

PAYMENT GATEWAY & PROCESSOR SECURITY.

A single card payment passes through a gateway, a processor, an acquiring bank, the card network, and an issuing bank before it's actually approved. **Each hop is a distinct system with its own security responsibilities, and authorization is only the first of three separate events before money actually moves.**

- TRANSACTION ROUTING
- HOSTED PAYMENT PAGES
- WEBHOOK VERIFICATION
- API KEY MANAGEMENT

01 THE MECHANISM



Five Parties, One Transaction

A card payment passes through a gateway, a processor, the merchant's acquiring bank, the card network, and the cardholder's issuing bank — **each hop is a distinct system with its own security responsibilities.**



Authorization, Clearing, and Settlement Differ

A real-time authorization hold happens in seconds, but the actual movement of funds in clearing and settlement **can take one to three business days**, with different integrity requirements at each phase.



Hosted Pages Keep Card Data Off the Merchant

Rather than the merchant's own backend ever touching a card number, a gateway-hosted checkout page or iframe sends card data **directly from the customer's browser to the gateway.**

02 THE FLOW

The Properly Scoped Integration

CONTAINED

1

The checkout page embeds the gateway's own hosted fields, so **raw card data never reaches the merchant's own server** at any point.

2

The secret API key lives only in server-side configuration, never in client-side code or a public repository, and is **scoped to only the permissions it needs.**

3

Every webhook from the gateway is verified against a signature before the merchant's system acts on it, so a **forged "payment successful" callback can't trick it.**

The Naive Integration

EXPOSED

1

The merchant's own checkout form collects raw card data directly, meaning **every request to the merchant's server is now in PCI DSS scope** and worth attacking.

2

A developer accidentally commits the gateway's secret API key to a public code repository, and an **attacker starts processing fraudulent charges or refunds** directly.

3

The merchant's backend trusts an unverified webhook payload as proof of payment, and an attacker **sends a forged event to receive goods without paying.**

03 CORE PILLARS & COVERAGE

PILLAR	WHAT IT COVERS
Data Capture Isolation	Hosted payment pages or gateway-provided fields that keep raw card data off the merchant's own servers entirely.
API Credential Management	Scoped, server-side-only secret keys, distinct from client-safe publishable keys, with no exposure in client code or public repositories.
Webhook Signature Verification	Cryptographically verifying that a payment event notification actually originated from the gateway before acting on it.
Fraud Scoring at Authorization	Real-time velocity, AVS, and CVV checks run by the processor or acquirer alongside the issuer's own fraud decisioning.

04 DETECTION APPROACHES COMPARED

APPROACH	SUFFICIENT ALONE?	WHAT IT MISSES
Merchant's Own Server Collects Raw Card Data	No	Every system that touches raw card data becomes a high-value target and expands PCI DSS scope well beyond a hosted-page integration.
Secret API Keys Embedded in Client-Side Code	No	Client-side code is visible to anyone, turning a leaked secret key into direct, unauthorized access to process charges or refunds.
Webhooks Trusted Without Signature Verification	No	Any unauthenticated system can send a forged event, tricking the merchant's backend into treating a fake payment as real.
Fraud Scoring Only at the Issuer, Not the Acquirer	No	Relying solely on the issuing bank's decision skips the velocity and address-verification checks a processor is positioned to catch earlier.
Hosted Data Capture + Scoped Server-Side Keys + Verified Webhooks + Acquirer-Side Scoring	Yes	Closes the gaps at each hop between the checkout page and the actual movement of funds.

PAYMENT GATEWAY & PROCESSOR.

ROUTE · VERIFY · SETTLE

05 PROGRAM OPERATIONS

Authorization, clearing, and settlement are genuinely different events with different timelines. Fraud caught at authorization is stopped before it happens; fraud caught after settlement is a recovery problem.

A publishable key and a secret key serve completely different purposes. Leaking the secret key is a direct path to unauthorized transaction processing, not just a data exposure.

Webhook forgery is a well-documented, concrete vulnerability class. Any integration that treats an unverified callback as proof of payment has no way to confirm it actually came from the gateway.

Hosted payment pages remove an entire category of attack surface, not just reduce PCI DSS scope as a compliance nicety, by keeping raw card data off the merchant's own infrastructure.

06 CONTROL MAPPING

PCI DSS SCOPE REDUCTION GUIDANCE

PCI SSC guidance specifically addressing how hosted payment pages and iframes affect a merchant's cardholder data environment scope.

WEBHOOK SIGNATURE STANDARDS

Cryptographic signing conventions gateways use so merchants can verify a payment event notification's authenticity before acting on it.

AVS & CVV VERIFICATION

Address Verification System and card verification value checks processors run as part of real-time authorization fraud scoring.

CARD NETWORK PROCESSOR REGISTRATION

Requirements card networks impose directly on processors and acquirers, distinct from and in addition to a merchant's own PCI DSS obligations.

07 WHAT THIS DOES NOT SOLVE

Fraud at the Cardholder Level

Gateway and processor controls protect the transaction pipeline; they **don't stop a stolen card number** from being used successfully if it passes every automated check.

Merchant Application Logic Flaws

A gateway can't fix a merchant's own **broken authorization on order or refund endpoints** that let one customer manipulate another's transaction data.

Settlement Timing Risk

Even a fully secured pipeline doesn't change the **one-to-three-day settlement window**, during which a merchant's cash flow still depends on fund-movement timing.

Issuer-Side Decisions

A gateway or processor can score and flag a transaction, but the **final approve-or-decline decision rests with the issuing bank**, outside the merchant's control.

08 MYTHS VS. REALITY

MYTH	REALITY
"Using a payment gateway means PCI DSS isn't the merchant's concern."	A hosted-page integration significantly reduces scope, but the merchant still has obligations for whatever touches their own systems, including webhooks and API credentials .
MYTH	REALITY
"A leaked API key is just a data exposure, not a financial risk."	A leaked secret key can let an attacker directly process fraudulent charges or refunds against the merchant's account.
MYTH	REALITY
"If a webhook comes from the gateway's known IP range, it's safe to trust."	IP-based trust is weak and spoofable; a cryptographic signature check is what actually verifies a webhook's authenticity.
MYTH	REALITY
"Authorization means the payment is complete."	Authorization is just a real-time hold; clearing and settlement are separate events that can take one to three business days to move the funds.
MYTH	REALITY
"Fraud scoring is the issuing bank's job, not the processor's."	Processors and acquirers run their own velocity, AVS, and CVV checks as part of authorization, working alongside the issuer's decision , not instead of it.

09 SPEAK THE PROGRAM — KEY TERMS

PAYMENT GATEWAY

The system that captures and routes a transaction from a merchant's checkout to the payment processor, **often the first point where card data is encrypted or tokenized**.

PAYMENT PROCESSOR

The entity that communicates with the card networks and banks to **actually authorize a transaction** on the merchant's behalf.

ACQUIRING BANK

The merchant's bank, which **receives funds on the merchant's behalf** after a transaction clears and settles.

AUTHORIZATION

The real-time check and hold confirming funds or credit are available, **distinct from the actual movement of money**.

CLEARING & SETTLEMENT

The batch-oriented exchange of transaction details and the eventual movement of funds between banks, **which can take one to three business days**.

HOSTED PAYMENT PAGE

A gateway-served checkout page or embedded field that captures card data directly from the customer's browser, **keeping it off the merchant's own servers**.