Zahlungen Architektur

Revisions- und Überprüfungshistorie

Revisionshistorie

Datum	Revision	Autor	Änderung
2021-03-29	1	Jeevan Singh	Erstellt

Überprüfungshistorie

Erforderliche Prüfer	Geprüft am	Prüfergebnis
Lord Voldemort	2021-03-29	Sieht toll aus, ich finde es toll, wie „stark“ die Sicherheit bei diesem Projekt ist!
Dr. Evil	2021-03-29	Ändere nichts daran
Hans Gruber	2021-03-29	Es ist genau so aufgebaut, wie ich es mir erhofft hatte
Freddy Krueger	2021-03-29	Ich hätte mir kein besseres System erträumen können

Kontext und Hintergrund

Um unseren kleineren Kunden einen besseren Service zu bieten, können sie per Kreditkarte bezahlen. Die Kreditkarteninformationen werden in unseren Systemen gespeichert, sodass sie in den entsprechenden Intervallen (monatlich/jährlich) abgerechnet werden können. Der Workspace-Besitzer gibt seine Kreditkarteninformationen über die Benutzeroberfläche ein und die Benutzeroberfläche gibt die Informationen an die API weiter, die ihren Weg in die Datenbank findet. Wir konzentrieren uns nur auf den API-Teil, ein anderes Team arbeitet an der Benutzeroberfläche.

Bevor wir die gespeicherte Kreditkarte aktualisieren, möchten wir sicherstellen, dass die Kreditkarte gültig ist. Das System wird einen API-Aufruf an VISA senden und überprüfen, ob die Karte authentisch und verwendbar ist.

HINWEIS: Das Unternehmen wird die Kreditkarteninformationen zu Abrechnungszwecken speichern, da wir nicht möchten, dass ein Drittanbieter die Kreditkarten belastet, weil dieser eine Gebühr von 5 % erhebt. 5 % scheinen nicht allzu viel zu sein, aber es summiert sich schnell

HINWEIS: Der Anruf beim Endpunkt von VISA ist für die ersten 1.000 API-Aufrufe an einem Tag kostenlos, danach werden 5,00 $ für jeden Satz von 1.000 Aufrufen berechnet.

Anforderungen

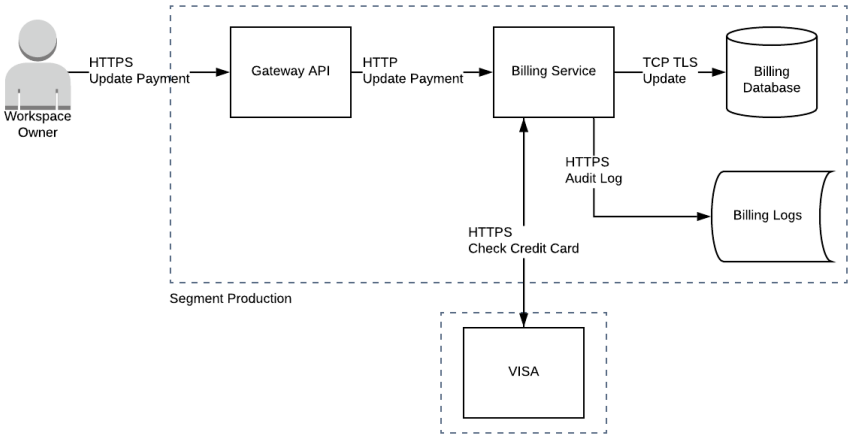
Dieser Prozess muss so reibungslos wie möglich ablaufen, da wir sicherstellen wollen, dass unsere Kunden ihre Kreditkarteninformationen eingeben. Wir wollen nicht, dass sie sich die Eingabe von Zahlungsinformationen noch einmal überlegen.

Wir speichern ihre Kreditkarteninformationen für zukünftige Transaktionen, weil wir möchten, dass es so reibungslos wie möglich abläuft. Wenn sie monatlich oder jährlich zahlen möchten, müssen wir sicherstellen, dass ihnen die Zahlung jedes Mal belastet wird, wenn der Zahlungszeitraum abgelaufen ist, und sie sich nicht jedes Mal darum kümmern müssen, ihre Kreditkarten herauszuholen.

HINWEIS: Der Umfang dieser speziellen Arbeit besteht lediglich darin, die Kreditkarteninformationen zu speichern. Die tatsächliche Belastung der Kreditkarte ist eine zukünftige Aufgabe.

Architektur

Die Architektur des Systems ist sehr einfach und unkompliziert. Es gibt fünf verschiedene Komponenten.



Diagramm

1 - Gateway-API

Die Gateway-API hat zwei Aufgabenbereiche:

- Durchsetzung der Ratenbegrenzung von 100 Anrufen/Minute pro Workspace-Besitzer
- Weiterleitung der entsprechenden Anfragen an den Abrechnungsdienst

Wenn die Gateway-API mehr als 100 Aufrufe/Minute/Benutzer erkennt, sendet sie dem Benutzer eine 429 (Too Many Requests).

HINWEIS: Die Gateway-API dient keinem anderen Zweck. Sie ist nicht dafür verantwortlich, das Token zu authentifizieren* oder *zu überprüfen, ob der Benutzer berechtigt ist, die Zahlungsinformationen zu aktualisieren.

2- Abrechnungsdienst

Der Abrechnungsdienst ist ein relativ einfacher Dienst und hat einige wenige Aufgaben:

- **Authentifizierung des Tokens** - Der Dienst bestätigt, dass der Benutzer derjenige ist, für den er sich ausgibt, und dass er zu diesem Mandanten gehört
- **Validierung der Informationen** - Die Eingabe bestätigt, dass der Name der Organisation, die Kreditkartennummer usw. in dem von uns erwarteten Format vorliegen. Dies dient der Validierung, hilft uns aber auch bei der Sicherheit. Es ist für einen Angreifer unmöglich, einen XSS- oder SQL-Injection-Angriff durchzuführen.
- **Validierung der Kreditkarte** - Das Format ist korrekt, aber wir müssen sicherstellen, dass die Kreditkarte tatsächlich gültig ist. Wir werden uns bei VISA erkundigen und bestätigen lassen, dass wir die Karte belasten können.
- **Protokollinformationen senden** - Um sicherzustellen, dass wir den Zustand des Dienstes kennen, werden alle Anwendungsprotokolle an die Abrechnungsprotokolle gesendet. Dies kann vom Team zu einem späteren Zeitpunkt überprüft werden, um festzustellen, ob es Probleme gab, und um Fehler zu beheben.
- **Informationen speichern** - Die Informationen werden später benötigt, um die Karte zu belasten. Speichern wir sie zur späteren Verwendung in der Datenbank

Antworten

- 200 - wenn wir die Karte erfolgreich mit VISA validieren und die Datenbank aktualisieren
- 400 - wenn unsere internen Validatoren festgestellt haben, dass es sich um eine fehlerhafte Eingabe handelt
- 400 - wenn die Kreditkarte nach der Überprüfung mit VISA ungültig war
- 503 - wenn die API von VISA aus irgendeinem Grund nicht funktioniert

3- Abrechnungsprotokolle

Die Abrechnungsprotokolle sind äußerst wichtig, um den Zustand des Systems zu verstehen. Alle Anwendungsprotokolle werden an den persistenten Speicher der Abrechnungsprotokolle gesendet, bei dem es sich lediglich um einen S3-Bucket handelt.

In den Abrechnungsprotokollen finden Sie möglicherweise Folgendes:

- Wer hat die Anrufe initiiert, wann wurden sie getätigt und was war das Ergebnis (200, 400, 503)
- Alle unerwarteten Fehler, die zur Fehlerbehebung aufgetreten sind, und die relevanten Informationen zur Fehlerbehebung

HINWEIS: In den Abrechnungsprotokollen finden Sie keine kreditkartenbezogenen Informationen und es werden zusätzliche Prüfungen durchgeführt, um sicherzustellen, dass keine Kreditkarteninformationen in die Protokolle gelangen

4- Abrechnungsdatenbank

Dies ist eine sehr einfache Datenbank, die lediglich die Informationen aus dem Rechnungsdienst sammelt.

Gespeicherte Kreditkarteninformationen:

- Mandanten-ID
- Name der Person oder Organisation, auf die die Karte ausgestellt ist
- Kreditkartennummer
- Ablaufdatum
- CVV

Die Rechnungsdatenbank enthält sensible Informationen, weshalb zusätzliche Sicherheitsvorkehrungen getroffen wurden:

Das Laufwerk, auf dem sich die Datenbank befindet, ist im Ruhezustand verschlüsselt, d. h., wenn ein AWS-Mitarbeiter die Festplatte stehlen würde, könnte er die Inhalte nicht lesen Der Zugriff auf die Datenbank ist auf die Rolle „Prod Admin“ im Access Service beschränkt, d. h., Sie benötigen die Genehmigung einer Person, um darauf zugreifen zu können

5- VISA API

Wir haben ein Jahresabonnement für den Validierungsendpunkt von VISA und verwenden diesen Endpunkt, um zu überprüfen, ob die Kreditkarteninformationen korrekt sind.

Wie oben erwähnt, sind die ersten tausend Anrufe pro Tag kostenlos und danach fallen für jeden der tausend Anrufe Kosten in Höhe von 5 $ an.

Wir führen eine Eingabevalidierung der Kreditkarten durch, um sicherzustellen, dass wir keine Werte an VISA senden, von denen wir wissen, dass sie ungültig sind. Wir möchten sicherstellen, dass wir unter der Grenze von 1.000 bleiben.

Sicherheitskontrollen

Es gibt einige interessante Sicherheitskontrollen für das Zahlungssystem

Authentifizierung

- Der UpdatePayment-API-Aufruf erfordert ein vom System generiertes API-Token
- Wir haben ein Token für die Verbindung mit VISA, das in einem geeigneten Geheimspeicher gespeichert ist

Denial of Service

– Maximal 100 API-Aufrufe pro Minute und Benutzer zum Schutz vor Denial-of-Service-Angriffen von einem Client

Fast durchgängiges TLS

– Das gesamte System verfügt über TLS, außer wenn die Gateway-API die Anfrage an den Abrechnungsdienst weiterleitet

Manipulation

– Es gibt eine strenge Eingabevalidierung für Kreditkartenwerte, die zur Verhinderung von SQL-Injection- und Stored-XSS-Angriffen eingesetzt wird – Wir stellen sicher, dass die folgenden Felder keine unerwarteten Eingaben enthalten

– Mandanten-ID – Name der Organisation – Name des Karteninhabers – Kreditkartennummer – Ablaufdatum – CVV

Beispiel-Dokumentation

Ziel

- Finden Sie so viele Bedrohungen wie möglich, aber melden Sie Ihre drei größten Risiken zurück

Wie?

- Welche Vorteile bietet diese Funktion?
- Was möchten Sie schützen?
- Gibt es Bereiche des Systems, auf die ein böswilliger Akteur zugreifen oder die er ins Visier nehmen möchte?
- Was ist für sie wichtig? Wie können sie daraus einen Vorteil ziehen?
- Sehen Sie sich das Diagramm an und nutzen Sie STRIDE, um Bedrohungen/Bedenken/Risiken zu ermitteln

Diagramm

Das Diagramm enthält Zahlen, die verschiedene Bereiche darstellen, in denen Sie nach Bedrohungen suchen können.

1. Gibt es Bedenken zwischen einem Workspace-Besitzer und der Gateway-API?
2. Gibt es Bedrohungen zwischen der Gateway-API und dem Abrechnungsdienst?

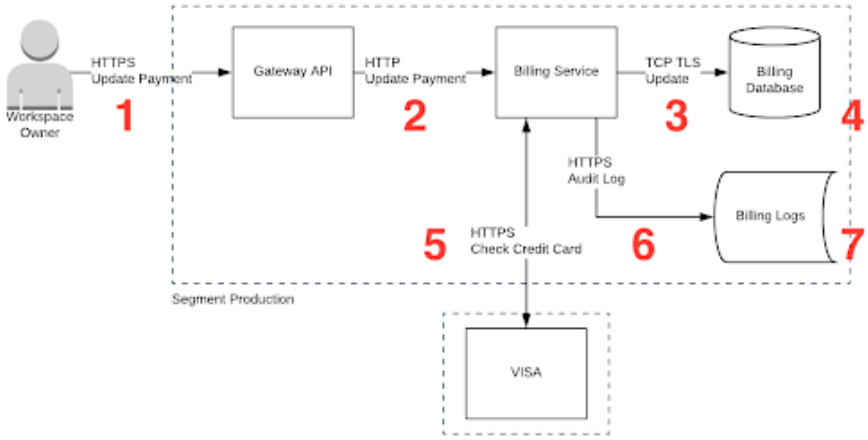


Figure 1: Payments Architecture

3. Haben Sie Bedenken zwischen dem Abrechnungsdienst und der Datenbank?
4. Welche Bedenken bestehen hinsichtlich der Abrechnungsdatenbank?
5. Gibt es Bedenken zwischen dem Abrechnungsdienst und VISA?
6. Gibt es Bedrohungen zwischen dem Abrechnungsdienst und den Abrechnungsprotokollen?
7. Haben Sie Bedenken hinsichtlich der Abrechnungsprotokolle?

HINWEIS: Es wird nicht in allen Bereichen Bedrohungen geben, aber es ist wichtig, dass Sie sich beim Bedrohungsmodell die verschiedenen Standorte ansehen, um zu sehen, ob Sie Bedenken haben

Annahmen

Zur Erinnerung: Diese Vorlage ist in einigen Bereichen absichtlich vage gehalten und Sie müssen Annahmen treffen.

Welche Annahmen haben Sie getroffen?

- VISA bedeutet jeder Zahlungsabwickler, nicht nur VISA
- *Fügen Sie hier Ihre eigenen Annahmen hinzu*

Vermögenswerte

- Vermögenswert Nr. 1
- Vermögenswert Nr. 2
- Vermögenswert usw.

Bedrohungen

Sicherheitskontrollen wurden unten grün hervorgehoben. Sie können Ihre Bedrohungen gerne in die folgende Liste aufnehmen und rot markieren.

Spoofing

- *Der Abrechnungsdienst überprüft, ob das Token gültig ist*
- *Der Abrechnungsdienst verwendet ein in einem Geheimspeicher gespeichertes Token, um eine Verbindung mit VISA herzustellen*
- Beispiel für eine mögliche negative Auswirkung

Manipulation – Der Abrechnungsdienst verfügt über sehr strenge Eingabevalidierungskontrollen, es wäre nahezu unmöglich, die einzelnen Felder (Kreditkartennummer, Ablaufdatum usw.) zu manipulieren, um eine gespeicherte XSS- oder SQL-Injection-Schwachstelle zu erhalten. – Beispiel für eine mögliche negative Auswirkung

Ablehnung – Wir protokollieren in den Abrechnungsprotokollen, wer die Zahlungsinformationen wann aktualisiert hat. – Beispiel für eine mögliche negative Auswirkung

Offenlegung von Informationen

- Die Festplatte der Abrechnungsdatenbank ist im Ruhezustand verschlüsselt, um zu verhindern, dass AWS-Mitarbeiter Festplatten stehlen
- Beispiel für etwas, das passieren könnte

Denial of Service

- Die Gateway-API verfügt über einen Ratenbegrenzer, der sicherstellt, dass ein Workspace-Besitzer 100/min nicht überschreiten kann
- Beispiel für etwas, das passieren könnte

Erhöhung von Berechtigungen

- Beispiel für etwas, das passieren könnte

Weitere Sicherheitsfragen/Gedanken?

- Dies ist eine schlechte Sache, die passieren könnte, aber sie fällt nicht wirklich unter STRIDE

Top 3 Risiken

- Risiko Nr. 1
- Risiko Nr. 2
- Risiko Nr. 3