

KNJ Panel

Penetration Test Summary

External, black-box assessment of a fresh KNJ Panel installation, with remediation and retest.

PRODUCT TESTED

KNJ Panel v0.20.9 (retested on v0.20.12; SEC-3 verified on v0.20.19)

TEST DATE

14 October 2026

PERSPECTIVE

External attacker. No server access and no prior knowledge of credentials or internals during testing

CLASSIFICATION

Public summary

TARGET

A dedicated, isolated virtual machine with a fresh install, reached through the panel's public web ports

1. Executive summary

KNJ Panel was installed on a dedicated virtual machine exactly as a customer would install it, then attacked from the outside through its public web interfaces. The test looked for ways to gain access, read or change another customer's data, escalate privileges, and learn more about the system than an anonymous visitor should.

Result: no way to bypass authentication, escalate privileges or reach another account's data was found. The assessment produced one install-time defect and eight hardening findings, none of them a direct compromise. Eight were fixed (six security items were retested on v0.20.12 and the phpMyAdmin item was verified on v0.20.19) and one was accepted as a deliberate trade-off. Nothing remains open.

9

findings in total

8

fixed and verified

1

accepted on purpose

0

still open

SEVERITY	COUNT	FIXED	ACCEPTED / OPEN
High (install blocker, not an exploit)	1	1	0
Medium	1	1	0
Low to Medium	1	1	0
Low	3	2	1 accepted
Informational	3	3	0

Two further observations were positive security controls and are listed in section 6.

2. Scope and method

In scope. The controller panel (port 2087), the account panel (port 2083), the default web vhost (ports 80 and 443) and the bundled phpMyAdmin entry point. Out of scope: the underlying operating system, other servers, denial-of-service testing and social engineering.

Method. Setup used the same steps as a customer: run the bootstrap script, complete the Initial Setup wizard, create an administrator, a package and a hosting account, then stop using server access. From that point every test was made over HTTPS from an outside network. Where the panel's route list was read from the project source to decide what to probe, that was used only to choose targets, not to skip a step an outsider would have to take. Tests covered:

- HTTP response headers, error pages and version disclosure
- Login: rate limiting, brute-force protection, user enumeration
- Forgot-password flow
- CSRF protection, session fixation and session invalidation on logout
- Unauthenticated endpoint discovery, including the API and internal routes
- After logging in as the administrator: HTTP verb tampering, direct object references, path traversal, mass assignment, and access to the API with a web session

Limits. This was a time-boxed assessment of one installation by a single tester. It did not include source-code review, dependency scanning, or testing of customer-hosted applications. Absence of findings is not proof of absence of vulnerabilities.

3. Findings summary

ID	FINDING	SEVERITY	STATUS
BOOTSTRAP-1	Installer aborts on cloud images with ext4 native quota	High	Fixed
SEC-5	Login-failure ban blocked every port, including SSH and customer sites	Medium	Fixed, retested
SEC-8	Forgot-password revealed which email addresses are registered	Low to Medium	Fixed, retested
SEC-2	Missing Content-Security-Policy, Referrer-Policy, Permissions-Policy	Low	Fixed (baseline CSP), retested
SEC-3	phpMyAdmin entry point reachable and identifiable on port 443	Low	Fixed, verified on v0.20.19
SEC-7	Session cookie uses SameSite=Lax rather than Strict	Low	Accepted
SEC-1	nginx version in the <code>Server</code> header	Info	Fixed, retested
SEC-9	nginx version in the port 443 error page	Info	Fixed, retested
SEC-10	API route existence revealed by 401 versus 404	Info	Fixed, retested

4. Detailed findings

BOOTSTRAP-1: Installer aborts on cloud images with ext4 native quota High

WHERE	<code>bootstrap-server.sh</code> , disk quota step
ISSUE	Ubuntu cloud images create the filesystem with ext4's built-in quota feature. The installer then tried to enable the older external quota files, the kernel refused ("Quota format not supported"), and the script stopped at step 2 of 14. A customer on a standard cloud image could not install the panel at all. This is an availability defect, not an exploitable vulnerability.
FIX	The installer detects native quota and skips the legacy quota steps, because the kernel already tracks usage through the mount options. Applied to both the root and home filesystem blocks.
STATUS	Fixed. Included in v0.20.10. The fix was applied before the test VM was provisioned.

SEC-5: Login-failure ban blocked every port Medium

WHERE	fail2ban jail <code>knjpanel-auth</code>
ISSUE	Five failed logins on the panel within ten minutes banned the source address from all ports for one hour: 2087, 2083, 80, 443 and SSH. Consequences: (1) a customer's public website becomes unreachable to a visitor who shares an address with someone mistyping a panel password, such as an office; (2) an attacker who can cause failed logins from a victim's address, for example behind a shared proxy, can deny that address access to the whole server; (3) an administrator who forgets their password can lock themselves out of SSH. Separately, unbanning an address does not clear its recent failure count, so a freshly unbanned user can be banned again after one or two more failures.
FIX	The jail now bans only ports 2082, 2083, 2086 and 2087. The upgrade restarts the jail when its live firewall rule still covers every port, because a plain reload keeps the old rule.
RETEST	After repeated failed logins from the test host, ports 2087 and 2083 refused connections while 443, 80 and SSH stayed reachable.
TRADE-OFF	The panel login reached through a customer's own domain on port 443 is no longer covered by the ban. It remains limited to five attempts per minute by the application.
STATUS	Fixed, retested on v0.20.12.

SEC-8: Forgot-password revealed registered addresses Low to Medium

WHERE	POST <code>/forgot-password</code> on port 2087
ISSUE	A registered address produced "We have emailed your password reset link." An unknown address produced "We can't find a user with that email address." Anyone could use this to build a list of valid administrator, reseller and customer addresses for phishing or targeted guessing. The endpoint was throttled to five requests a minute per email and address together, which does not limit one source probing many different addresses.
FIX	An unknown address now returns the same message and status as a known one. A second limit of 20 requests a minute per source address was added.
RETEST	Several addresses, including an invented one, all returned the same message.
STATUS	Fixed, retested on v0.20.12.

SEC-2: Missing browser security headers Low

ISSUE	The panel already sent <code>Strict-Transport-Security</code> , <code>X-Content-Type-Options</code> and <code>X-Frame-Options</code> . It did not send <code>Content-Security-Policy</code> , <code>Referrer-Policy</code> or <code>Permissions-Policy</code> .
FIX	All panel pages now send <code>Referrer-Policy: strict-origin-when-cross-origin</code> , a restrictive <code>Permissions-Policy</code> , and <code>Content-Security-Policy: base-uri 'self'; object-src 'none'</code> .
LIMIT	The policy is a baseline. It does not restrict where scripts load from, because the panel uses inline scripts and loads scripts from the panel origin when reached through a customer's domain. A full script-restricting policy needs the inline scripts reworked first and is future work.
STATUS	Fixed (baseline), retested on ports 2087 and 2083.

SEC-3: phpMyAdmin entry point identifiable on port 443 Low

ISSUE	<code>/phpmyadmin/</code> on port 443 redirects through a sign-on bridge to the account panel login. Access requires a valid account login, but the redirect confirms phpMyAdmin is installed, and it would become reachable if account authentication were ever bypassed.
RECOMMENDATION	Answer a plain not-found for every address under <code>/phpmyadmin/</code> unless the browser holds a signed-on phpMyAdmin session.
STATUS	Fixed in v0.20.18, verified from outside on a development server and on a production server running v0.20.19. Every address under <code>/phpmyadmin/</code> (the folder, pages, scripts and themes) now gives the same plain not-found as an address that does not exist, the sign-on bridge gives nothing away without a valid one-time token, and the sign-on cookie is secure and HTTP-only. Customers still open phpMyAdmin with one click from their account panel.

SEC-7: Session cookie uses SameSite=Lax Low

ISSUE	The session cookie is <code>SameSite=Lax</code> . Strict is marginally tighter against cross-site requests.
DECISION	Kept as Lax. Under Strict, the single sign-on hand-off from a customer's own domain into the panel would arrive without the session cookie. All state-changing requests are already protected by a CSRF token, and every attempt to bypass it during testing returned HTTP 419.
STATUS	Accepted risk.

SEC-1, SEC-9: nginx version disclosure Info

ISSUE	Every response carried <code>Server: nginx/1.24.0 (Ubuntu)</code> , and the default error page on port 443 printed the same string in its body. This helps an attacker choose exploits for that exact version.
FIX	<code>server_tokens off</code> on the panel's own virtual hosts. The upgrade rewrites an existing panel vhost to add it.
RETEST	The header now reads <code>Server: nginx</code> on ports 443, 2087 and 2083, and no error page body contains a version.
STATUS	Fixed, retested on v0.20.12.

SEC-10: API route enumeration Info

ISSUE	An unauthenticated request to a real API route returned 401, while a made-up route returned 404. An outsider could map the API without credentials.
FIX	Every unauthenticated request under <code>/api</code> now returns 401. Authenticated callers still get 404 for unknown routes.
RETEST	Real routes, invented routes and POST requests all returned 401.
STATUS	Fixed, retested on v0.20.12.

5. Retest

The test VM was upgraded to v0.20.12 and the affected checks were repeated from outside. Two upgrade defects surfaced and were fixed during the retest: an upgrade's first run is executed by the previous release's script, so it could not rewrite the nginx vhost or restart the ban rule; and the stale ban rule used a full port range that the first fix did not recognise. Both are corrected in v0.20.12. For any server, run the upgrade twice when moving from an earlier release.

CHECK	RESULT ON V0.20.12
Server header and error-page version (443, 2087, 2083)	No version shown
CSP, Referrer-Policy, Permissions-Policy (2087, 2083)	Present
Forgot-password, known versus unknown address	Identical response
API, real versus invented routes, unauthenticated	401 for all
Login POST without CSRF token	419
Ban after repeated failed logins	2087 and 2083 refused; 443, 80 and SSH reachable

6. Controls that held

- **Login rate limiting.** Five attempts a minute per address, then HTTP 429, backed by a network-level ban.
- **No user enumeration at login.** The same message and timing for an unknown user and a wrong password, on both panels.
- **CSRF.** A missing, empty or wrong token returned 419 on every state-changing request tried.
- **Sessions.** A new session identifier is issued at login, and the old session is rejected after logout.
- **Direct object references.** Requests for other or non-existent account identifiers returned 404.
- **Path traversal.** The log download parameter rejected traversal payloads and a null-byte variant read nothing outside the allowed files.
- **Mass assignment.** Extra fields such as `role` and `is_admin` on an account update were ignored.
- **Verb tampering.** Method override via a form field or header is honoured, as designed, but still requires a valid CSRF token.
- **API.** A logged-in web session does not grant API access; the API needs a separate token.
- **Exposure.** The setup page is locked once setup completes, debugging tools are absent, `.env` and other sensitive files return 403, and invalid internal-route tokens return 404.

7. Conclusion

The panel's authentication, session handling, request forgery protection and access checks held up against the attacks tried. The issues found were about disclosure, scoping and defaults, and every item except one accepted low-severity trade-off (SEC-7) was fixed, seven of them verified by retest. A stronger Content-Security-Policy remains as future work. A repeat test is recommended after any significant change to authentication, the API or the installer.

KNJ Panel penetration test summary, 14 October 2026. Tested on v0.20.9, retested on v0.20.12; SEC-3 verified on v0.20.19.