Online Payments Processing Platform

Owner: A development team
Reviewer: A security architect
Contributors: development engineers, product managers, security architects
Date Generated: Tue Oct 07 2025

Executive Summary

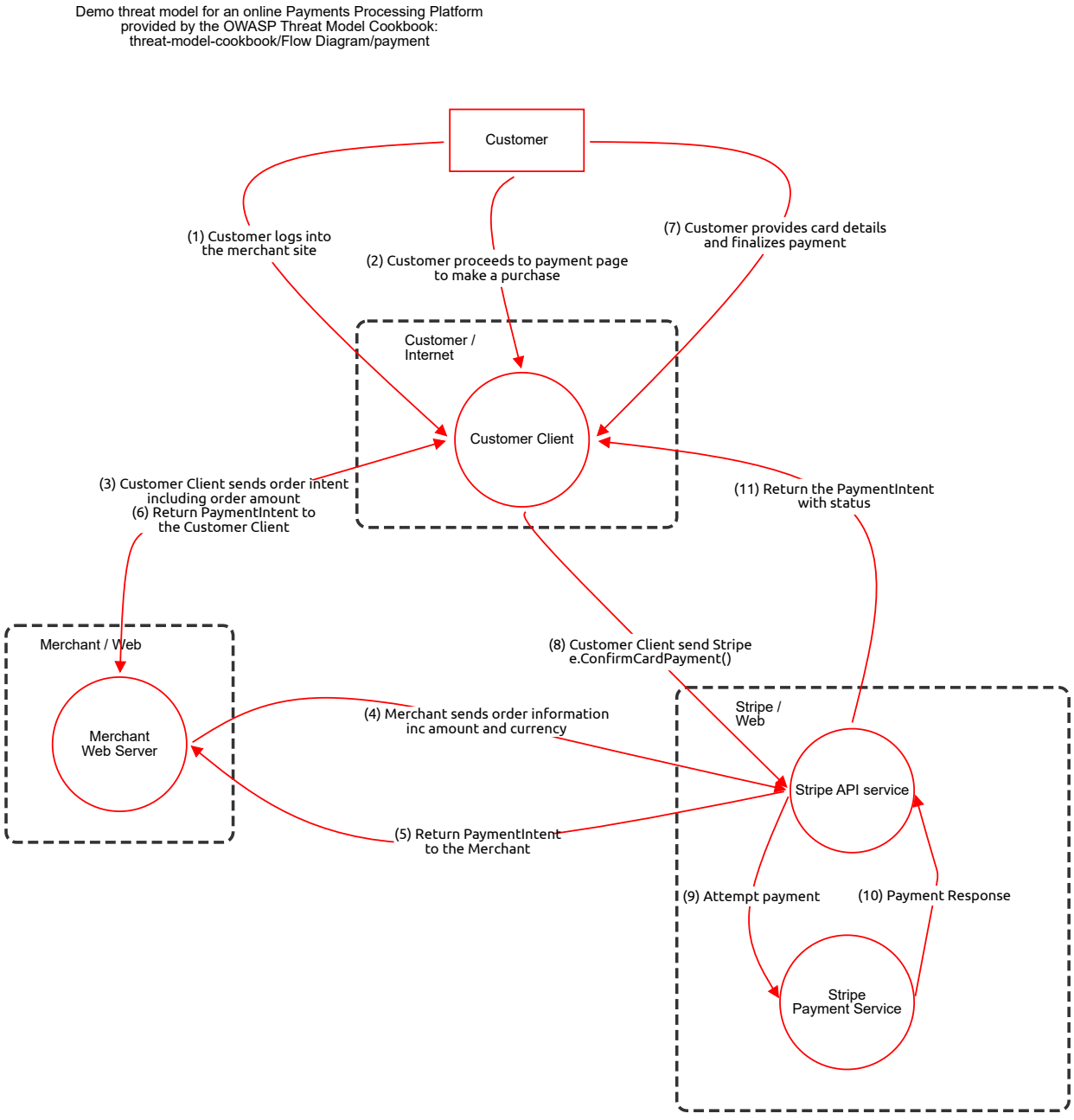
High level system description

This threat model has been provided by the OWASP Threat Model Cookbook:
threat-model-cookbook/Flow Diagram/payment

Summary

Total Threats	22
Total Mitigated	0
Total Open	22
Open / Critical Severity	0
Open / High Severity	16
Open / Medium Severity	6
Open / Low Severity	0

Payment


Demo threat model for an online Payments Processing Platform
provided by the OWASP Threat Model Cookbook:
threat-model-cookbook/Flow Diagram/payment
Customer
(1) Customer logs into
the merchant site
(2) Customer proceeds to payment page
to make a purchase
(7) Customer provides card details
and finalizes payment
Customer /
Internet
Customer Client
(3) Customer Client sends order intent
including order amount
(6) Return PaymentIntent to
the Customer Client
(11) Return the PaymentIntent
with status
Merchant / Web
Merchant
Web Server
(8) Customer Client send Stripe
e.ConfirmCardPayment()
(4) Merchant sends order information
inc amount and currency
(5) Return PaymentIntent
to the Merchant
Stripe /
Web
Stripe API service
(9) Attempt payment
(10) Payment Response
Stripe
Payment Service

Payment

Customer (Actor)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing of Customer Identity	Spoofing	High	Open		The Customer actor, positioned externally at (450,120) outside the Customer/Internet trust boundary, initiates multiple flows including login, purchase intent, and card details submission to the Customer Client process. An attacker could spoof the customer's identity to gain unauthorized access and perform fraudulent transactions.	Implement multi-factor authentication (MFA) for customer login and use client certificates or device binding for payment confirmations to prevent impersonation.

Customer Client (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure in Customer Client	Information Disclosure	High	Open		The Customer Client process, located at (450,375) within the Customer/Internet trust boundary, receives sensitive data flows such as card details from the external Customer actor and forwards payment intents to the Merchant Web Server. Without encryption flags enabled, sensitive payment information could be disclosed during processing or storage in the client.	Enforce end-to-end encryption for all sensitive data handling within the client, use secure storage mechanisms like encrypted local storage, and avoid logging sensitive data.
	Elevation of Privilege via Client Vulnerabilities	Elevation of Privilege	Medium	Open		The Customer Client process interacts with external flows from the Customer actor and internal flows to the Merchant Web Server, potentially allowing an attacker to exploit client-side vulnerabilities (e.g., XSS) to elevate privileges and manipulate payment flows.	Apply content security policy (CSP), sanitize all inputs, and regularly update client-side libraries to mitigate exploitation risks.

(1) Customer logs into the merchant site (Data Flow)

Description: OAuth

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Login Flow	Tampering	High	Open		The flow '(1) Customer logs into the merchant site' from Customer actor (external at 450,120) to Customer Client process (inside boundary at 450,375) uses HTTPS protocol but has isEncrypted flag set to false, crossing from external zone to internal, allowing potential tampering with login credentials over the public network.	Enforce TLS 1.3 with certificate pinning and implement integrity checks like HMAC for login requests to prevent tampering.

(2) Customer proceeds to payment page to make a purchase (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure in Purchase Intent Flow	Information Disclosure	Medium	Open		The flow '(2) Customer proceeds to payment page to make a purchase' connects the external Customer actor to the Customer Client process, traversing potential public network exposure without explicit encryption, risking disclosure of purchase details like item information.	Ensure all flows use HTTPS with HSTS enabled and avoid transmitting unnecessary sensitive data in purchase intents.

(7) Customer provides card details and finalizes payment (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Card Details Submission	Tampering	High	Open		The flow '(7) Customer provides card details and finalizes payment' from external Customer actor to Customer Client process lacks encryption flag and protocol specification, crossing from untrusted external zone, enabling attackers to tamper with card details during transmission.	Use tokenization for card details (e.g., Stripe Elements) and enforce mTLS for sensitive submissions to ensure integrity.
	Information Disclosure of Card Information	Information Disclosure	High	Open		This flow transmits highly sensitive card details from the external Customer actor across a potential public network to the Customer Client, with isEncrypted false, increasing risk of interception and disclosure.	Implement PCI DSS compliant encryption and never transmit full card details; use iframe isolation for payment forms.

(3) Customer Client sends order intent including order amount (6) Return PaymentIntent to the Customer Client (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Bidirectional Order Intent Flow	Tampering	High	Open		The bidirectional flow '(3) Customer Client sends order intent including order amount (6) Return PaymentIntent to the Customer Client' between Customer Client (450,375) and Merchant Web Server (65,670) crosses trust boundaries from Customer/Internet to Merchant/Web zones without encryption, allowing tampering with order amounts or PaymentIntents.	Apply digital signatures to order intents and PaymentIntents, and use secure channels with TLS 1.2+ for all bidirectional communications.

(9) Attempt payment (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service on Payment Attempt Flow	Denial of Service	Medium	Open		The flow '(9) Attempt payment' from Stripe API service (780,720) inside Stripe/Web boundary to Stripe Payment Service (770,920) also inside, but as an internal API call, could be susceptible to DoS if not rate-limited, especially given the payment processing context.	Implement rate limiting and circuit breakers on the payment attempt endpoint to prevent resource exhaustion from malicious requests.

(10) Payment Response (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure in Payment Response	Information Disclosure	High	Open		The flow '(10) Payment Response' from Stripe Payment Service to Stripe API service, both within Stripe/Web boundary at positions (770,920) to (780,720), may contain sensitive payment status and details without specified encryption, risking internal disclosure if intercepted.	Encrypt all internal responses containing sensitive data and use access controls to limit response visibility within the Stripe services.

(11) Return the PaymentIntent with status (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with PaymentIntent Return Flow	Tampering	High	Open		The flow '(11) Return the PaymentIntent with status' from Merchant Web Server (65,670) in Merchant/Web to Customer Client (450,375) in Customer/Internet crosses trust boundaries outward, with no encryption flag, allowing tampering with payment status to mislead the client.	Sign PaymentIntents with cryptographic hashes and validate integrity on receipt in the Customer Client.

(8) Customer Client send Stripe e.ConfirmCardPayment() (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Elevation of Privilege in ConfirmCardPayment Flow	Elevation of Privilege	High	Open		The flow '(8) Customer Client send Stripe e.ConfirmCardPayment()' from Customer Client (450,375) external to Stripe API service (780,720) inside Stripe boundary crosses public networks without encryption, potentially allowing an attacker to elevate privileges by forging confirmations.	Require API keys with least privilege and implement client-side validation before sending confirmations, plus mTLS for the flow.
	Information Disclosure of Confirmation Data	Information Disclosure	High	Open		This flow transmits card payment confirmation details across boundaries from Customer/Internet to Stripe/Web over potential public network with isEncrypted false, exposing sensitive transaction data.	Tokenize all payment data in transit and enforce strict TLS configurations for API calls to Stripe.

(5) Return PaymentIntent to the Merchant (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with PaymentIntent Return to Merchant	Tampering	Medium	Open		The flow '(5) Return PaymentIntent to the Merchant' from Stripe API service (780,720) to Merchant Web Server (65,670) crosses from Stripe/Web to Merchant/Web boundaries over potential external network without encryption, risking tampering with intent details.	Use message authentication codes (MAC) for PaymentIntents and validate on the merchant side before processing.

(4) Merchant sends order information inc amount and currency (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure of Order Information	Information Disclosure	High	Open		The flow '(4) Merchant sends order information inc amount and currency' from Merchant Web Server (65,670) inside Merchant/Web to Stripe API service (780,720) in Stripe/Web crosses external boundaries with no encryption flag, exposing order amounts and currency to interception.	Encrypt order details with AES before transmission and use secure API gateways with WAF to protect sensitive data in transit.
	Spoofing of Order Submission	Spoofing	High	Open		This flow sends order information from the merchant to Stripe, but an attacker could spoof the merchant's identity to submit fraudulent orders, given the cross-boundary nature without mutual authentication.	Implement mutual TLS (mTLS) authentication between merchant and Stripe services to verify identities.

Merchant Web Server (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service on Merchant Web Server	Denial of Service	Medium	Open		The Merchant Web Server process at (65,670) within Merchant/Web boundary receives multiple ingress flows from external Customer Client, including order intents and payment confirmations, making it vulnerable to DoS attacks flooding the server.	Deploy rate limiting, DDoS protection services, and auto-scaling to handle high volumes of payment-related requests.
	Elevation of Privilege in Merchant Server	Elevation of Privilege	High	Open		As the central process handling order information and PaymentIntents, positioned inside the trust boundary and connected to external flows, vulnerabilities could allow attackers to elevate privileges and authorize unauthorized transactions.	Enforce role-based access control (RBAC) and principle of least privilege for server processes, with regular vulnerability scanning.

Stripe API service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Stripe API Service	Tampering	High	Open		The Stripe API service process at (780,720) inside Stripe/Web boundary receives sensitive flows like order information and card confirmations from external Merchant Web Server and Customer Client, crossing public networks without encryption, enabling tampering with API requests.	Validate all inputs with schema validation and use API gateways with integrity checks for incoming payment data.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Repudiation of API Transactions	Repudiation	Medium	Open		Transactions processed by the Stripe API service could be repudiated due to lack of non-repudiation mechanisms in cross-boundary flows from external entities.	Implement audit logging with timestamps and digital signatures for all API transactions to ensure non-repudiation.

Stripe Payment Service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service in Payment Service	Denial of Service	High	Open		The Stripe Payment Service process at (770,920) within the same Stripe/Web boundary receives payment attempt flows from the adjacent Stripe API service, but as a critical payment processor, it is prone to DoS disrupting financial operations.	Use redundancy, load balancing, and intrusion prevention systems to protect against DoS targeting payment processing.
	Information Disclosure from Payment Service	Information Disclosure	High	Open		This process handles final payment responses containing sensitive financial data, connected internally but potentially exposed if internal flows lack encryption, risking disclosure within the Stripe boundary.	Encrypt all data at rest and in transit within the payment service, and apply data masking for non-essential logs.