-required state.
Remote provisioning	The reviewed RustDesk path fixes package metadata server-side and verifies HTTPS, size, SHA-256, Windows trust, signer pin, and version. It accepts no unattended password.	Use a disposable endpoint, compare the observed installed version with the approved version, and retain deployment and audit evidence.
Data boundaries	Nizlo documents bounded inventory and operational telemetry. It does not intentionally collect Windows, browser, Wi-Fi, or other stored user credentials.	Compare the documented inventory with captured lab traffic and console fields. Stop the pilot if an unexplained field appears.

Stop condition: if the target, authorization, collected data, result, or removal path cannot be explained, do not expand the pilot.

A 45-minute validation run

Start with an authorized, disposable Windows endpoint. This sequence is designed to reveal trust gaps before a broader evaluation.

1	Create two test organizations and two technician roles.	2	Require MFA for every pilot technician.
3	Enroll one disposable Windows endpoint with a dedicated placement token.	4	Confirm that the endpoint appears only inside its assigned organization.
5	Run one read-only inventory refresh and one harmless finite job.	6	Verify requester, target, timestamps, states, result, and audit entry.
7	Attempt the same action from the limited role and retain the denial evidence.	8	Disconnect the endpoint; queue a harmless short-lived job; inspect expiry or recovery.
9	Review every collected field and every external provider involved.	10	Remove the endpoint and revoke the test technician's access.

WHAT THIS PACKET DOES NOT CLAIM

No claim is made here for SOC 2, ISO 27001, FedRAMP, HIPAA certification, a published independent penetration test, or any assurance not explicitly identified on Nizlo's public pages. Architecture documentation is not proof by itself. PostgreSQL row-level security is described as planned defense in depth, not as a current authorization boundary.

FOUNDING TESTER OFFER

14 days free. First 5 successful paid upgrades: 75% off for life. Next 5: 50% off for life. Next 10: 25% off for life. Applying or starting a trial does not reserve a position.

app.nizlo.com/founding-testers

nizlo.com/security | Security checklist | Privacy | Support