CHECKLIST

Fintech Penetration Testing Checklist

What to test in a fintech pentest, mapped to PCI DSS v4.0.1, SOC 2 and OWASP

Prepared by Software Secured

Use this checklist to scope and track a penetration test of your fintech platform. It covers account takeover, account-level authorization, payment and money movement logic, KYC onboarding, partner APIs, mobile banking apps and cloud infrastructure. Each item is mapped to PCI DSS v4.0.1, SOC 2 or OWASP standards, so your results double as audit and bank partner evidence.

Key takeaways

- Broken object level authorization (BOLA) ranks first in the OWASP API Security Top 10. Test that no customer can read or move another customer's money.
- Payment logic flaws such as transaction tampering and race conditions on balances need manual testing. Scanners rarely catch them.
- Step-up MFA on high-risk actions, like adding a payee or raising limits, matters as much as MFA at login.
- KYC document uploads are a prime target. Validate uploaded files, and make sure stored identity documents can't be reached by guessing URLs.
- PCI DSS v4.0.1 requires internal and external pentests at least every 12 months and after significant changes, plus retesting of exploitable findings (Requirement 11.4).
- Before testing starts, give testers at least two funded test accounts per role and clear rules of engagement for money movement.

How to use this checklist

Work through each section with your engineering lead before scoping a pentest, then tick the Done column to track coverage. Items marked PRIORITY are the ones to test first if your engagement is time-boxed.

PCI PCI DSS v4.0.1 **SOC 2** AICPA Trust Services Criteria **API** OWASP API Security Top 10 (2023) **MASVS** OWASP MASVS v2

0 Before you start: scope confirmation

Confirm these before testing begins. Most delays in fintech pentests come from missing test accounts or unclear rules for money movement.

- [ ] Testing environment confirmed: staging or UAT that mirrors production, with payment processors in sandbox mode
- [ ] At least two test customers per role (retail, business admin, business sub-user, support agent), each with funded test balances
- [ ] Test card numbers and sandbox bank accounts only. No live cardholder data or real customer PII in the test environment
- [ ] API documentation (OpenAPI or Postman collections) for customer, partner and internal APIs
- [ ] Current iOS and Android builds, ideally unstripped or debug builds, plus any SDKs you ship to partners
- [ ] List of third-party integrations in scope: payment processors, KYC and identity vendors, banking partners, data aggregators
- [ ] Written rules of engagement for money movement: which flows testers may execute, transaction limits and who to call if something looks real

1 Authentication and account takeover

Applies to every customer-facing and staff-facing application.

Account takeover is the shortest path to fraud. Test every way a user proves who they are, and every way an attacker could skip that step.

Test item	Reference	Done
Credential stuffing and brute-force protections on login, including rate limiting and lockout **PRIORITY**	API2; SOC 2 CC6.1	☐
MFA enforced on login and stepped up for high-risk actions: adding a payee, changing contact details, raising limits **PRIORITY**	API2; SOC 2 CC6.1	☐
MFA required for all staff and admin access into the cardholder data environment	PCI 8.4.2	☐
Session and token handling: JWT signature validation, expiry, refresh token rotation and revocation on logout or password change	API2	☐
Password reset and account recovery: OTP brute force, reset token leakage, user enumeration	API2	☐
New device enrollment and trusted device changes require re-verification	API2; SOC 2 CC6.1	☐

Common gap

MFA protects the login screen but not the actions that move money. Attackers with a stolen session go straight to adding a payee, so test step-up authentication on every high-risk action.

2 Authorization and account data (BOLA)

Applies to any API or screen that returns or changes account, balance or transaction data.

The core question: can one customer read or move another customer's money? Broken object level authorization (BOLA) is the most common critical API flaw.

Test item	Reference	Done
Object-level authorization on account, balance, transaction, statement and card endpoints: user A cannot read or act on user B's records **PRIORITY**	API1; PCI 6.2.4	☐
Function-level authorization: customers cannot call support, admin or internal operations endpoints **PRIORITY**	API5; PCI 6.2.4	☐
Property-level authorization and mass assignment: users cannot set fields such as balance, KYC status, credit limit or fee tier **PRIORITY**	API3	☐
Business account roles enforced server-side: initiators cannot approve their own payments, and dual approval cannot be bypassed	API5; SOC 2 CC6.3	☐
Tenant isolation for B2B and embedded finance customers: one business cannot see another business's users or data	SOC 2 CC6.1	☐

Common gap

Testing with a single account hides BOLA entirely. Provide at least two customers per role so testers can attempt cross-account access on every endpoint.

3 Payments and money movement

Applies to transfers, payouts, card transactions, top-ups, refunds and rewards.

Business logic flaws in payment flows rarely show up in scanners. They need a tester who thinks like a fraudster.

Test item	Reference	Done
Transaction tampering: amount, currency, fee or recipient cannot be changed between confirmation and submission **PRIORITY**	API6; PCI 6.2.4	☐
Race conditions on balance updates: concurrent withdrawals or transfers cannot spend the same funds twice **PRIORITY**	PCI 6.2.4	☐
Replay protection and idempotency keys on payment initiation requests	API6	☐
Input edge cases: negative, zero, fractional and very large amounts, rounding and currency conversion abuse	PCI 6.2.4	☐
Transfer limits and velocity controls cannot be bypassed by splitting requests or switching channels	API6	☐
Refund, chargeback, promo code and rewards flows cannot be abused for credit or cash-out	API6	☐
Script integrity and tamper detection on any payment page you host	PCI 6.4.3; PCI 11.6.1	☐

Software Secured note

Our fintech pentest plan simulates fraud, payment abuse and business logic abuse against your real flows, not just generic injection checks.

4 Onboarding and KYC

Applies to signup, identity verification and any flow that collects identity documents.

Onboarding collects the most sensitive data you hold and is a common route to synthetic identity fraud.

Test item	Reference	Done
Document upload handling: file type validation, malicious files, path traversal and SSRF through URL-based uploads **PRIORITY**	API7; PCI 6.2.4	☐
Uploaded identity documents are not reachable through guessable, sequential or public URLs **PRIORITY**	API1; SOC 2 CC6.1	☐
Features that require verification cannot be reached before KYC is complete, and verification status cannot be set by the client	API3; API5	☐
Callbacks and webhooks from identity verification vendors are authenticated and cannot be spoofed	API10	☐
Signup and verification flows do not reveal whether an email, phone number or ID is already registered	API2	☐

Common gap

Identity documents stored in object storage with predictable file names. If one customer can guess another's document URL, you have a data breach, not a bug.

5 APIs, partners and integrations

Applies to partner APIs, open banking connections, data aggregators and payment processor integrations.

Fintech platforms trust a long chain of third parties. Test what happens when that trust is abused.

Test item	Reference	Done
Complete API inventory: deprecated versions and undocumented endpoints are identified and either secured or retired **PRIORITY**	API9	☐
Webhook signatures from payment processors and partners are verified, and replayed webhooks are rejected **PRIORITY**	API10	☐
Data received from aggregators and partner APIs is validated before it changes balances or account state	API10	☐
OAuth scope for partners and aggregators grant only the access each integration needs	API5; SOC 2 CC6.3	☐
Server-side request forgery is not possible through integration or callback URLs	API7	☐
Rate limits on expensive operations such as statement exports, search and bulk transfers	API4	☐

Common gap

Old API versions keep running after a mobile release, without the authorization fixes made in the new version. Ask testers to enumerate every version still reachable.

6 Mobile banking apps

Applies if you ship iOS or Android apps, or SDKs that partners embed in theirs.

The mobile app sits on a device you do not control. Assume attackers will decompile it, modify it and intercept its traffic.

Test item	Reference	Done
Tokens, account numbers and PII are stored in the iOS Keychain or Android Keystore, not in plain local storage **PRIORITY**	MASVS-STORAGE	☐
No hardcoded API keys, secrets or encryption keys in the app binary **PRIORITY**	MASVS-CRYPTO	☐
TLS enforced on all traffic, with certificate pinning on sensitive endpoints	MASVS-NETWORK; PCI 4.2.1	☐
Biometric login is tied to a cryptographic key and cannot be bypassed locally	MASVS-AUTH	☐
Deep links and app intents cannot trigger transactions or expose data without authentication	MASVS-PLATFORM	☐
Root and jailbreak detection, plus resistance to repackaging and runtime tampering	MASVS-RESILIENCE	☐
Sensitive screens are hidden in the app switcher, and account data is not written to logs	MASVS-STORAGE	☐

Software Secured note

Mobile testing should always include the backend APIs the app calls. Most serious mobile findings turn out to be server-side authorization flaws reached through the app.

7 Cloud, data and cryptography

Applies to the infrastructure that stores and processes payment and customer data.

Validate that a foothold in one service cannot become access to payment systems or customer data.

Test item	Reference	Done
Stored primary account numbers (PAN) are rendered unreadable, and sensitive authentication data is not retained after authorization **PRIORITY**	PCI 3.5.1; PCI 3.3.1	☐
Encryption keys are managed in a KMS or HSM with restricted access and rotation, and never stored in code or config	PCI 3.6.1; PCI 3.7	☐
Cloud IAM follows least privilege, and testers cannot move laterally from application workloads to payment systems **PRIORITY**	SOC 2 CC6.1; SOC 2 CC6.3	☐
Storage buckets, snapshots and backups containing customer data are not publicly accessible	SOC 2 CC6.1	☐
Network segmentation isolates the cardholder data environment from other systems	PCI 11.4.5	☐
Security events are logged, and logs contain no card numbers, secrets or full identity documents	PCI 10.2.1; PCI 3.5.1	☐

Common gap

Segmentation exists on the network diagram but not in the cloud account. Overly broad IAM roles let a compromised web service read from the payments database directly.

8 Evidence for auditors and bank partners

Applies to every engagement. This is what you hand over when the test is done.

A pentest only helps your audit or partner review if the deliverables match what reviewers ask for.

Deliverable	Reference	Done
Documented penetration testing methodology covering the full CDE perimeter, application layer and network layer	PCI 11.4.1	☐
Internal and external penetration tests at least every 12 months and after significant changes, performed by a qualified, organizationally independent tester	PCI 11.4.2; PCI 11.4.3; SOC 2 CC4.1	☐
Retest report confirming exploitable findings were corrected	PCI 11.4.4; SOC 2 CC4.2	☐
Segmentation test results	PCI 11.4.5; PCI 11.4.6	☐
Full report with scope, methodology, findings and severity ratings for auditors, plus an executive summary for leadership	SOC 2 CC4.1	☐
Pentest certificate that proves your application was tested without exposing what was found, for auditors, bank partners and enterprise customers	SOC 2; ISO 27001; partner due diligence	☐

Software Secured note

Bank partners and enterprise customers often want proof of testing without the full findings. Every Software Secured engagement includes a pentest certificate (executive summary) that does exactly that, and clients can download it anytime from the Reports & Certificates tab in Portal.

Timing reference

Activity	Recommended timing
Internal and external penetration test	At least every 12 months, and after any significant change (PCI 11.4.2, 11.4.3)
Segmentation testing	Every 12 months for merchants, every 6 months for service providers (PCI 11.4.5, 11.4.6)
External vulnerability scans by an ASV	Quarterly (PCI 11.3.2)
Retest of exploitable findings	After remediation, before submitting evidence to auditors or partners
Pentest of a new payment flow or major mobile release	Before launch
Pentest ahead of bank partner due diligence	Allow time to remediate and retest before the review starts

For PCI DSS compliance levels, SAQ types and the full Requirement 11 breakdown, see our PCI DSS v4.0.1 Pentesting Guide.

What this checklist does not cover

- Social engineering and phishing simulations: available as a separate social engineering testing engagement
- Red teaming: available as a separate red teaming engagement
- Secure code review: available as a separate secure code review engagement
- Quarterly ASV vulnerability scanning and the PCI DSS assessment itself (ROC or SAQ), which are handled by your ASV and QSA
- Fraud detection models, AML transaction monitoring and sanctions screening

Ready to test your fintech platform?

Our fintech penetration testing covers every section of this checklist, with manual testing of payment flows, APIs and mobile apps, three rounds of retesting, reports built for PCI DSS and SOC 2, and a pentest certificate you can share in bank partner and enterprise reviews.

Book a consultation Explore fintech penetration testing

References: PCI DSS v4.0.1 (PCI Security Standards Council, June 2024); AICPA 2017 Trust Services Criteria with 2022 Revised Points of Focus; OWASP API Security Top 10 (2023); OWASP MASVS v2.

This checklist is for information only and is not legal, audit or compliance advice. Requirements vary by your role (merchant or service provider), your assessor and your system scope. Confirm obligations with your QSA, auditor or legal counsel.

Last reviewed: October 2026.