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

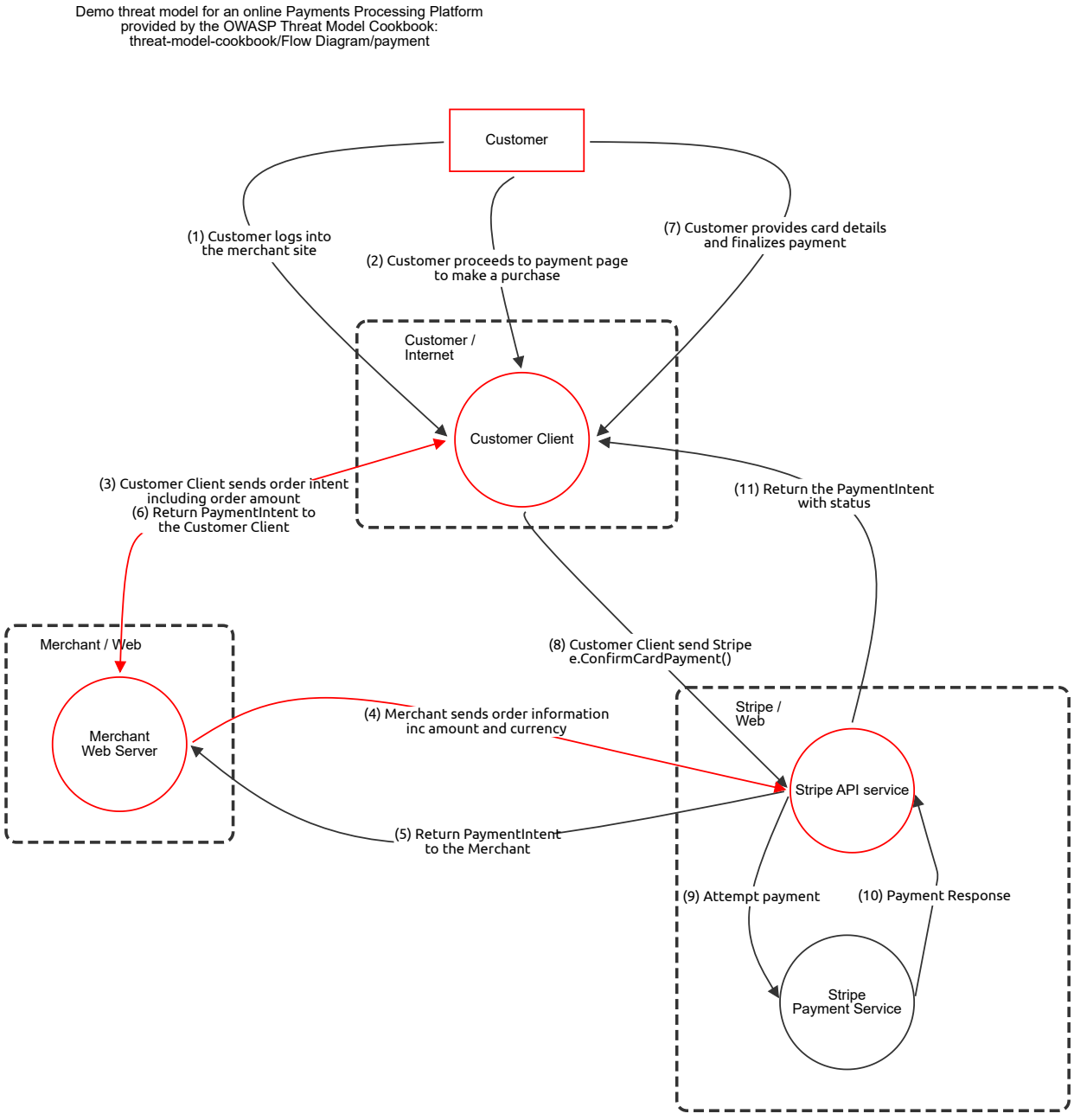
High level system description

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

Summary

Total Threats	9
Total Mitigated	0
Total Open	9
Open / Critical Severity	0
Open / High Severity	6
Open / Medium Severity	3
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
	Customer Account Spoofing and Takeover	Spoofing	High	Open		A malicious actor could impersonate a legitimate 'Customer' by stealing credentials through phishing or credential stuffing. This would allow them to make fraudulent purchases or access sensitive user data, originating from the untrusted 'Customer / Internet' zone.	Implement multi-factor authentication (MFA) for customer accounts. Enforce strong password policies, monitor for suspicious login attempts, and educate users about phishing risks.
	Transaction Repudiation	Repudiation	Medium	Open		A legitimate 'Customer' could falsely deny making a purchase (chargeback fraud), claiming their account was compromised or they never authorized the transaction.	Maintain detailed transaction logs, including customer IP address, device fingerprint, and a clear audit trail of the checkout process. Use services like 3D Secure for cardholder authentication to shift liability.

Customer Client (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Client-Side Payment Data Tampering	Tampering	High	Open		A malicious user could manipulate data within the 'Customer Client' (e.g., using browser developer tools) to change the price or quantity of items in the order intent before it is sent to the Merchant Web Server. This exploits the trust placed in the client application residing in the 'Customer / Internet' zone.	The Merchant Web Server must perform strict, server-side validation of all order details. The final price should always be calculated on the server based on product IDs, ignoring any price data submitted from the client.
	Sensitive Data Leakage from Client	Information Disclosure	High	Open		Malware (keyloggers, screen scrapers) or a Cross-Site Scripting (XSS) vulnerability on the merchant page could capture sensitive data entered into the 'Customer Client', such as credentials or payment card information.	Use a payment provider's client-side library (e.g., Stripe Elements) which renders sensitive payment fields in a secure iframe, isolating them from the host page. Implement a strong Content Security Policy (CSP) to prevent XSS.

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

Description: OAuth

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

(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
	Eavesdropping on Order Intent Flow	Information Disclosure	Medium	Open		An attacker could perform a Man-in-the-Middle (MitM) attack on the flow between the 'Customer Client' and 'Merchant Web Server' to intercept order details. This flow crosses from the untrusted 'Customer / Internet' zone to the 'Merchant / Web' zone.	Enforce the use of strong, up-to-date TLS (e.g., TLS 1.2 or 1.3) for all communication between the client and the server. Implement HTTP Strict Transport Security (HSTS).

(9) Attempt payment (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations

(10) Payment Response (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Compromise and Misuse of API Credentials	Spoofing	High	Open		If an attacker compromises the 'Merchant Web Server', they could steal the API keys used to authenticate with the 'Stripe API service'. This would allow them to spoof the merchant and perform unauthorized actions like issuing fraudulent refunds or creating fake transactions.	Store API keys securely in a dedicated secrets management system, not in code or configuration files. Use API keys with the principle of least privilege. Rotate keys regularly and monitor API logs for anomalous activity.

Merchant Web Server (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Server-Side Request Forgery (SSRF)	Spoofing	High	Open		If the 'Merchant Web Server' makes requests to other internal or external systems based on user-supplied data, an attacker could potentially trick it into making malicious requests on their behalf, effectively spoofing the server's identity to other services.	Implement a strict allowlist of permissible domains and IP addresses for outgoing requests. Validate and sanitize all user-supplied URLs and hostnames. Run the web server with minimal network permissions.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service Against Merchant Server	Denial of Service	High	Open		The 'Merchant Web Server' is a key chokepoint and is exposed to the internet. It can be targeted with application-layer DoS attacks (e.g., sending numerous incomplete order intents) or network-layer attacks, making the entire payment process unavailable.	Use a Web Application Firewall (WAF) and a DDoS mitigation service. Implement strict rate limiting on incoming requests from the 'Customer / Internet' zone.

Stripe API service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Abuse of Payment API Logic	Elevation of Privilege	Medium	Open		An attacker could attempt to find and exploit business logic flaws in how the 'Stripe API service' is used, such as replaying payment confirmation requests or exploiting race conditions between payment confirmation and order fulfillment to receive goods without a completed payment.	The merchant server must use idempotency keys for all payment-creating requests to Stripe to prevent replay attacks. The order fulfillment logic must rely on asynchronous webhooks from Stripe as the definitive source of truth for payment success, not the synchronous API response.

Stripe Payment Service (Process)

Number	Title	Type	Severity	Status	Score	Description	Mitigations