Guvenkaya

SAMPLE ENGAGEMENT
15 SEPTEMBER 2026
REF GUV-VLT-2026-01

Review model
White-box penetration test with targeted source code review

SECURITY ASSESSMENT REPORT

Veltora - app.veltora.io

Anyone who knew the settlement webhook URL could credit any account with any amount and withdraw it as real funds.

Prepared for
Veltora Labs Ltd.

Client
Veltora Labs Ltd.

Review period
31 August 2026 to 11 September 2026

Reviewed baseline
a41c07e9d2b85f3e

Review team
Piotr Cielas - Principal
Timur Güvenkaya - Specialist
Łukasz Mikuła - Specialist

7	13	71	14	1
Flows reviewed	Invariants evaluated	Test cases	Findings raised	Fixed in review

Engagement Dashboard

ASSESSMENT CONCLUSION

Anyone who knew the settlement webhook URL could credit any account with any amount and withdraw it as real funds. The route checked no signature. We reported it on day two, Veltora shipped signature verification and a replay guard two days later, and the fix is in the baseline this draft covers. Two High findings remain open. A payment link ID that turns up in a log, a support ticket or a screenshot recovers the escrow key and claims the funds. Any member of an organisation can change the recipients of a payout batch after an approver has signed it off. Four Medium findings need a precondition first: unauthenticated passkey writes that stall the credential backup, a session that survives sign-out on a shared browser, an owner signing path that forwards whatever typed data it is given, and a confirmation screen that shows only eight characters of the recipient. Before the next release: re-issue the payment link format, freeze approved payout batches on the server, clear the refresh cookie on sign-out, and allowlist what the owner path will sign. Relaymint settles and Wardkey holds the embedded keys. We did not review either service, only how this application handles what they return.

The 7 flows are how coverage is organised end to end; file families group related paths, not individual files. The 14 findings are pieces of work to track, not 14 separate vulnerabilities, because several share one cause. A threat scenario is an attack we modelled; an invariant is a property the system should hold. "No issue identified" means evidence rejected that scenario against commit a41c07e9d2b85f3e.

ENGAGEMENT RECORD

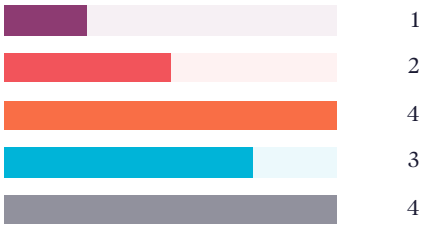
CLIENT	Veltora Labs Ltd.
SYSTEM	Veltora - app.veltora.io: the Next.js web wallet for passkey sign-in, embedded wallets, send, swap, payment links and recurring organisation payouts over the Relaymint settlement API and Wardkey embedded wallets
REVIEWED BASELINE	a41c07e9d2b85f3e6c19d04b7a2e58f1c93d6b20
REVIEWED SCOPE	8 groups of files, read line by line or as supporting context. The Scope section lists them
PRINCIPAL ADVISOR	Piotr Cielas
SPECIALISTS	Timur Güvenkaya, Łukasz Mikuła

TIMELINE

01 Review start
31 August 2026

02 Draft report
15 September 2026

FINDINGS BY SEVERITY



ASSESSMENT SNAPSHOT

7
FLOWS ASSESSED

1
FIXED AND VERIFIED

13
OPEN ENTRIES

THREAT SCENARIOS

19
ASSESSED

12
ISSUE IDENTIFIED

7
NO ISSUE IDENTIFIED

SECURITY INVARIANTS

13
EVALUATED

8
DO NOT HOLD

1
PARTLY HOLD

4
HOLD

Review Team

The principal and specialists named on this engagement. Each profile is the public Guvenkaya biography for that reviewer.

Piotr Cielas

Principal advisor

CURRENTLY
Head of Security, Agora · $45B+ in volume

OSCP · OSWE · U.S. patents

Piotr is Head of Security at Agora, where he oversees information security, cybersecurity, data protection, and corporate IT risk management. He brings both industry and consulting experience, having led information security advisory engagements and security program development for global financial institutions and large organizations. Earlier in his career, Piotr was a Senior Cybersecurity Consultant at Ernst & Young (EY), leading security assessments across financial services, healthcare, and government. He holds CEH, OSCP, and OSWE certifications, has contributed to the CVE program, and is the inventor of multiple U.S. patents related to information security and blockchain technology.

Timur Güvenkaya

Specialist advisor

Rust · NEAR & Substrate · Custody

Timur founded Guvenkaya after seeing teams reduce security to code review while their real risk spans architecture, infrastructure, operations, custody, and launch decisions. Before Guvenkaya, he established and led a security engineering practice for complex blockchain systems, specializing in Rust-based and non-EVM ecosystems including Substrate and NEAR. Earlier at Invicti, he helped build enterprise vulnerability-scanning and security detection engines used by Fortune 50 companies and public-sector organizations.

Łukasz Mikuła

Specialist advisor

OSCP · OSCE · eWPT · eWPTX · 100+ audits

Łukasz is a security researcher with 10+ years in offensive security and a public portfolio of 100+ audits across 8+ ecosystems. His smart-contract work spans EVM/Solidity, Move, Rust-based ecosystems, CosmWasm, Solana, Substrate, and TON, including assessments for Coinbase, MegaETH, Kyber, Jupiter, Zilliqa, IOTA, and Initia Move. At ING, he worked across web application and infrastructure penetration testing, red-team work, exploit development, reverse engineering, mobile security, adversary simulation, and smart-device testing. At Binance, he worked on security concerns for high-scale digital asset systems. He holds the OSCP, OSCE, eWPT, and eWPTX certifications and has CVE disclosures affecting IBM, Oracle, F5, Dell, and Red Hat.

Contents

Every contents row and every underlined record ID is clickable.

Advisory and Client

Guvenkaya

THE ADVISORY

Guvenkaya

Guvenkaya specializes in security for digital assets and financial infrastructure, from smart contracts and protocols to applications, infrastructure, custody and cryptography.

Our team brings experience from Kraken, OpenZeppelin, Binance, ZKsync, Nillion, Invicti and EY, alongside the sitting Heads of Security at Gauntlet and Agora. Across the team: 200+ security reviews, $30B+ in volume secured, 13 published cryptography research papers, and credentials including OSCP, OSWE and CISM.

- Every reviewer has a public bio and is named before you sign. We match expertise to your system, with a principal leading scoping and threat modeling, and checking findings.
- Alongside findings, you receive a threat model, invariant register, new tests in your repo and a record of the attack scenarios we tested.

VELTORA

THE CLIENT

Veltora Labs Ltd.

Veltora is a web wallet for individuals and small businesses. Users sign in with a passkey or an external wallet, hold funds in an embedded wallet custodied by Wardkey, send and swap across chains through the Relaymint settlement API, and share claimable payment links. Organisations run recurring payouts to contractors from a shared balance, with an approver role that signs off each batch.

The application under review is the Next.js frontend and its API routes. It constructs, displays and signs every payment those flows execute. Settlement belongs to Relaymint and key custody to Wardkey, and both sit outside this repository as trust boundaries.

- The system we reviewed builds, signs and submits. It does not settle.
- Keys live with Wardkey; quotes and settlement live with Relaymint.
- The server-side controls live in three places: the API routes, one service-role database client, and a scheduled payout job.

System Components and Trust Boundaries

The diagram separates the code we read from external services, data stores, privileged jobs and inherited libraries. Each component carries the number of findings that landed on it, by severity, so you can see which parts carry the most, and how severe, before reading a single finding.

WHAT A BOUNDARY MEANS

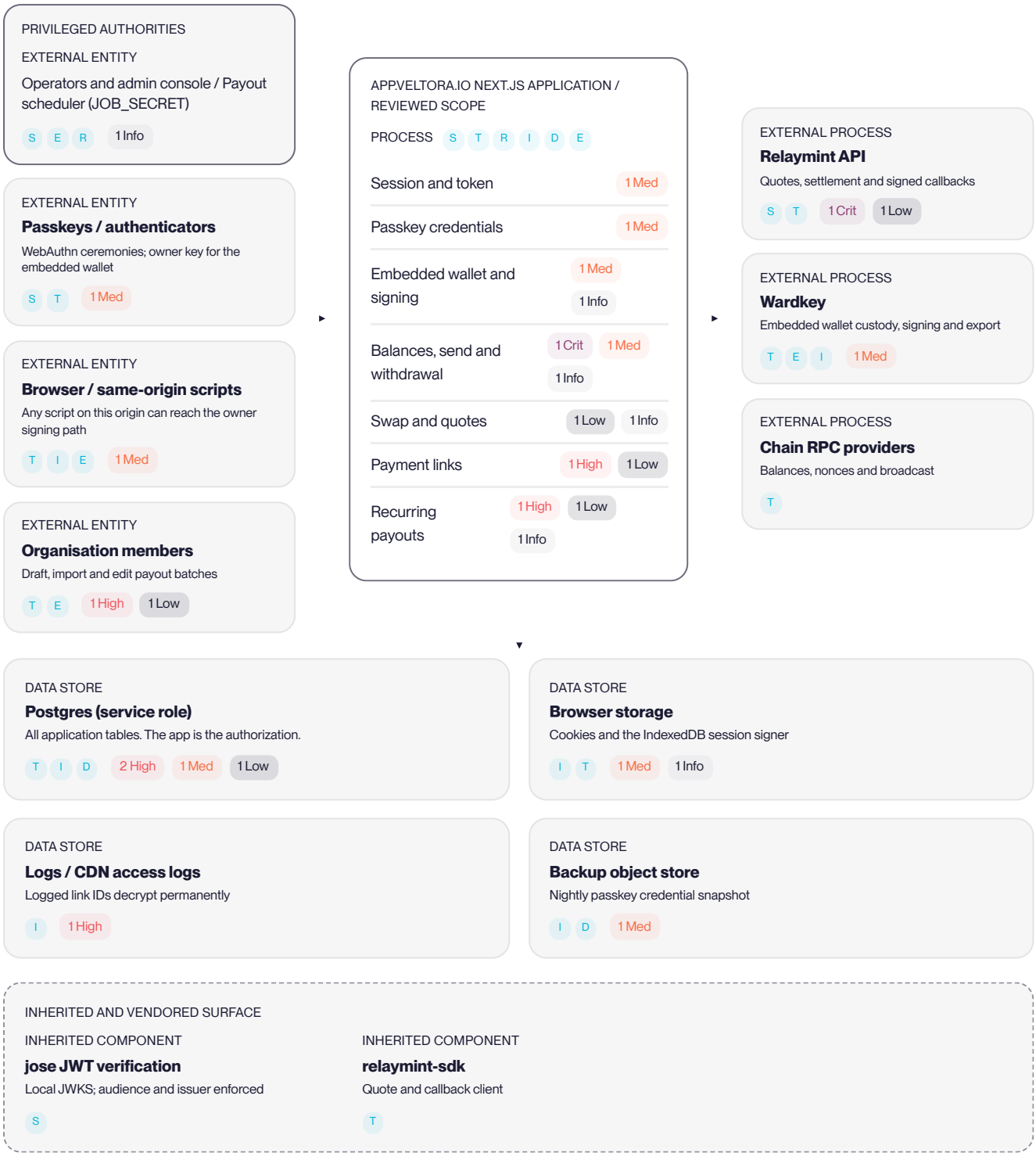
Reviewed scope	Veltora-controlled Next.js routes, state machines, the payout job and client construction of quotes, payments and signatures.
Trust boundary	We took the documented behaviour on trust and reviewed how Veltora copes when it turns out to be wrong. Relaymint, Wardkey and the chain RPC providers.
Inherited	We reviewed the parts the application actually calls. If the library changes, that is a new review.
Capability channel	A secret that belongs in a URL fragment, a cookie or a local signer and nowhere else. If it reaches a log, an API response or another user, confidentiality is gone.

The database client always holds the service-role key, so the application code is the only authorization there is. Everything else in this report serves one requirement: the asset, amount, recipient, fee and signer shown on screen have to be the ones the application authorizes and submits.

Input ownership

This table records who gets to set each value, which is what decides whether a bad value is a finding. An operator account someone has stolen is not in this table; that is outside the attacker model.

OWNER	WHAT THEY CONTROL
Attacker	Paths and IDs; query strings; bodies; headers on non-browser clients; wallet addresses, recipients, amounts, tokens and slippage; payment link blobs; passkey credential IDs and public keys; webhook bodies sent to a public URL; every response from Relaymint, Wardkey, RPC providers and token metadata
Organisation member	Payout batch drafts, CSV imports and recipient lists within their own organisation
Operator	Environment variables and secrets; webhook signing keys; fee settings; feature flags; database schema; job schedules
Developer	Source, lockfiles, framework configuration, static content and scripts



Letters are the STRIDE classes prioritized for that element in this engagement: S spoofing, T tampering, R repudiation, I information disclosure, D denial of service, E elevation of privilege

Security-Critical Flows

We worked through seven flows: identity and signing authority first, then money movement in the order the developers ranked it, then payouts.

01 Authentication and session

INVARIANTS INV-01 INV-02

FINDINGS 1 Med GUV-5

PATH THROUGH THE SYSTEM

Wallet or passkey ▸ Challenge ▸ Session token ▸ Cookies ▸ Route

SECURITY PROPERTIES

Who you are comes from a verified session token, checked against the wallet the client claims. A path or body parameter cannot pick a different account. Signing out ends the session.

02 Passkey registration and recovery

INVARIANTS INV-03 INV-12

FINDINGS 1 Med GUV-4

PATH THROUGH THE SYSTEM

Authenticator ▸ POST /api/passkeys/credentials ▸ Postgres ▸ Backup

SECURITY PROPERTIES

A public key is stored only after a real registration ceremony. Unauthenticated callers cannot grow the credential table until the backup fails.

03 Embedded wallet and signing authority

INVARIANTS INV-04 INV-05 INV-02 INV-13

FINDINGS 1 Med 1 Info GUV-7 GUV-13

PATH THROUGH THE SYSTEM

Session ▸ Wallet lookup ▸ Passkey proof ▸ Wardkey ▸ Session signer

SECURITY PROPERTIES

A caller reaches only their own embedded wallet. The owner path signs only what the user has seen decoded. A session signer does not outlive the wallet it was issued for.

04 Balances, send and withdrawal

INVARIANTS INV-06 INV-07 INV-11

FINDINGS 1 Crit 1 Med 1 Info GUV-2 GUV-8 GUV-11

PATH THROUGH THE SYSTEM

Relaymint callback ▸ Ledger ▸ Review ▸ sendMachine ▸ Refund

SECURITY PROPERTIES

A ledger credit comes only from a signed, fresh settlement event, once. What the user reviews is what is sent, and the review shows enough of the recipient to tell a lookalike apart. A failed withdrawal refunds once, to the source.

05 Swap and quote execution

INVARIANTS INV-06 INV-08

FINDINGS 1 Low 1 Info GUV-6 GUV-14

PATH THROUGH THE SYSTEM

Form ▸ Dry quote ▸ Review ▸ swapSubmitMachine ▸ Live quote

SECURITY PROPERTIES

The reviewed quote is a dry run and submit fetches a live one, so the two have to be compared before anything is signed.

06 Payment links

INVARIANTS INV-09

FINDINGS 1 High 1 Low GUV-1 GUV-9

PATH THROUGH THE SYSTEM

Escrow keypair ▸ Encrypt ▸ Store ▸ Fragment link ▸ Claim

SECURITY PROPERTIES

The link ID on its own cannot decrypt the link or recover the escrow key. The complete link is a bearer instrument by design.

07 Recurring payouts

INVARIANTS INV-10 INV-11

FINDINGS 1 High 1 Low 1 Info GUV-3 GUV-10 GUV-12

PATH THROUGH THE SYSTEM

Draft batch ▸ Import ▸ Approve ▸ Scheduler ▸ Relaymint

SECURITY PROPERTIES

Only an approver can release a batch, and a batch cannot change after approval. The job that submits it takes a lease and never submits the same batch twice.

Scope and Review Boundaries

REVIEW DEPTHS

Deep	Line-by-line review with explicit attack hypotheses, and a reproduction attempt against the staging deployment for each hypothesis that survived reading.
Supporting	The code was read for its effect on a reviewed flow. Behaviour outside that effect was not assessed.
Targeted	Entry points were exercised or triaged against the flow's security properties rather than read in full.
Trust boundary	We took the documented behaviour on trust and reviewed how the application copes when it turns out to be wrong. We did not review the component itself.

This is the code we read, and how closely.

PATH	REVIEW DEPTH	PRIMARY FLOW
src/features/auth/, src/lib/auth/verifySessionToken.ts	Deep	Authentication and session
src/middleware.ts, src/app/api/activity/	Supporting	Authentication and session
src/app/api/passkeys/, src/app/api/jobs/backup-passkeys/route.ts	Deep	Passkey registration and recovery
src/app/api/wallets/, src/features/signing/, localSessionSigner.ts	Deep	Embedded wallet and signing
src/app/api/webhooks/settlement/route.ts, src/features/send/, src/features/withdraw/machines/	Deep	Balances, send and withdrawal
src/features/swap/, src/lib/relaymint/	Deep	Swap and quote execution
src/app/api/links/, link encoder and claim flow	Deep	Payment links
src/app/api/payouts/, src/jobs/payoutScheduler.ts	Deep	Recurring payouts
src/app/api/payouts/import/, CSV parser	Targeted	Recurring payouts
Relaymint, Wardkey, chain RPC providers and hosting platform internals	Trust boundary	all
Presentational components, marketing pages, analytics dashboards, unrelated migrations	Excluded	

Out of scope

We did not run destructive tests against production, load-test for denial of service, attempt social engineering, or touch accounts and funds that were not ours.

Assessment Framework

This section defines the words the Findings Summary uses. Every finding in this report records an impact and a likelihood, and its severity is what the matrix below returns for that pair. Nothing else sets it.

Every finding goes through the same chain, from the attack scenario we modelled to what we are asking you to do about it.

RISK CHAIN APPLIED TO EVERY FINDING

01	02	03	04	05	06
Threat scenario	The weakness	Likelihood	Impact	Severity	What to do
TM-xx, from the threat scenario register	What we actually found in the code	Likely, possible or rare	Severe, high, moderate or negligible	Derived from the matrix below	Fix it, accept it with a compensating control, or watch it

Impact

What happens if someone succeeds, judged against this system and this client.

IMPACT	WHAT IT MEANS HERE
Severe	User or organisation funds are lost or taken at scale, or the signing authority behind them is compromised, with no precondition the attacker does not already control.
High	Funds or an account are lost, taken over or exposed, but the attacker needs a precondition first: a leaked identifier, organisation membership, physical access to a signed-in browser, or an upstream service behaving dishonestly.
Moderate	A security control is lost, bypassed or degraded with no realised loss behind it, so harm needs a second and independent failure. Also private data reaching someone who should not see it, and availability lost on a path that has a manual recovery.
Negligible	The behaviour is wrong, or a defence is missing, and we could demonstrate no security consequence.

Likelihood

How easy the problem is to find combined with how hard it is to exploit. It is not a forecast of how often it will happen.

LIKELIHOOD	WHAT IT MEANS HERE
Likely	Reachable by an unauthenticated caller with no special knowledge, with the effect following from a normal request.
Possible	Reachable, but the attacker has to do some work: hold an identifier, win a race, accumulate state, or already be authenticated.
Rare	Needs a precondition the attacker does not ordinarily get: an upstream API or platform behaving dishonestly, or physical access to a signed-in device.

Severity

Severity comes out of the matrix mechanically, from the impact and likelihood we recorded. Correctness and hardening findings carry Informational, and we do not count them as vulnerabilities.

LIKELIHOOD / IMPACT	SEVERE	HIGH	MODERATE	NEGLIGIBLE
LIKELY	Critical	High	High	Medium
POSSIBLE	High	High	Medium	Low
RARE	Medium	Medium	Low	Informational

RATING RULE

Two rules constrain what goes into the matrix. First, a missing control or a disproved assumption is rated on what we actually demonstrated, not on the worst thing the missing control might have prevented. Second, any finding that needs an upstream API, platform or dependency to be compromised first is rated Rare. There is no override that quietly bumps a severity. The matrix is Guvenkaya's own rating model and is not a NIST risk score.

Remediation complexity

Every finding carries an effort estimate as well. **Simple** is a local change with no consequences for stored state. **Moderate** means the fix touches the data model, a migration, caching behaviour or a shared component. **Involved** means the design has to change.

Annex D records the external frameworks this method aligns with.

Findings Summary

The one Critical finding is fixed. Both High findings are open. There are 10 security findings in total: one Critical, two High, four Medium and three Low. A further four findings are correctness or hardening work, rated Informational; they carry GUV numbers so engineers can track them, and they are not counted as vulnerabilities.

Counting findings is not the same as counting causes. GUV-1 and GUV-9 are one link-format problem written up twice because they need separate fixes, and GUV-3 and GUV-10 are two gaps in the same approval control.

1	2	4	3	4
CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
Unsigned settlement webhook credited balances. Fixed during the review	A link ID that claims the link, and payout batches editable after approval	Passkey table flooding, a session that survives sign-out, an unconstrained owner path and a truncated recipient	2 demonstrated defects and 1 that depends on an unverified external guarantee	3 correctness bugs and 1 piece of hardening

FINDING TYPES

Defect	We showed an attacker can cause the effect in the code we reviewed.
Conditional	The effect needs an external service or state to behave in a way we could not check.
Correctness	The code gets state, arithmetic or display wrong, with no security consequence we could demonstrate.
Hardening	A defence worth adding. We found no way to exploit its absence.

Severity is not a judgement call. Every finding records an impact and a likelihood, and the severity in the table below is what the matrix returns for that pair. The Assessment Framework above defines all three scales and the Simple / Moderate / Involved scale in the remediation-complexity column.

Findings register

FINDING	TYPE	IMPACT	LIKELIHOOD	SEVERITY	REMEDIATION COMPLEXITY	STATUS
GUV-1 - Payment link ID alone recovers the escrow key and can claim a funded link	Defect	High	Possible	High	Moderate	Open
GUV-2 - Settlement webhook accepted without signature verification, crediting ledger balances	Defect	Severe	Likely	Critical	Simple	Fixed
GUV-3 - Any organisation member can change payout recipients after approval	Defect	High	Possible	High	Moderate	Open
GUV-4 - Unauthenticated passkey writes grow the credential table and stall the backup	Defect	Moderate	Possible	Medium	Moderate	Open

TABLE CONTINUED

FINDING	TYPE	IMPACT	LIKELIHOOD	SEVERITY	REMEDIATION COMPLEXITY	STATUS
GUV-5 - Sign-out leaves the refresh cookie behind, so the session resumes without a passkey	Defect	High	Rare	Medium	Simple	Open
GUV-6 - Submit signs a live quote without comparing it to the reviewed quote	Conditional	Moderate	Rare	Low	Moderate	Open
GUV-7 - Owner signing path forwards arbitrary typed data with no server-side allowlist	Defect	Moderate	Possible	Medium	Involved	Open
GUV-8 - Confirmation screen shows only eight characters of the recipient address	Defect	Moderate	Possible	Medium	Simple	Open
GUV-9 - Payment link rows never expire and stay decryptable after claim	Defect	Moderate	Rare	Low	Moderate	Open
GUV-10 - Payout batch total computed in the browser and stored without recomputation	Defect	Negligible	Possible	Low	Simple	Open
GUV-11 - Displayed network fee uses a cached rate; the submitted fee uses a live one	Correctness	Negligible	Rare	Informational	Simple	Open
GUV-12 - Payout scheduler takes no lease, so overlapping runs resubmit a batch	Correctness	Negligible	Rare	Informational	Moderate	Open
GUV-13 - Session signer key survives a wallet switch	Hardening	Negligible	Rare	Informational	Simple	Open
GUV-14 - Price impact computed against output rather than input	Correctness	Negligible	Rare	Informational	Simple	Open

Detailed Findings

Each finding says what we actually demonstrated, then says separately what would have to be true for the worst case, so you can tell the two apart. Then the code, then the change that closes it. Correctness and Hardening findings use the same **Security impact** heading; there it explains why the item matters operationally, and does not claim an exploit.

GUV-1 Payment link ID alone recovers the escrow key and can claim a funded link

SEVERITY	TYPE	FLOW
High	Defect	06 Payment links
THREAT SCENARIO	**INVARIANTS**	**STATUS**
TM-01	INV-09	Open
IMPACT	**LIKELIHOOD**	**NAVIGATION**
High	Possible	**Risk summary**

SECURITY IMPACT

A payment link is meant to be a bearer instrument: whoever holds the complete link, including the fragment after #, can claim the funds, and nobody else can. The fragment carries the IV that the encrypted escrow key needs. The link ID in the path is supposed to be useless on its own, which is why it is safe to log, and it does get logged: by the CDN, the hosting platform, the error monitor and every support ticket that pastes a failing link.

The ID is not useless. GET /api/links/[linkId] returns the AES key with the ciphertext and the GCM tag, and with the key in hand the IV is not secret. XOR the tag with the GHASH of the ciphertext to get the encrypted counter block, decrypt that block with the key, and the first 96 bits are the IV. The key and the IV then decrypt the escrow key. Our proof of concept recovered the escrow private key from a link ID alone and signed a claim with it, using the application's own encoder, without claiming a live link.

So anyone who can read a log line that contains a link ID can claim that link while it is funded. The attacker needs the ID, which is why likelihood is Possible and not Likely.

SRC/APP/API/LINKS/[LINKID]/ROUTE.TS · TS

```ts
export async function GET(_req: Request, { params }: { params: { linkId: string } }) {
  const { data } = await db
    .from("payment_links")
    .select("ciphertext, aes_key, expires_at")
    .eq("id", params.linkId)
    .single()
  return Response.json({ ciphertext: data.ciphertext, key: data.aes_key })
}
```

RECOMMENDATION

Stop returning the key. Split the secret so that the server holds nothing that decrypts the link: derive the encryption key from material that lives only in the fragment, and make the link ID a random value with no relationship to it. Re-issue every unclaimed link under the new format and invalidate the old ones.

REMEDIATION AND VERIFICATION

Open

GUV-2

Settlement webhook accepted without signature verification, crediting ledger balances

SEVERITY	TYPE	FLOW
Critical	Defect	04 Balances, send and withdrawal
THREAT SCENARIO	**INVARIANTS**	**STATUS**
TM-02	INV-11	Fixed
IMPACT	**LIKELIHOOD**	**FIX COMMIT**
Severe	Likely	9c3e1a7f04b2d86e5a1f7c20e4d93b58a6f2c071

VERIFICATION

Code review of the fix commit plus two contributed regression tests

NAVIGATION

Risk summary

SECURITY IMPACT

POST /api/webhooks/settlement is where Relaymint tells Veltora that a deposit or swap has settled. The route parsed the body, looked up the user by the account field and credited the ledger with amount. It did not check the Relaymint-Signature header, did not check a timestamp, and did not record the event ID. The URL is in the public client bundle because the staging build also uses it for polling.

Anyone could therefore credit any account with any amount, and the ledger balance is what the withdrawal flow spends from. We credited a test account on staging with a forged event and withdrew the credit to an external address. This needed no account, no identifier and no race, which is what Likely means, and the impact is funds taken at scale.

We reported this on the second day of the review. Veltora shipped signature verification and an event-ID uniqueness constraint two days later. The fix commit is in the baseline this draft covers, and we verified it there.

SRC/APP/API/WEBHOOKS/SETTLEMENT/ROUTE.TS · TS

```ts
export async function POST(request: Request) {
  const event = settlementSchema.parse(await request.json())
  const account = await findAccount(event.account)
  await ledger.credit(account.id, event.amount, event.asset)
  return new Response(null, { status: 204 })
}
```

RECOMMENDATION

Verify the Relaymint signature over the raw body before parsing, reject events older than five minutes, and record each event ID under a unique constraint so a replay credits nothing. Treat the credit as the last step of a transaction that includes the event-ID insert.

REMEDIATION AND VERIFICATION

Fixed in commit 9c3e1a7f04b2d86e5a1f7c20e4d93b58a6f2c071, which is included in the reviewed baseline. The route now verifies the signature over the raw body with a constant-time comparison, rejects stale timestamps, and inserts the event ID under a unique constraint in the same transaction as the credit. Two contributed regression tests cover a forged signature and a replayed event; both pass.

GUV-3 Any organisation member can change payout recipients after approval

SEVERITY	TYPE	FLOW
High	Defect	07 Recurring payouts
THREAT SCENARIO	**INVARIANTS**	**STATUS**
TM-03	INV-10	Open
IMPACT	**LIKELIHOOD**	**NAVIGATION**
High	Possible	**Risk summary**

SECURITY IMPACT

A payout batch has a draft state, an approved state and a submitted state. The approver role exists so that one person cannot both prepare and release a payment. The UI hides the edit button once a batch is approved and shows it only to editors, but PATCH /api/payouts/[batchId] checks only that the caller is a member of the batch's organisation. It does not check the caller's role, and it does not check the batch status.

So a viewer, or an editor after approval, can replace the recipient address on any line of an approved batch. The scheduler submits whatever the row holds at run time. We changed a recipient on an approved staging batch as a viewer and watched the scheduler pay the new address.

The attacker has to be a member of the organisation, which is why likelihood is Possible. The approval step exists because membership alone should not be enough to move money.

SRC/APP/API/PAYOUTS/[BATCHID]/ROUTE.TS · TS

```ts
export async function PATCH(request: Request, { params }: { params: { batchId: string } }) {
  const session = await requireSession()
  await requireOrgMember(session, params.batchId)
  const patch = batchPatchSchema.parse(await request.json())
  await db.from("payout_lines").upsert(patch.lines)
  return Response.json({ ok: true })
}
```

RECOMMENDATION

Enforce the role and the status on the server: only editors can change a draft, nobody can change an approved batch, and approval signs a hash of the lines that the scheduler re-checks before submitting. Record who changed what on each line.

REMEDIATION AND VERIFICATION

Open

GUV-4 Unauthenticated passkey writes grow the credential table and stall the backup

SEVERITY	TYPE	FLOW
Medium	Defect	02 Passkey registration and recovery
THREAT SCENARIO	INVARIANTS	STATUS
TM-04	INV-03, INV-12	Open
IMPACT	LIKELIHOOD	NAVIGATION
Moderate	Possible	**Risk summary**

SECURITY IMPACT

POST /api/passkeys/credentials stores a credential ID and public key before any registration ceremony completes, and it asks for no session. The public key field has no length limit in the schema or the database. The nightly backup reads the whole table into memory and serializes it as one JSON value.

Anyone can therefore write rows until the backup job runs out of memory or time. Nobody is locked out when that happens. What is lost is the ability to restore passkey credentials if the live table is later lost, and account access depends on them. Rows never expire, so the condition persists until an operator clears it by hand.

We rate the impact Moderate because the harm is the loss of a recovery control rather than a realised loss.

RECOMMENDATION

Persist a credential only after a verified registration ceremony. Cap the public key length in the schema and the database. Stream the backup in bounded chunks.

REMEDIATION AND VERIFICATION

Open

GUV-5 Sign-out leaves the refresh cookie behind, so the session resumes without a passkey

SEVERITY	TYPE	FLOW
Medium	Defect	01 Authentication and session
THREAT SCENARIO	INVARIANTS	STATUS
TM-05	INV-02	Open
IMPACT	LIKELIHOOD	NAVIGATION
High	Rare	**Risk summary**

SECURITY IMPACT

Sign-out clears the access cookie and leaves the per-wallet refresh cookie, which lives for fourteen days. On the next connect with the same wallet, the client finds no access token, exchanges the refresh cookie for a new one, and lands in a signed-in session. Nothing on that path asks for a passkey assertion.

So on a shared laptop with the wallet extension still unlocked, whoever connects that wallet next resumes the session without the passkey that is supposed to guard the embedded wallet and the payout routes. The cookie is httpOnly and SameSite=Lax, so it cannot be driven from another origin. That is why likelihood is Rare: it needs the device.

RECOMMENDATION

Delete the refresh cookie on explicit sign-out, and require a fresh assertion before a refresh restores a session that was signed out.

REMEDIATION AND VERIFICATION

Open

GUV-6 Submit signs a live quote without comparing it to the reviewed quote

SEVERITY	TYPE	FLOW
Low	Conditional	05 Swap and quote execution
THREAT SCENARIO	INVARIANTS	STATUS
TM-06	INV-06, INV-08	Open
IMPACT	LIKELIHOOD	NAVIGATION
Moderate	Rare	**Risk summary**

SECURITY IMPACT

The quote on the review screen is a dry run. When the user confirms, swapSubmitMachine requests a live quote and signs the intent Relaymint returns. Nothing compares the live quote's recipient, output asset or minimum received with the reviewed one. If Relaymint returns different terms, through a bug or a compromise, the user signs terms they never saw. We rate it Rare because it needs Relaymint to misbehave.

RECOMMENDATION

Compare the live quote field by field against the reviewed quote and fail closed on any difference in recipient, asset, amount or deadline.

REMEDIATION AND VERIFICATION

Open

GUV-7 Owner signing path forwards arbitrary typed data with no server-side allowlist

SEVERITY	TYPE	FLOW
Medium	Defect	03 Embedded wallet and signing authority
THREAT SCENARIO	INVARIANTS	STATUS
TM-07	INV-05	Open
IMPACT	LIKELIHOOD	NAVIGATION
Moderate	Possible	**Risk summary**

SECURITY IMPACT

POST /api/wallets/[walletId]/sign takes a typed-data payload and a passkey assertion and forwards both to Wardkey. It checks that the wallet belongs to the caller. It does not check what is being signed. The passkey prompt shows the same generic text for every request, so a user cannot tell a routine approval from a token allowance to an attacker.

Any script running on the page while the user is signed in can therefore request a signature over anything, and the user approves it believing it is routine. We rate likelihood Possible because it needs a script foothold on the origin, and impact Moderate because we demonstrated the missing control, not a drained wallet.

RECOMMENDATION

Allowlist the typed-data types the product actually uses, decode them on the server, and show the decoded action in the confirmation before the passkey prompt.

REMEDIATION AND VERIFICATION

Open

GUV-8 Confirmation screen shows only eight characters of the recipient address

SEVERITY	TYPE	FLOW
Medium	Defect	04 Balances, send and withdrawal
THREAT SCENARIO	INVARIANTS	STATUS
TM-08	INV-06	Open
IMPACT	LIKELIHOOD	NAVIGATION
Moderate	Possible	**Risk summary**

SECURITY IMPACT

The send confirmation renders the recipient as the first four and last four characters. Eight hex characters are 32 bits, so a matching address takes about four billion attempts, which is minutes on one GPU. Address poisoning then puts that address in the user's transaction history. The confirmation screen, which is the last control, cannot tell the two apart.

RECOMMENDATION

Show the full address on the confirmation, grouped for reading, and flag a recipient that is new to this user.

REMEDIATION AND VERIFICATION

Open

GUV-9 Payment link rows never expire and stay decryptable after claim

SEVERITY	TYPE	FLOW
Low	Defect	06 Payment links
THREAT SCENARIO	INVARIANTS	STATUS
TM-01	INV-09	Open
IMPACT	LIKELIHOOD	NAVIGATION
Moderate	Rare	**Risk summary**

SECURITY IMPACT

Claimed and expired links keep their ciphertext and key in the table, and the GET route keeps serving them. With GUV-1 that means a link ID in an old log decrypts the link's contents indefinitely, including the sender's note and amount, after the funds have gone.

RECOMMENDATION

Delete the key when a link is claimed or expires, and return 410 for those links.

REMEDIATION AND VERIFICATION

Open

GUV-10 Payout batch total computed in the browser and stored without recomputation

SEVERITY	TYPE	FLOW
Low	Defect	07 Recurring payouts
THREAT SCENARIO	INVARIANTS	STATUS
TM-09	INV-10	Open
IMPACT	LIKELIHOOD	NAVIGATION
Negligible	Possible	**Risk summary**

SECURITY IMPACT

The CSV import sums the batch in the browser and posts the total with the lines. The server stores the total as sent. The approver screen leads with that total. An editor can make a batch look smaller than it is; the per-line amounts are still correct and still shown, which is why impact is Negligible.

RECOMMENDATION

Recompute the total on the server from the stored lines and ignore the client value.

REMEDIATION AND VERIFICATION

Open

GUV-11 Displayed network fee uses a cached rate; the submitted fee uses a live one

SEVERITY	TYPE	FLOW
Informational	Correctness	04 Balances, send and withdrawal
THREAT SCENARIO	INVARIANTS	STATUS
TM-12	INV-06	Open
IMPACT	LIKELIHOOD	NAVIGATION
Negligible	Rare	**Risk summary**

SECURITY IMPACT

The review screen shows a fee computed from a rate cached for up to sixty seconds. Submit fetches the live rate. In volatile periods the charged fee can exceed the shown one by a few percent.

RECOMMENDATION

Show the fee the submission will use, or cap the submitted fee at the displayed value.

REMEDIATION AND VERIFICATION

Open

GUV-12 Payout scheduler takes no lease, so overlapping runs resubmit a batch

SEVERITY	TYPE	FLOW
Informational	Correctness	07 Recurring payouts
THREAT SCENARIO	INVARIANTS	STATUS
TM-10	INV-11	Open
IMPACT	LIKELIHOOD	NAVIGATION
Negligible	Rare	**Risk summary**

SECURITY IMPACT

The scheduler picks up approved batches and marks them submitted only after Relaymint accepts them. Two overlapping runs both pick up the same batch. Relaymint rejects the second submission because Veltora sends the batch ID as an idempotency key, so nobody is paid twice. The job still does the work twice and logs a failure every time.

RECOMMENDATION

Take a row-level lease before submitting, and keep the idempotency key.

REMEDIATION AND VERIFICATION

Open

GUV-13 Session signer key survives a wallet switch

SEVERITY	TYPE	FLOW
Informational	Hardening	03 Embedded wallet and signing authority
THREAT SCENARIO	INVARIANTS	STATUS
TM-11	INV-02	Open
IMPACT	LIKELIHOOD	NAVIGATION
Negligible	Rare	**Risk summary**

SECURITY IMPACT

The IndexedDB session signer is keyed by address and is not removed when the user switches wallet. Wardkey rejects its signatures for the new wallet, so we found no exploit. It is key material with no owner.

RECOMMENDATION

Delete the session signer on wallet switch and on sign-out.

REMEDIATION AND VERIFICATION

Open

GUV-14 Price impact computed against output rather than input

SEVERITY	TYPE	FLOW
Informational	Correctness	05 Swap and quote execution
THREAT SCENARIO	INVARIANTS	STATUS
TM-12	INV-06	Open
IMPACT	LIKELIHOOD	NAVIGATION
Negligible	Rare	**Risk summary**

SECURITY IMPACT

The price-impact warning divides by the output value instead of the input value, so it understates impact on thin pools and the warning threshold triggers late.

RECOMMENDATION

Compute impact against the input value at the reference price.

REMEDIATION AND VERIFICATION

Open

Remediation Plan

The findings above are the work. This section is how to sequence it: which findings are really one problem, what order to fix them in, what to do about the payment link IDs already in the wild, and which properties a remediation pass must not break.

Principal risk clusters

Three pairs of findings share one cause each. They stay as separate GUV numbers because engineers need separate fixes and separate tests, but each pair is one piece of work.

CLUSTER	FINDINGS	WHAT IT AMOUNTS TO
The payment link format hands out its own key	GUV-1 GUV-9	One design mistake with two consequences: a link ID claims a funded link, and the link stays readable after it is spent.
Approval is a UI state, not a server rule	GUV-3 GUV-10	The server trusts membership where it should trust role and status, and trusts a client total where it should recompute.
A session that outlives its owner	GUV-5 GUV-13	Sign-out and wallet switch both leave credentials behind in the browser.

Remediation priorities

Fix the code in this order. Priorities 1 to 3 belong in the first response; 4 and 5 should land before the next release.

PRIORITY	WHAT HAS TO BE TRUE	WHY IT COMES FIRST	HOW YOU WOULD KNOW	FINDINGS
1	The payment link format no longer lets the server hand out a decrypting key, and old links are re-issued	A link ID on its own claims a funded link	The ID-only decryption test passes and the GET route returns no key	GUV-1 GUV-9
2	Approved payout batches are immutable on the server, and only editors change drafts	Any member can redirect an approved payout	The role and status negatives pass, and the scheduler re-checks the approval hash	GUV-3 GUV-10
3	Signing out ends the session and removes every credential the browser holds	A shared browser resumes the previous user's session	Sign-out deletes the refresh cookie and the session signer	GUV-5 GUV-13
4	The owner path signs only allowlisted, decoded actions	Any script on the page can get an arbitrary signature approved	Per-type schemas and a confirmation test that renders the decoded action	GUV-7

TABLE CONTINUED

PRIORITY	WHAT HAS TO BE TRUE	WHY IT COMES FIRST	HOW YOU WOULD KNOW	FINDINGS
5	What the user signs is what they reviewed: the full recipient and the reviewed quote	A lookalike address or a changed quote passes the last screen	The recipient and quote-mismatch tests pass at each signing call site	GUV-8 GUV-6

Existing data and incident response

Fixing the code does not retire the payment link IDs already created. Anyone holding one from a log or a support transcript can still use it against the old format.

PRIORITY	ACTION	FINDINGS
1	Find unclaimed link IDs in CDN, platform, application, support and error-monitoring logs, and re-issue those links under the new format.	GUV-1 GUV-9
2	Review payout line edits made after approval since the feature launched, and confirm each with the approver.	GUV-3
3	Reconcile ledger credits against signed Relaymint events for the period before the GUV-2 fix, and record the result.	GUV-2
4	Identify and remove unauthenticated passkey rows, and verify the most recent complete backup.	GUV-4

What held

Three properties hold outright and one holds on the part that matters. Whoever does the remediation should not undo them, and each is pinned by a passing test in the suite we contributed.

PROPERTY	STATE	PINNED BY
Wallet routes cannot read, sign with or export another user's embedded wallet	Holds	EVT-03 EVH-02
A failed withdrawal refunds once, to the source	Holds	EVT-07 EVH-04
Session identity and cached responses stay with the caller	Holds	EVT-10 EVH-01 EVH-06
A settlement event credits once, after the GUV-2 fix	Partially holds	EVT-08 EVH-07

Seven of the nineteen attack scenarios we modelled turned out not to work, and we have the evidence for why in each case. Annex A lists them.

Assurance Opinion

This draft records what we found as at 15 September 2026. The opinion below covers the baseline, scope and evidence named in this report and nothing else. It is not a statement that the application has no other defects. Change the reviewed code paths, the payout approval rules, the signing paths or the payment link format, and the opinion no longer covers the code you changed.

SECURITY OPINION

The one Critical finding was fixed during the review and the fix is in the baseline this report covers. We modelled 19 attack scenarios. Twelve produced a finding; seven closed with no issue found, each against recorded evidence. Of the 13 security properties we set out to test, 4 hold, 1 holds in part and 8 do not hold at this draft. We contributed 71 test cases across 12 files: 43 pass and 28 fail against this code, and the failing ones turn green as the fixes land.

Code Reviewed

a41c07e9d2b85f3e6c19d04b7a2e58f1c93d6b20

Review Period

31 August 2026 to 11 September 2026

Finding Status

1 Fixed; 9 security findings and 4 correctness or hardening findings open

Attack Scenarios

19 modelled: 12 found something, 7 did not

Scope Of Opinion

The seven security-critical flows

Piotr Cielas

Principal Advisor

Guvenkaya Advisory, 15 September 2026

ANNEX A

Threat Model

We went through the system twice looking for threats. The first pass walks the components on the architecture map. The second walks the same system by its edges, asking what can go wrong between two parts rather than inside one. Both passes throw up candidates, and they are deduplicated into the single register at the end of this annex.

THE SIX STRIDE CLASSES

S	**Spoofing.** Pretending to be someone or something else.
T	**Tampering.** Changing data or code in transit or at rest.
R	**Repudiation.** Acting without leaving evidence of who did it.
I	**Information disclosure.** Reading what you should not be able to read.
D	**Denial of service.** Making something unavailable.
E	**Elevation of privilege.** Doing what your role does not permit.

THE TWO PASSES

Element	Found by walking the components on the architecture map.
Interaction	Found by walking the edges between components, below.
Both	Both passes found it, independently of each other.

STRIDE is a prompt for the enumeration, not a complete list of what can go wrong with payments. Approval integrity, quote reminting and leaked capability links come from the invariants instead.

Actors

The architecture map shows the parts. This list shows who can set a scenario off. A compromised operator account and a poisoned dependency are both outside what we modelled.

01 Unauthenticated remote caller

Anyone on the internet, with no account. They reach the settlement webhook and the passkey routes.

SCENARIOS TM-02, TM-04

FINDINGS GUV-2, GUV-4

02 Organisation member

A signed-in viewer or editor in an organisation. The approval control exists so that this person cannot release or redirect a payout alone.

SCENARIOS TM-03, TM-09

FINDINGS GUV-3, GUV-10

03 Link or log holder

Whoever holds a payment link ID from a log, a ticket or a screenshot. The ID alone is meant to be useless, and today it is not.

SCENARIOS TM-01, TM-16

FINDINGS GUV-1, GUV-9

04 Same-origin script

JavaScript running on app.veltora.io while a user is signed in. It can reach the owner signing path and the session signer.

SCENARIOS TM-07, TM-11

FINDINGS GUV-7, GUV-13

05 Untrusted upstream

Relaymint, Wardkey and the RPC providers. We reviewed what this application signs and displays if one of them lies to it.

SCENARIOS TM-06

FINDINGS GUV-6

06 Next user of a shared browser

Someone who connects the same wallet on a device where the previous user signed out.

SCENARIOS TM-05

FINDINGS GUV-5

STRIDE enumeration coverage

S	Spoofing 3 SCENARIOS	TM-02 TM-08 TM-13 Forged callbacks, lookalike recipients and a body parameter that names another account
T	Tampering 10 SCENARIOS	TM-02 TM-03 TM-04 TM-06 TM-08 TM-09 TM-10 TM-12 TM-15 TM-18 The biggest group. What the user or approver reviewed drifting from what gets signed or paid
R	Repudiation NOT MATERIALLY APPLICABLE	Every money-moving action is signed by a passkey or by Relaymint and recorded by them, so no repudiation scenario survived
I	Information disclosure 4 SCENARIOS	TM-01 TM-16 TM-17 TM-19 Link keys reaching a log, and one user's data reaching another
D	Denial of service 2 SCENARIOS	TM-04 TM-10 Storage and jobs that grow or repeat without limit
E	Elevation of privilege 7 SCENARIOS	TM-01 TM-03 TM-05 TM-07 TM-11 TM-14 TM-15 Includes scenarios that closed clean. This strip records what we looked at, not only what we found

Interaction threat view

A Browser — SESSION / ORIGIN / CACHE → Next.js API — S T I D E

- Is the caller who the token says, or who the path or body claims?
- Does anything the browser keeps after sign-out restore the session?
- Can an anonymous caller grow a table the backup has to read?

SCENARIOS TM-04 TM-05 TM-13 TM-17
FINDINGS GUV-4 GUV-5

B Next.js — APP-AS-AUTHORIZATION → Postgres service role — T I E

- Does any route return a secret the row only needs to store?
- Does any route check membership where it should check role and status?

SCENARIOS TM-01 TM-03
FINDINGS GUV-1 GUV-9 GUV-3

C Next.js — EMBEDDED-WALLET AUTHORITY → Wardkey — E

- Can a caller reach another user's wallet by supplying its ID?
- Does the owner path sign anything it is given?

SCENARIOS TM-07 TM-14
FINDINGS GUV-7

D Machines — QUOTE AND SETTLEMENT → Relaymint — T E

- Is the live quote compared with the reviewed one before signing?
- Can a refund pay twice or to another address?

SCENARIOS TM-06 TM-15
FINDINGS GUV-6

E Relaymint — SETTLEMENT CALLBACK → Settlement webhook — S T

- Does the route verify who sent the event?
- Can the same event credit twice?

SCENARIOS TM-02 TM-18
FINDINGS GUV-2

Threat scenario register

ID	THREAT SCENARIO	ORIGIN	STRIDE	FLOW	PROPERTIES	OUTCOME	EVIDENCE
TM-01	A payment link ID from a log recovers the escrow key and claims the link	Both	I E	06	INV-09	Issue	GUV-1 GUV-9 EVT-09 EVH-08
TM-02	A forged settlement callback credits a ledger balance that can be withdrawn	Both	S T	04	INV-11	Issue	GUV-2 EVR-01 EVT-08
TM-03	A member without the approver role changes an approved payout batch	Interaction	E T	07	INV-10	Issue	GUV-3 EVT-11
TM-04	Unauthenticated credential writes grow the table until the backup fails	Both	T D	02	INV-03, INV-12	Issue	GUV-4 EVT-02
TM-05	A signed-out session resumes on a shared browser with no ceremony	Interaction	E	01	INV-02	Issue	GUV-5 EVT-01

TABLE CONTINUED

ID	THREAT SCENARIO	ORIGIN	STRIDE	FLOW	PROPERTIES	OUTCOME	EVIDENCE
TM-06	A reminted or tampered quote is signed as if it were the reviewed one	Interaction	T	05	INV-06, INV-08	Issue	GUV-6, EVT-06
TM-07	The owner path signs a high-risk action the user never saw decoded	Both	E	03	INV-05	Issue	GUV-7, EVT-04
TM-08	A lookalike recipient passes the confirmation screen	Element	S, T	04	INV-06	Issue	GUV-8, EVT-05
TM-09	A client-computed total misstates an approved batch	Element	T	07	INV-10	Issue	GUV-10
TM-10	Overlapping scheduler runs submit one batch twice	Element	T, D	07	INV-11	Issue	GUV-12, EVT-12
TM-11	A session signer key is reused after a wallet switch	Element	E	03	INV-02	Issue	GUV-13
TM-12	Displayed fee or price impact diverges from what settles	Element	T	04, 05	INV-06	Issue	GUV-11, GUV-14
TM-13	A path or body parameter selects another account	Interaction	S	01	INV-01	No issue	EVH-01
TM-14	A caller reaches another user's embedded wallet by supplying its ID	Interaction	E	03	INV-04	No issue	EVT-03, EVH-02
TM-15	A failed withdrawal refunds twice or to another address	Interaction	T, E	04	INV-07	No issue	EVT-07, EVH-04
TM-16	Payment link IDs can be enumerated	Element	I	06	INV-09	No issue	EVH-03
TM-17	A cached response is served to another user	Interaction	I	01	INV-01	No issue	EVT-10, EVH-06
TM-18	A replayed settlement event credits twice after the fix	Interaction	T	04	INV-11	No issue	EVT-08, EVH-07
TM-19	Private key material reaches a log	Element	I	03	INV-13	No issue	EVH-05

ANNEX B

Security Objectives and Invariants

13

INVARIANTS EVALUATED

Across five areas. Each one is a property this codebase can make true or false on its own

4

HOLD

INV-01, INV-04, INV-07 and INV-13. Not one failing case against any of them

1

HOLD IN PART

INV-11. The main property holds; one named part of it does not

8

DO NOT HOLD

The main property itself is broken

A Identity and authority isolation

Sessions, passkeys, embedded wallets and the owner signing path each stay tied to one account and do not drift between them.

INVARIANTS	INV-01 to INV-05
STATE AT BASELINE	2 hold, 3 do not hold

B Intent integrity

The asset, amount, recipient and fee shown to the user are exactly what is signed and submitted.

INVARIANTS	INV-06, INV-08
STATE AT BASELINE	Both do not hold

C Outcome containment and single effect

Refunds, payouts and ledger credits happen once, to the authorized destination, and only after the authorized approval.

INVARIANTS	INV-07, INV-10, INV-11
STATE AT BASELINE	INV-07 holds; INV-11 partially holds; INV-10 does not hold

D Confidentiality and capability confinement

Link keys and signing material reach only the authorized principal.

INVARIANTS	INV-09, INV-13
STATE AT BASELINE	INV-09 does not hold; INV-13 holds

E Abuse and availability

Outside input cannot grow storage or a privileged job without limit.

INVARIANTS	INV-12
STATE AT BASELINE	Does not hold

INVARIANT STATE

Holds	Nothing broken and no failing case against it.
Partially holds	The main property holds. One specific part of it does not, and we name which.
Does not hold	The main property itself is broken.
Cases pass / fail	How the contributed tests for that property split, counted per test group.

Which state a property gets depends on what failed, not on how many tests passed. INV-11 is partial because the fixed webhook now verifies and deduplicates, and the payout scheduler still has no lease.

Every property listed here passes two tests: this codebase can make it true or false on its own, and the product wants it to be true. Full wording, evidence and test links are in Annex F.

ANNEX C

Security Test Contribution

12

TEST FILES

*.security.test.ts, each sitting beside the module it exercises

71

TEST CASES

One locked run of the security suite against the reviewed commit

43

PASSING

Properties that hold, the GUV-2 fix, and current behaviour pinned

28

FAILING

Open findings. They fail on purpose and go green when the fix lands

Contribution GUV-xx means the test was written against that finding. ASSURANCE means we added it with no finding attached, to keep a property that holds from quietly regressing.

Result Whether the case passes against the reviewed commit. A failing case goes green by itself once the production code is fixed.

The full list is in Annex F.

FLOW	SECURITY PROPERTY TESTED	TEST TYPE	TEST FILE	CONTRIBUTION	RESULT
01	Sign-out removes every session credential	Negative	auth/actions/signOut	GUV-5	Fail
01	Responses carry no-store at both auth levels	Regression	api/activity	ASSURANCE	Pass
02	Unauthenticated credential write rejected, key size bounded	Boundary	api/passkeys/credentials	GUV-4	Fail
03	Cross-user wallet read is 404	Negative	api/wallets/[walletId]	ASSURANCE	Pass
03	Owner path rejects unlisted typed data	Negative	api/wallets/[walletId]/sign	GUV-7	Fail
04	Forged and replayed settlement events rejected	Regression	api/webhooks/settlement	GUV-2	Pass
04	Lookalike recipient does not render identically	Invariant	send/components/ConfirmSend	GUV-8	Fail
04	Failed withdrawal refunds once, to the source	Invariant	withdraw/machines/refundMachine	ASSURANCE	Pass
05	Live quote matches the reviewed quote	Invariant	swap/machines/swapSubmitMachine	GUV-6	Fail
06	Link ID alone must not decrypt	Negative	api/links/[linkId]	GUV-1	Fail
07	Viewer cannot edit an approved batch	Negative	api/payouts/[batchId]	GUV-3	Fail

Each *.security.test.ts file sits next to the production module it exercises, and outside the normal bun test gate so it cannot break the main build. bun run test:security runs the suite. Fix the production behaviour and the assertion goes green where it is.

Coverage by flow

SECURITY FLOW	MANUAL REVIEW	RUNTIME TESTING	INVARIANTS	EVIDENCE MAPPINGS	DOMAIN REFERENCE
01 Authentication and session	Deep on auth actions and token verification	Staging session matrix, shared-browser replay	2	2	ASVS V2, V3
02 Passkey registration and recovery	Deep on both credential routes and the backup job	Oversized-write case on staging	2	1	ASVS V2, V3
03 Embedded wallet and signing authority	Deep on wallet routes and the signer	404 negatives against staging	4	2	ASVS V4
04 Balances, send and withdrawal	Deep on the webhook, send and refund machines	Forged callback before the fix, replay after it; refund matrix	3	3	ASVS V4, V5
05 Swap and quote execution	Deep on swap machines and the Relaymint client	Quote substitution with a local proxy	2	1	ASVS V4, V5
06 Payment links	Deep on both routes and the encoder	Self-test PoC against the encoder; no live link claimed	1	1	ASVS V6, V8
07 Recurring payouts	Deep on payout routes and the scheduler	Viewer edit of an approved staging batch	2	2	ASVS V4

ANNEX D

Methodology Alignment

This is the approach we actually used on this engagement, not a generic template.

LAYER	USED FOR
NIST SP 800-30 Rev. 1	Risk-assessment principles: threat condition, predisposing condition, likelihood, impact. No NIST score is printed.
STRIDE	Two enumeration passes: per element (the architecture map) and per interaction (the edge view). Candidates dedupe into the TM register.
OWASP ASVS 5.0	Coverage cross-check after enumeration. Applicable families: V2 Auth, V3 Session, V4 Access Control, V5 Validation, V6 Crypto, V8 Data Protection.
Guvenkaya	Flows, invariants, hypotheses, adversarial testing, evidence.

HOW THE METHOD COMPOSES

01
System model
Components, stores, boundaries and flows

02
STRIDE enumeration
Per element and per interaction

03
Threat scenarios
Deduplicated into the TM register

04
OWASP cross-check
ASVS families checked for gaps

05
Invariants and testing
Adversarial review, tests contributed

06
NIST risk chain
Finding, or property held with evidence

STANDARDS ALIGNMENT

We take our risk-assessment principles from NIST SP 800-30 Rev. 1, use STRIDE as a way of enumerating threats systematically, and cite OWASP ASVS families to check our coverage by domain. Those frameworks sit underneath the flows, invariants and attack hypotheses we built for this system. They do not replace them, and this report is not a compliance statement against any of them.

ANNEX E

External and Inherited Security Surface

Everything the system leans on but does not control, what each one can do to it, and what we decided about that.

COMPONENT	WHAT IT DOES HERE	REVIEW DEPTH	WHAT IT CAN DO TO YOU	THE CONCERN	WHAT WE DECIDED
Relaymint API	Quotes, settlement and signed callbacks	Trust boundary; integration code reviewed Deep	Authors the terms the user signs and reports settlement	No reviewed-versus-live comparison (GUV-6); callback authentication fixed (GUV-2)	Treat as a trust boundary the client must survive; compare locally
Wardkey	Embedded wallet custody, signing and export	Trust boundary; route integration reviewed Deep	Holds the keys and enforces wallet policy	Owner path unconstrained app-side (GUV-7)	Allowlist on the server
Chain RPC providers	Balances, nonces and broadcast	Trust boundary	Report state the UI shows	RPC state is used for display only	Accept for display; settle through Relaymint
jose, relaymint-sdk	Inherited code	Capability actually used reviewed	Token verification and the quote client	Audience and issuer enforced; client used as documented	No change

ANNEX F

Evidence Appendix

Invariant register

The full wording of every security property this review set out to test, and what the evidence says about it. In the Cases column the first number is the contributed test cases that pass against the reviewed commit and the second is those that fail.

OBJECTIVE DOMAIN

A	Identity and authority isolation. Sessions, passkeys, wallets and the owner path stay tied to one account.
B	Intent integrity. What the user reviewed is what gets signed and submitted.
C	Outcome containment and single effect. Refunds, payouts and credits happen once, as authorized.
D	Confidentiality and capability confinement. Link keys and signing material reach only the authorized principal.
E	Abuse and availability. Outside input cannot grow storage or a job without limit.

Each letter links to the domain card in Annex B.

ID	SECURITY INVARIANT	OBJECTIVE DOMAIN	STATE	CASES PASS / FAIL	EVIDENCE
INV-01	Session identity is taken from a verified token compared against the claimed wallet; a path or body parameter cannot select another account	A	Holds	9 / 0	EVT-10, EVH-01, EVH-06
INV-02	Signing out or switching wallet ends the session, and no credential the browser keeps can restore it without a new ceremony	A	Does not hold	1 / 3	GUV-5, GUV-13, EVT-01
INV-03	A passkey public key is stored only after a verified registration ceremony	A	Does not hold	0 / 4	GUV-4, EVT-02
INV-04	Wallet routes cannot read, sign with or export another user's embedded wallet	A	Holds	6 / 0	EVT-03, EVH-02
INV-05	The owner signing path cannot sign a high-risk action without a server-side allowlist and a decoded confirmation	A	Does not hold	2 / 4	GUV-7, EVT-04
INV-06	The asset, amount, recipient and fee shown to the user are exactly what is signed and submitted	B	Does not hold	6 / 5	GUV-8, GUV-6, GUV-11, GUV-14, EVT-05, EVT-06

TABLE CONTINUED

ID	SECURITY INVARIANT	OBJECTIVE DOMAIN	STATE	CASES PASS / FAIL	EVIDENCE
INV-07	A failed withdrawal refunds once, to the source	C	Holds	5 / 0	EVT-07, EVH-04
INV-08	The live quote a user signs matches the quote they reviewed	B	Does not hold	1 / 2	GUV-6, EVT-06
INV-09	Knowledge of a payment link ID, or of logs that contain it, is not sufficient to decrypt the link or recover its escrow key	D	Does not hold	3 / 4	GUV-1, GUV-9, EVT-09, EVH-03, EVH-08
INV-10	Only an editor changes a draft payout batch, and nobody changes a batch after approval	C	Does not hold	1 / 3	GUV-3, GUV-10, EVT-11
INV-11	Ledger credits and payouts come only from authenticated events and approved batches, and each takes effect once	C	Partially holds	6 / 1	Webhook verification and replay guard hold after the fix (GUV-2, EVR-01, EVT-08, EVH-07); the scheduler takes no lease (GUV-12, EVT-12)
INV-12	Untrusted input cannot grow storage or a privileged job without limit	E	Does not hold	1 / 2	GUV-4, EVT-02
INV-13	Key material and secrets do not leave through responses, logs or errors	D	Holds	2 / 0	EVH-05

Evidence register

Every EVT row points at a real *.security.test.ts file sitting beside the code it tests. Paths are shortened: app/api/.../route.ts handlers appear as api/<path>, and the src/ and features/ prefixes are dropped.

EVIDENCE ID	TYPE	FLOW	SCENARIO	INVARIANT	FINDING	WHAT IT IS
EVR-01	Remediation	04	TM-02	INV-11	GUV-2	9c3e1a7f04b2d86e5a1f7c20e4d93b58a6f2c071
EVH-01	Validation	01	TM-13	INV-01		Identity binding sound: the verified token account is compared to the canonicalized claimed wallet; a path or body parameter is never used to select the account
EVH-02	Validation	03	TM-14	INV-04		Wallet-ID substitution: the expected wallet ID is re-derived from the caller's own link row; a foreign ID returns 404
EVH-03	Validation	06	TM-16	INV-09		Enumeration closed: no listing endpoint, no owner column, and the ID space is too large to search
EVH-04	Validation	04	TM-15	INV-07		Refund destination is bound to the source at request time; the refund state machine has one terminal transition
EVH-05	Validation	03	TM-19	INV-13		No application code logs exported key material; the export response is ciphertext for the requester's key
EVH-06	Validation	01	TM-17	INV-01		Deployed cache matrix: every API response leaves as no-store at both auth levels; cross-user probes return 401
EVH-07	Validation	04	TM-18	INV-11		After the fix, the event-ID insert and the credit share one transaction under a unique constraint
EVH-08	Validation	06	TM-01	INV-09		Link-claim PoC: escrow key recovered from the link ID alone and a claim signed with it; no live link claimed
EVT-01	Negative	01	TM-05	INV-02	GUV-5	auth/actions/signOut
EVT-02	Boundary	02	TM-04	INV-03, INV-12	GUV-4	api/passkeys/credentials

TABLE CONTINUED

EVIDENCE ID	TYPE	FLOW	SCENARIO	INVARIANT	FINDING	WHAT IT IS
EVT-03	Negative	03	TM-14	INV-04		api/wallets/[walletId]
EVT-04	Negative	03	TM-07	INV-05	GUV-7	api/wallets/[walletId]/sign
EVT-05	Invariant	04	TM-08	INV-06	GUV-8	send/components/ConfirmSend
EVT-06	Invariant	05	TM-06	INV-06, INV-08	GUV-6	swap/machines/swapSubmitMachine
EVT-07	Invariant	04	TM-15	INV-07		withdraw/machines/refundMachine
EVT-08	Regression	04	TM-02, TM-18	INV-11	GUV-2	api/webhooks/settlement
EVT-09	Negative	06	TM-01	INV-09	GUV-1 GUV-9	api/links/[linkId]
EVT-10	Regression	01	TM-17	INV-01		api/activity cache headers
EVT-11	Negative	07	TM-03	INV-10	GUV-3	api/payouts/[batchId]
EVT-12	Invariant	07	TM-10	INV-11	GUV-12	jobs/payoutScheduler