netidee PROJEKTE

Opaque

Endbericht | Call 17 | Projekt ID 6374

Inhalt

1 Einleitung

Im Rahmen der Entwicklung der Software Applikation Serenity sind wir auf das OPAQUE-Protokoll gestoßen. Dieses Protokoll ermöglicht Username/Password-basierte Authentifizierung, ohne dass jemals das Password in seiner originalen Form an den Server geschickt wird und besitzt dabei noch weitere wichtige Security-Eigenschaften. Indem das Password sowohl bei der Registrierung als auch dem Login nie an den Server geschickt wird, können Passwörter nicht unbeabsichtigt durch Logs oder sogar beabsichtigt kompromittiert werden.

Eine weitere wichtige Eigenschaft war, dass die Authentifizierung mittels des Protokolls zu einem stabilen „Export-Key“ führt, welcher ausschließlich am Client verfügbar ist und sich daher anbietet als Basis für die Entwicklung von Software mit Ende-zu-Ende-Verschlüsselung.

In weiteren Nachforschungen haben wir herausgefunden, dass Meta eine Implementierung in Rust open sourced hat und diese für die Ende-zu-Ende-Verschlüsselung von WhatsApp-Backups verwendet. Ohne größeren Aufwand und Vorwissen in Rust war es jedoch nicht möglich die Implementierung direkt im Browser, Mobile Apps oder in Node Backends einzusetzen.

2 Projektbeschreibung

Ziel unseres Projektes war eine produktionsreife Implementierung für alle gängigen JavaScript Umgebungen (Node, Browser & ReactNative) zu entwickeln. Es sollte möglich sein, ohne kryptographisches Vorwissen in wenigen Zeilen Code das Package einzubinden. Out-of-the-Box TypeScript Support war uns ebenfalls ein anliegen.

Des Weiteren war es uns ein Anliegen eine hervorragende Dokumentation zu erstellen, die die Anwendung den EntwicklerInnen schnell und einfach vermittelt, als auch das OPAQUE-Protokoll einfacher vermittelt als der geplante RFC der IETF.

Damit ist dieses Projekt sinnvoll für alle Unternehmen, die passwortbasierte Authentifizierung anbieten. Sei es vom Ein-Personen-Unternehmen, welches ein Admin-Interface für die erstellte Restaurant-Webseite anbietet, als auch Unternehmen mit mehreren hundert MitarbeiterInnen die Finanzsoftware erstellen.

Eine weitere essentielle Fähigkeit des OPAQUE-Protokolls ist, dass es nach jeder Authentifizierung einen “Export-Key” ausschließlich am Client ausgibt. Dieser ist pro Benutzer immer identisch und lässt sich daher ideal verwenden, um Ende-zu-Ende-Verschlüsselte Applikationen zu entwickeln und dabei die gewohnte Benutzername/Passwort User Experience anzubieten.

Das Protokoll verwendet moderne Kryptographie z.B. Oblivious Pseudorandom Functions (OPRFs). Da die Folgen von kleinen Fehlern hier drastische Folgen haben können, war es besonders wichtig, potenzielle Fehlermöglichkeiten so gering wie möglich zu halten. Dies hat dazu geführt, dass wir uns entschlossen haben, das bestehende opaque-ke Rust Package

aufzubauen. Dieses wurde bereits bei einem Security Audit durch die NCC Group überprüft. Des Weiteren haben wir entschieden, das Projekt auf mehrere Package aufzusplitten, um die Abhängigkeiten isolierbarer zu machen und damit auch die potentiellen Angriffsvektoren durch inkludierte Software Package zu reduzieren.

Die Firma 7ASecurity hat ein Security Audit durchgeführt und es konnten in den Software Paketen keine Fehler gefunden werden. In den Beispielprojekten, die Teil der Dokumentation sind, wurden potentielle Angriffsvektoren entdeckt. Alle wurden korrigiert und durch 7ASecurity verifiziert.

Das Team hinter dem Opaque Projekt ist das Kernteam der Naisho Gmbh und besteht aus Susanne Kristufek, Bettina Ecker (Freelancerin), Stefan Oestreicher und Nikolaus Graf.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - *Detailplanung und Formales am Projektstart*

Das Projekt hat etwas später gestartet als im Antrag ursprünglich geplant, aber sobald begonnen wurde, konnte die Detailprojektion, Liste der Projektergebnisse, Projektwebseite, erster Blogeintrag und Ansuchen zur ersten Förderrate recht flott umgesetzt werden.

3.2 Arbeitspaket 2 - *Beta (Web Client & Node)*

Zu Beginn wurde das Rust Projekt inkl. Web Assembly Build aufgesetzt und mit der Implementierung des Registration und Login Vorgänge begonnen. Diese Arbeiten erfolgten um einiges schneller als erwartet.

Daraufhin wurde ein starker Fokus auf eine automatisierte Test-Suite gesetzt. Es gibt sowohl Unit als auch end-to-end Tests. Des Weiteren wurden mehrere Beispielprojekte angelegt, um die unterschiedlichen Umgebungen (verschiedene Web-Bundler als auch Node) zu testen. Dies war besonders wichtig, da wir hier auf unerwartete Probleme mit der Einbindung (wegen WebAssembly) des Software-Pakets gestoßen sind. Diese Probleme konnten in AP3 gut gelöst werden.

3.3 Arbeitspaket 3 - *Production ready (Web Client & Node)*

Dieses Arbeitspaket wurde großteils schon umgesetzt und es gibt eine produktionsreif Version des opaque Paketes in der Variation ristretto255 und P256. Wir haben beide angeboten, da beide

in der Spezifikation des Protokolls angegeben sind und Kevin Lewi von Meta empfohlen hat auch beide anzubieten.

Wie oben erwähnt, hat der Einsatz von WebAssembly zu verschiedenen Problemen in unterschiedlichen Umgebungen geführt. Wir konnten das Problem lösen, indem wir das Laden von WebAssembly code in das Software Paket eingebettet haben. Hierbei wird der Binary Code als base64 encoded und beim initialisieren des Moduls geladen.

Als Test haben wir das opaque Softwarepaket auch in eine Applikation eingebunden. Dies war wichtig, da uns dadurch ein paar unverständliche API Entscheidungen aufgefallen sind, welche wir mit einem API Redesign ausmerzen konnten.

Des Weiteren haben wir Tests durchgeführt, welcher Rust-WebAssembly Memory-Allocator sinnvoll ist und die Ergebnisse in der Dokumentation dokumentiert.

Während die gebauten Software Pakete in Richtung produktionreifen Zustand waren entwickelt wurden, wurde Kontakt zu diversen Firmen für eine Security Audit aufgenommen. Unser Budget betrug in etwa 4.000 EUR. Die Angebote von diversen Firmen reichten von 21.000 bis ca. 90.000 EUR und haben damit unser Budget deutlich gesprengt. Danach haben wir Kontakt zu Open Technology Fund aufgenommen und konnten diesen als Sponsor für das Security Audit gewinnen.

Das Audit wurde von 7ASecurity im Oktober und November durchgeführt und in den Software Paketen wurden keine Fehler gefunden. Verbesserungen im Dependency Management und Release Prozess wurden empfohlen und diese wurden auch umgesetzt.

3.4 Arbeitspaket 4 - *React Native Integration*

Ursprünglich war geplant, dass mit diesem Arbeitspaket erst Mitte Juni begonnen werden sollte. Der Hauptentwickler Stefan Oestreicher hat jedoch nach den frühen Erfolgen in Arbeitspaket 2 vorgeschlagen, dass wir den Start hier deutlich vorziehen, da er mit React Native nicht vertraut war. Dies hat sich als richtige und wichtige Entscheidung herausgestellt, da der Aufwand hier unterschätzt wurde.

Mittlerweile haben wir eine produktionsreife Version des react-native-opaque Paketes in der Variation ristretto255. Die Generierung der Variante P256 wurde gegen Ende des Projektes nachgeliefert.

Weiters gibt es auch eine Test-Suite die sowohl auf iOS, Android und der react-native-web Version läuft.

3.5 Arbeitspaket 5 - *Basic Documentation*

In diesem Arbeitspaket wurde die grundlegende Dokumentations-Website erstellt. Dies beinhaltet das automatisierte veröffentlichen nach jeder Änderung, die Struktur der Website, die Suche und das Design.

Teil dieses Arbeitspaketes war es auch das Logo und Branding für das Projekt zu designen. Bettina Ecker hat diese Arbeiten bereits abgeschlossen.

Ebenfalls wurden die ersten Inhalte und eine grobe Struktur angelegt.

3.6 Arbeitspaket 6 - *Polished Documentation*

Besonderen Wert haben wir auf die Landing Page gelegt. Hier geht es darum, das Projekt visuell ansprechend und einfach den BesucherInnen nahe zu bringen.

Besonders viele Ressourcen wurden in die Visualisierung einer Systemsimulation auf der Landing-Page investiert. Ziel dabei war es, das OPAQUE-Protokoll so einfach wie möglich zu erklären, um dem Landingpage Besucher Vertrauen zu geben, indem der ganze Ablauf nachvollziehbar gemacht wird. Dies hat einige Skizzen, visuelle Experimente und das Erstellen von Animationen erfordert. Einige dieser Artefakte konnten an anderen Stellen in der Dokumentation wiederverwendet werden.

Ein weiterer Schwerpunkt war das Erstellen von Beispielprojekten, die sich mittels einer Handvoll von Kommandos lokal erstellen lassen und somit als Basis für ein neues Projekt dienen können bzw. Als Inspiration für die Migration von existierenden Projekten.

An Inhalten wurden ein Getting Started Guide, eine Advanced Usage Sektion und diverse Seiten mit Hintergrundinformationen wie z.B. Bundler Configuration, Architecture Decision und Specification.

Wir haben auch die Produktion eines Video Kurses erwogen, jedoch entschieden wir uns vorrangig in die Verbesserung der Visualisierung und Beispiele zu investieren.

3.7 Arbeitspaket 7 - *Dokumentation und Formales am Projektende*

Die erforderlichen Aufzeichnungen wie z.B. Zeitaufzeichnungen werden durchgehend gemacht und alle 2 Monate haben wir einen Blog-Post veröffentlicht. Das Projektende wurde zwischenzeitlich für zwei Monate pausiert und das Projektende auf Februar 2024 gelegt. Daher wurde in dieser Zeit kein Blogpost erstellt.

4 Umsetzung Förderauflagen

Für dieses Projekt gab es keine besondern Förderauflagen.

5 Liste Projektendergebnisse

Kurzbeschreibung der erreichten Projektendergebnisse jeweils mit Open Source Lizenz und Webadresse (netidee Vorgaben beachten!)

1	*Projektzwischenbericht*	*CC BY- SA 4.0*	*netidee.at/opaque*
2	*Projektendbericht*	*CC BY- SA 4.0*	*netidee.at/opaque*
3	*Entwickler_innen-DOKUMENTATION des Projektergebnisses für andere Entwickler_innen ("Dritte"), die das Projektergebnis nach Projektende nutzen/weiterentwickeln wollen* *Für Entwickler_innen (Systemkonzept, ggf. Grobspezifikationen):* *a. WAS IST ES* *b. FÜR WEN IST ES /WEM HILFT ES WODURCH* *c. WIE FUNKTIONIERT ES (für Entwickler_innen: Übersicht und detailliertes Systemkonzept, SW-Struktur)*	*MIT*	*https://github.com/serenity-kit/opaque/blob/main/CONTRIBUTING.md* *https://github.com/serenity-kit/react-native-opaque/blob/main/CONTRIBUTING.md*
4	*Anwender_innen-DOKUMENTATION des Projektergebnisses für Anwender_innen, die das Projektergebnis nach Projektende nutzen wollen* *Für Anwender_innen ("Bedienungsanleitung") :* *a. WAS IST ES* *b. FÜR WEN IST ES /WEM HILFT ES WODURCH*	*CC BY- SA 4.0*	*https://opaque-auth.com/*

	c. WIE FUNKTIONIERT ES		
5	*Veröffentlichungsfähiger Einseiter / Zusammenfassung* ** Kurzfassung WAS | FÜR WEN | WIE* ** Liste Projektergebnisse - also diese Liste, ggf. kompromiert* ** mit Angabe Open Source Lizenz/Webadresse* ** wo finden Dritte die Projektergebnisse (inkl. Dokumentation Anwender_innen bzw. Entwickler_innen)* ** mögliche Weiterentwicklungen/ weitere Einsatz-/ Nutzungsmöglichkeiten*	*CC BY- SA 4.0*	*netidee.at/opaque*
6	*Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (als Teil des Endberichtes)* ** Welche Maßnahmen wurden in welchem Umfang gesetzt* ** Jeweils Bewertung Aufwand / Nutzen* ** Lessons Learned / Empfehlungen für andere Projekte*	*CC BY- SA 4.0*	*netidee.at/opaque*
7	*Opaque Package veröffentlicht in eine Github repository und publiziert auf NPM* ** Dieses Package erlaubt die Software in Websites einzubetten als auch in Node Server backend* ** Das Package ist mit Typescript typisiert und enthält eine solid Testsuite* ** Das Package enthält ein CLI Command um einen Server private key zu generieren"*	*MIT*	*https://github.com/ serenity-kit/opaque*
8	*Opaque Package für React Native veröffentlicht in eine Github repository und publiziert auf NPM. Die p256 Variante wurde als separates Repository angelegt.*	*MIT*	*https://github.com/ serenity-kit/react-native-opaque* *https://github.com/*

	** Dieses Package erlaubt die Software in React Native clients einzubetten und unterstützt iOS und Android* ** Das Package ist mit Typescript typisiert und enthält eine solid Testsuite*		*serenity-kit/react-native-opaque-p256*
9	*Dieses Package enthält das Script um eines der Opaque Beispiele lokal zu erstellen.*	*MIT*	*https://github.com/serenity-kit/create-opaque*

6 Verwertung der Projektergebnisse in der Praxis

Das Projekt wird bereits in der Software Applikation Serenity verwendet, aber vor der Fertigstellung des Security Audit Berichtes gab es unsererseits nur wenige Aktivitäten, um das Projekt bekannt zu machen.

Es hat doch schon Mitte Feb 2024 23 Github Sterne gesammelt und hat laut öffentlichen NPM Statistiken 470 wöchentliche Downloads und vermutlich von einer Handvoll von Projekten verwendet, die uns nicht bekannt sind.

Ein Entwickler einer Passwort-Manager Firma hat unser Projekt auch gefunden und uns kontaktiert. Sie evaluieren das Projekt und warten gespannt auf die Veröffentlichung des Security Audit Berichtes.

7 Öffentlichkeitsarbeit/ Vernetzung

Vor Beginn des Projektes haben wir mit Kevin Lewi, einer der Autoren der OPAQUE Protokoll Spezifikation und auch Hauptverantwortlicher für opaque-ke bei Meta, Kontakt aufgenommen.

Herr Lewi hat uns während der Entwicklung immer wieder Feedback gegeben und Unklarheiten in der Spezifikation aufgelöst. Er heißt unser Projekt gut, es wird mittlerweile auch schon direkt im Readme von opaque-ke verlinkt (https://github.com/facebook/opaque-ke#resources). Ein vergleichbares, aber bei weitem nicht so ausgearbeitetes Projekt wird nicht mehr gepflegt und der Author hat angeboten, dass sein Projekt unseres empfiehlt (https://github.com/marucjmar/opaque-wasm/issues/12).

Unser Opaque Projekt wird als Teil eines Vortrags von Nikolaus Graf auf der ReactSummit Konferenz in Amsterdam am 14. Juni 2024 vor mehreren hundert EntwicklerInnen vorgestellt. Dieser Vortrag wird dann auch auf diversen Meetups (ViennaJS, SBA Security Meetup, Rust Meetup) und anderen Konferenzen eingereicht.

Des Weiteren soll ein kurzer Videokurs erstellt werden. Diese kann dann auf diversen Plattformen wie Youtube oder Egghead veröffentlicht werden.

8 Eigene Projektwebsite

https://opaque-documentation.netlify.app/

9 Geplante Aktivitäten nach netidee-Projektende

Ein wichtiger Teil wird die Bekanntmachung und Verbreitung der Open Source Library.

Die nächste konkrete Aktivität ist ein Opaque als Teil eines Vortrags auf der ReactSummit Konferenz in Amsterdam am 14. Juni 2024. Derselbe Vortrag wird dieses Jahr auf diversen Meetups und Konferenzen einreichen.

Außerdem planen wir, sowohl die Open Source Library als auch die Dokumentation weiterhin zu pflegen. Dies ist auch eine Notwendigkeit, da wir die Library bereits für die Authentifikation in unserem Hauptprodukt Serenity verwenden.

10 Anregungen für Weiterentwicklungen durch Dritte

Unsere Lösung ist nun einfach in JavaScript-Systeme integrierbar. Dies erlaubt es Unternehmen, welche JavaScript (Node) oder Rust im Backend einsetzen, das Package zu integrieren und somit eine sichere Authentifizierung zu ermöglichen.

Potentielle Weiterentwicklungsmöglichkeiten sehen wir vor allem darin Opaque auch für diverse andere Programmiersprachen e.g. Python, Go zu portieren. Des weitern wäre es spannend, noch einfachere und bessere Erklärung und Visualisierung der OPAQUE-Protokolls in der Dokumentation.