Withdrawal-path self-assessment checklist

Forty-six yes / no / unknown questions about the route a withdrawal takes from a customer request or back-end command to a signed, broadcast transaction. Answer them with the engineers who run the system, not from the architecture diagram. Anything you answer "no" or "unknown" is a place to look, not a verdict.

– How to use it

1. Put an engineer who runs the path, a risk or compliance owner and someone from the team that operates the signing infrastructure in one room for 90 minutes.
2. Answer each question from evidence you can show (a config, a log, a test), not from memory. If you cannot show it, answer "unknown".
3. For each "no" or "unknown", write an owner and a date.
4. Repeat after any change to the signing set-up, a wallet or custody vendor, or the approval workflow.
5. Questions 3.6 to 3.13 apply only if you use what they ask about: pre-signed transactions, multisig, an HSM or MPC. If you do not, write “not applicable”.

– What this is not

It is not an audit, a certification, a regulatory submission or legal advice. Answering "yes" to every item does not mean the path is secure; it means these questions did not find a gap. It is a way to find work before someone else does.

01 Origin: who can start a withdrawal

1.1 Can you list every system, role and API key that can create a withdrawal request, including admin tools and back-office scripts, and is every one logged with who, from where and under which approval?

□ Yes □ No □ Unknown

1.2 Can an internal user create a withdrawal for a customer account without that customer's authenticated action and a second internal approver?

□ Yes □ No □ Unknown

1.3 Is a change to a customer's withdrawal address, or the creation of a new one, subject to a delay or a second factor before it can be used?

□ Yes □ No □ Unknown

1.4 Can a withdrawal be created while the customer's session, device or login anomaly checks are failing?

□ Yes □ No □ Unknown

---

02 Approval: who can approve or alter

2.1 Is the number of independent approvers for each withdrawal size band written down and enforced by the system, not by procedure alone?

□ Yes □ No □ Unknown

2.2 Can one person, one workstation or one set of credentials satisfy more than one approval step?

□ Yes □ No □ Unknown

2.3 After approval, can the amount, asset, destination or fee still be changed? By whom, and does a change reset the approval?

□ Yes □ No □ Unknown

2.4 Can an approver see the real destination, amount and asset, or only a reference or an internal ID?

□ Yes □ No □ Unknown

2.5 Are approvals tied to the exact request content (for example a hash), so approving one request cannot be reused on another?

□ Yes □ No □ Unknown

2.6 Is the approver list reviewed on a schedule, and are leavers and role changes removed within a stated time?

□ Yes □ No □ Unknown

---

03 Signing: does the signer see what executes

3.1 Does each signer see the destination, amount and asset on a screen or device that the instruction server and the operator's laptop cannot alter?

□ Yes □ No □ Unknown

3.2 Is the signed transaction compared, byte for byte or by field, with the approved request before it is broadcast? By which component, and is that component independent of the one that built the transaction?

□ Yes □ No □ Unknown

3.3 If the instruction server or the vendor's interface were compromised, what could the attacker add to, change in or replay on a request? Has anyone tested it?

□ Yes □ No □ Unknown

3.4 Where are keys or key shares held, and which people and systems can reach each one? Can one person reach enough to sign alone?

□ Yes □ No □ Unknown

3.5 Is there a documented key ceremony, with evidence kept, for generation, rotation and recovery?

□ Yes □ No □ Unknown

3.6 Are pre-signed transactions, durable nonces or standing approvals in use? If yes, who can use them and how are they revoked?

□ Yes □ No □ Unknown □ N/A

3.7 For multisig or MPC, are owners, thresholds, modules, delegates and policies exported and reviewed regularly, and is a change to any of them an alert?

□ Yes □ No □ Unknown □ N/A

3.8 If you use an HSM or similar device, are signing keys generated inside it and non-exportable, and is that enforced by the device configuration rather than by procedure?

□ Yes □ No □ Unknown □ N/A

3.9 Do HSM administration actions (policy changes, backup and restore, firmware or partition changes) need a quorum of separate people, and are they logged to storage that the HSM administrators cannot edit?

□ Yes □ No □ Unknown □ N/A

3.10 Is the signing policy (limits, destination allow-lists, approvals) enforced inside the HSM or its signing application, so that a compromised caller cannot bypass it?

□ Yes □ No □ Unknown □ N/A

3.11 If you use MPC, do the parties that hold shares run in independent administrative domains (different operators, cloud accounts or vendors), so that compromising one domain cannot reach the signing threshold?

□ Yes □ No □ Unknown □ N/A

3.12 Does each MPC node verify the content of the transaction it signs, independently of the coordinator that built the request?

□ Yes □ No □ Unknown □ N/A

3.13 Are share refresh, node replacement, threshold changes and recovery of a lost share governed by a quorum, tested, and alerted on?

□ Yes □ No □ Unknown □ N/A

---

04 Limits and detection: what stops a very large outflow

4.1 Are there per-withdrawal, per-hour and per-day limits on hot-wallet outflow, enforced outside the application that creates the withdrawal?

□ Yes □ No □ Unknown

4.2 Is there an automatic hold or circuit-breaker when outflow exceeds a threshold or deviates from the normal pattern, and who can release it?

□ Yes □ No □ Unknown

4.3 Is a withdrawal to a new or unusual destination treated differently from a repeat one?

□ Yes □ No □ Unknown

4.4 Would you detect an altered request before broadcast, and how quickly after broadcast would a human be paged?

□ Yes □ No □ Unknown

4.5 Is there a tested, written procedure to freeze withdrawals, and can it be executed by someone who is not the person who raised the incident?

□ Yes □ No □ Unknown

4.6 Is the balance of the hot wallet capped, and what moves funds to it from warm or cold storage and under whose approval?

□ Yes □ No □ Unknown

---

05 Ledger consistency: does what leaves match what is owed

5.1 Is the user balance debited and the withdrawal locked in one atomic step, so a withdrawal and a concurrent order cannot both spend the same funds?

□ Yes □ No □ Unknown

5.2 If a withdrawal fails, is cancelled or is retried, is the lock released exactly once?

□ Yes □ No □ Unknown

5.3 Is every withdrawal request idempotent under retries, timeouts and duplicate submissions, and what is the scope and lifetime of the idempotency key?

□ Yes □ No □ Unknown

5.4 Is there a regular reconciliation between the internal ledger, the on-chain wallet balances and any custodian's records, with a named person who reads the exceptions?

□ Yes □ No □ Unknown

5.5 Are fees and rounding applied in a defined direction, with the unit and precision per asset written down?

□ Yes □ No □ Unknown

5.6 Can deposits that arrive on-chain be credited twice, or credited and then reorganised away?

□ Yes □ No □ Unknown

06 Third parties: who else can touch the path

6.1 Do you have an inventory of every vendor, contractor and service that can read, queue, alter or sign a withdrawal?

□ Yes □ No □ Unknown

6.2 What does each vendor's staff access reach: the request queue, the signing environment, both? How is it granted, logged and reviewed?

□ Yes □ No □ Unknown

6.3 Is a vendor's production environment a path into yours? If it were compromised, what is the worst a request could be changed into?

□ Yes □ No □ Unknown

6.4 Does each vendor contract state security incident notice times and your right to evidence of their controls, and is it exercised?

□ Yes □ No □ Unknown

6.5 Have you tested a recovery from a vendor outage or exit without moving funds in a hurry?

□ Yes □ No □ Unknown

6.6 Who in your organisation owns each vendor relationship for security, not only for the contract?

□ Yes □ No □ Unknown

---

07 Change and recovery

7.1 Is a change to the withdrawal path (code, config, signing policy, vendor) reviewed by someone other than its author before release, with the approval recorded?

□ Yes □ No □ Unknown

7.2 Can one engineer deploy a change to the withdrawal path, or to the systems that construct transactions, without a second person?

□ Yes □ No □ Unknown

7.3 Is there a tested way to roll back such a change quickly?

□ Yes □ No □ Unknown

7.4 Are logs for the path kept long enough, and kept where an attacker with admin access to the application cannot edit them? Is the retention period decided, not default?

□ Yes □ No □ Unknown

7.5 Do you rehearse a withdrawal-path incident, including who decides to stop withdrawals and who talks to the regulator?

□ Yes □ No □ Unknown

– Reading your answers

- A "no" in sections 2 and 3 deserves attention first. Those are where the content of a request can be changed on its way to the chain.
- An "unknown" counts as a "no" until evidence exists. Finding out is itself the work.

- There is no score on purpose. A number invites a pass mark, and there is none.

---

deebug.io — security engineering & assurance · operated by 株式会社eMoment Japan KK

research@deebug.io · security.txt · privacy · LinkedIn