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

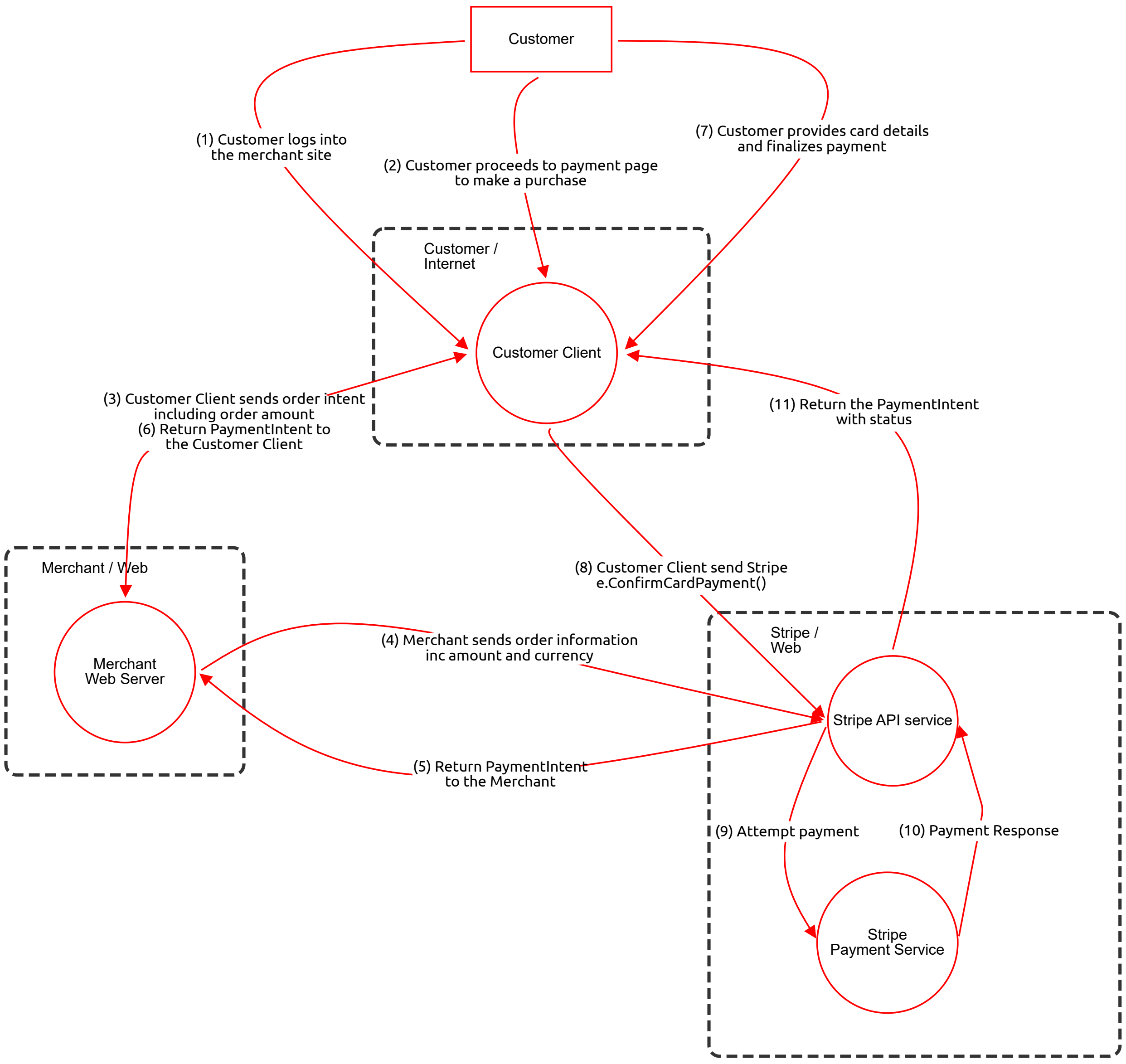
This threat model has been provided by the OWASP Threat Model Cookbook: threat-model-cookbook/Flow Diagram/payment

Summary

Total Threats	66
Total Mitigated	0
Total Open	66
Open / Critical Severity	0
Open / High Severity	46
Open / Medium Severity	20
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
Stripe / Web
Stripe API service
(5) Return PaymentIntent to the Merchant
(9) Attempt payment
(10) Payment Response
Stripe Payment Service

Payment

Customer (Actor)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing of Customer Actor	Spoofing	High	Open		The Customer actor, positioned outside all trust boundaries, could be spoofed by an attacker to initiate unauthorized interactions with the Customer Client inside the Customer/Internet trust boundary.	Implement strong authentication mechanisms such as multi-factor authentication for customer logins.
	Repudiation of Actions by Customer	Repudiation	Medium	Open		The Customer actor may deny performing actions like logging in or proceeding to payment, especially since it's external and flows originate from it to internal processes without evident logging.	Implement logging of all customer actions with timestamps and IP addresses, and require confirmations for critical actions.

Customer Client (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Customer Client Process	Tampering	High	Open		The Customer Client process, located within the Customer/Internet trust boundary, receives ingress flows from the external Customer actor over HTTPS, which could be tampered with if the client-side code is manipulated.	Use code signing and integrity checks on client-side scripts, and validate all inputs server-side.
	Information Disclosure from Customer Client	Information Disclosure	High	Open		Sensitive data like card details handled by the Customer Client could be disclosed if the process is compromised, especially with bidirectional flows crossing to Merchant Web Server.	Encrypt sensitive data in transit and at rest, and minimize data stored on the client.
	Denial of Service on Customer Client	Denial of Service	Medium	Open		The Customer Client could be targeted for DoS attacks via multiple ingress flows from the external actor, overwhelming the client within the public-facing boundary.	Implement rate limiting and CAPTCHA on client interactions.
	Elevation of Privilege in Customer Client	Elevation of Privilege	High	Open		An attacker could exploit vulnerabilities in the Customer Client to gain higher privileges, especially with flows to Stripe services crossing trust boundaries.	Apply principle of least privilege and regular security updates to the client application.

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

Description: OAuth

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing over Login Flow	Spoofing	High	Open		The login flow from external Customer to Customer Client inside Customer/Internet boundary over HTTPS could be spoofed by impersonating the customer.	Enforce mutual TLS and certificate validation for the flow.
	Tampering with Login Data	Tampering	High	Open		Data in the login flow could be tampered with as it crosses from external to internal boundary, potentially altering credentials.	Use cryptographic signing of messages and validate integrity on receipt.
	Information Disclosure in Login Flow	Information Disclosure	High	Open		Credentials could be disclosed if the HTTPS flow is intercepted, especially since it's over a public network implied by the internet boundary.	Ensure TLS 1.3 with forward secrecy and monitor for MITM attacks.
	Denial of Service on Login Flow	Denial of Service	Medium	Open		The flow could be flooded with requests, denying service to legitimate logins across the boundary.	Implement DDoS protection and rate limiting on the endpoint.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Payment Page Access	Spoofing	High	Open		The flow to the payment page from external Customer to Customer Client could be spoofed, leading to fake payment initiations.	Require authenticated sessions and CSRF tokens.
	Tampering with Payment Intent	Tampering	High	Open		Payment details in the flow could be altered as it ingress to the client inside the boundary.	Validate all payment data server-side and use HMAC for integrity.
	Information Disclosure of Payment Data	Information Disclosure	High	Open		Sensitive payment information could leak if the HTTPS flow is compromised crossing the public boundary.	Use encrypted payloads within HTTPS and avoid logging sensitive data.
	Denial of Service on Payment Flow	Denial of Service	Medium	Open		Repeated bogus requests could overwhelm the payment flow endpoint.	Deploy WAF and rate limiting specifically for payment endpoints.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Card Details Submission	Spoofing	High	Open		The flow providing card details from external Customer to Client could be spoofed, submitting fraudulent data across boundaries.	Implement device fingerprinting and behavioral analysis.
	Tampering with Card Details	Tampering	High	Open		Card information could be modified in transit without a specified protocol, increasing risk over public networks.	Mandate encryption and integrity checks for this flow.
	Information Disclosure of Card Details	Information Disclosure	High	Open		Card details could be exposed in this unencrypted flow crossing from external to internal boundary.	Enforce end-to-end encryption and tokenization of card data.
	Denial of Service on Finalization Flow	Denial of Service	Medium	Open		The finalization flow could be targeted for DoS, preventing payments.	Use queuing and load balancing for high-traffic flows.

(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
	Spoofing Order Intent Flow	Spoofing	High	Open		Bidirectional flow between Customer Client and Merchant Web Server crosses from Customer/Internet to Merchant/Web boundary, allowing spoofing of order intents.	Use mutual authentication and session tokens.
	Tampering with PaymentIntent	Tampering	High	Open		Order amounts or PaymentIntents could be tampered in this boundary-crossing flow without specified protocol.	Implement message signing and validation.
	Repudiation of Order Sent	Repudiation	Medium	Open		Parties could deny sending or receiving order intents in this bidirectional flow across trust boundaries.	Log all transactions with non-repudiable proofs like digital signatures.
	Information Disclosure in Order Flow	Information Disclosure	High	Open		Sensitive order data could leak as the flow crosses public-facing boundaries without encryption.	Encrypt the entire payload and use secure channels.
	Denial of Service on Intent Flow	Denial of Service	Medium	Open		The bidirectional flow could be disrupted, blocking order processing.	Implement redundancy and failover for API calls.
	Elevation via Forged Intent	Elevation of Privilege	High	Open		Attacker could elevate by forging intents in this crossing flow, potentially authorizing higher amounts.	Enforce strict authorization checks on received intents.

(9) Attempt payment (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Payment Attempt	Spoofing	High	Open		The attempt payment flow from Stripe API to Payment Service within Stripe/Web boundary could be spoofed internally if compromised.	Use internal mTLS for service-to-service communication.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with Payment Data	Tampering	High	Open		Payment details could be altered in this internal flow without encryption.	Apply data validation and checksums.
	Information Disclosure Internally	Information Disclosure	Medium	Open		Sensitive payment info could be disclosed if the internal flow is intercepted within the boundary.	Encrypt internal traffic despite being within boundary.
	Denial of Service on Payment Service	Denial of Service	Medium	Open		Overloading this flow could deny payment processing.	Auto-scale the payment service and rate limit internal calls.

(10) Payment Response (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Payment Response	Spoofing	High	Open		The payment response flow from Payment Service to API service within Stripe/Web could be spoofed to fake successes.	Secure internal communications with authentication.
	Tampering with Response	Tampering	High	Open		Response status could be tampered, leading to incorrect processing.	Use signed responses and verify signatures.
	Repudiation of Response	Repudiation	Medium	Open		The service could deny sending a response without logging.	Implement comprehensive logging with audits.
	Information Disclosure in Response	Information Disclosure	Medium	Open		Response data could leak internally if not protected.	Minimize sensitive data in responses and encrypt if necessary.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing PaymentIntent Return	Spoofing	High	Open		The return flow from Stripe API to Customer Client crosses from Stripe/Web to Customer/Internet boundary, allowing spoofing of status.	Validate source with certificates and tokens.
	Tampering with Status	Tampering	High	Open		Payment status could be altered in this egress flow over public networks.	Sign the status messages cryptographically.
	Information Disclosure of Status	Information Disclosure	Medium	Open		Status info could be disclosed if not encrypted crossing boundaries.	Encrypt the response payload.
	Denial of Service on Return Flow	Denial of Service	Medium	Open		Blocking this flow could prevent clients from receiving confirmations.	Use multiple endpoints and retries.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing ConfirmCardPayment Call	Spoofing	High	Open		The confirmation flow from Customer Client to Stripe API crosses Customer/Internet to Stripe/Web boundaries over public networks.	Require API keys and authentication headers.
	Tampering with Confirmation Data	Tampering	High	Open		Confirmation details could be modified without protocol protection.	Use integrity protection like HMAC.
	Information Disclosure in Confirmation	Information Disclosure	High	Open		Card confirmation data could leak in this unencrypted public flow.	Tokenize and encrypt sensitive parts.
	Denial of Service on Confirmation	Denial of Service	Medium	Open		Flooding this ingress flow could disrupt payments.	API rate limiting and monitoring.
	Elevation via Malicious Confirmation	Elevation of Privilege	High	Open		Exploiting this flow could allow unauthorized payment confirmations.	Strict RBAC and input validation.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing PaymentIntent Return to Merchant	Spoofing	High	Open		The return flow from Stripe API to Merchant Web Server crosses Stripe/Web to Merchant/Web boundaries.	Use webhook signatures for verification.
	Tampering with Intent Data	Tampering	High	Open		Intent data could be altered in this crossing flow.	Validate data integrity on receipt.
	Information Disclosure to Merchant	Information Disclosure	Medium	Open		Partial payment info could leak if not secured.	Limit exposed data and encrypt.

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Order Information Send	Spoofing	High	Open		The order send flow from Merchant to Stripe API crosses Merchant/Web to Stripe/Web boundaries over implied public network.	Authenticate API calls with secrets.
	Tampering with Order Details	Tampering	High	Open		Amount or currency could be changed in transit.	Sign the request payloads.
	Information Disclosure of Order	Information Disclosure	Medium	Open		Order info could be intercepted without encryption.	Use HTTPS for all API communications.
	Denial of Service on Order Flow	Denial of Service	Medium	Open		Disrupting this flow could halt order processing.	Retry mechanisms and circuit breakers.

Merchant Web Server (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Merchant Web Server	Spoofing	High	Open		The Merchant Web Server inside Merchant/Web boundary receives ingress flows from Customer Client across boundaries, vulnerable to spoofing.	Implement IP whitelisting and authentication.
	Tampering with Server Data	Tampering	High	Open		Data processed by the server could be tampered via incoming flows.	Input sanitization and validation.
	Repudiation of Merchant Actions	Repudiation	Medium	Open		Merchant could deny processing orders without proper logging.	Audit logging of all server actions.
	Information Disclosure from Server	Information Disclosure	High	Open		Server could leak data via egress flows to Stripe.	Data minimization and encryption.
	Denial of Service on Web Server	Denial of Service	High	Open		Ingress flows could overwhelm the server.	Load balancers and auto-scaling.
	Elevation of Privilege on Server	Elevation of Privilege	High	Open		Vulnerabilities could allow privilege escalation via boundary-crossing inputs.	Regular patching and least privilege.

Stripe API service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Stripe API Service	Spoofing	High	Open		The Stripe API service inside Stripe/Web receives ingress from Merchant and Customer Client across boundaries, prone to spoofing.	API authentication and rate limiting.
	Tampering with API Data	Tampering	High	Open		API requests could be tampered, affecting payment processing.	Request validation and signing.
	Repudiation in API Calls	Repudiation	Medium	Open		Callers could deny requests without non-repudiation.	Signed requests and logging.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure via API	Information Disclosure	High	Open		API could expose sensitive data in responses crossing boundaries.	API gateway with data filtering.
	Denial of Service on API	Denial of Service	High	Open		High volume of ingress flows could cause DoS.	DDoS protection and caching.
	Elevation of Privilege in API	Elevation of Privilege	High	Open		Exploits could elevate access through API endpoints.	RBAC and input validation.

Stripe Payment Service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing Payment Service	Spoofing	High	Open		The Payment Service inside Stripe/Web receives internal flows from API service, but could be spoofed if boundary is breached.	Internal authentication mechanisms.
	Tampering with Payment Processing	Tampering	High	Open		Payment data could be tampered internally.	Data integrity checks.
	Repudiation of Payments	Repudiation	Medium	Open		Payments could be denied without proper records.	Immutable logging.
	Information Disclosure in Processing	Information Disclosure	High	Open		Sensitive card data could leak during processing.	Tokenization and secure enclaves.
	Denial of Service on Payment Service	Denial of Service	High	Open		Overload from internal flows could deny service.	Resource monitoring and scaling.
	Elevation of Privilege in Service	Elevation of Privilege	High	Open		Vulnerabilities could allow unauthorized access to payment functions.	Security hardening and audits.