TH Brandenburg
Studiengang IT Sicherheit
Fachbereich Informatik und Medien
Entwicklung Sicherer Softwaresysteme
Prof. Dr.-Ing. Martin Schafföner

Einsendeaufgabe: Threat Analysis

Sommersemester 2024
Abgabetermin 7. Mai 2024

Mara Schulke
Matrikel-Nr. 20215853

Zusammenfassung

In dieser Einsendeaufgabe wird die Online-Meeting-Plattform Google Meet der Firma Google hinsichtlich ihrer softwareseitigen Risiken und Gefahren analysiert. Es wird die Vorgehensweise der Gefahrenanalyse, die entdeckten Risiken und Gefahren zusammengefasst und eine Priorisierung mit Handlungsempfehlung daraus abgeleitet. Des weiteren wird auf die Methodik der Gefahrenanalyse eingegangen und von alternativen Vorgehensweisen abgegrenzt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Projektauswahl: Google Meet

Die Auswahl des zu analysierenden Projektes viel auf die Onlne-Meeting-Plattform Google Meet da diese mehrere Punkte abdeckt:

1. Ihre Funktionsweise- und -umfang ist ihm Rahmen einer Kurzanalyse greifbar
2. Die Anwendungsarchitektur lässt sich aus vergleichbaren Echtzeit-Meeting Anwendungen ableiten und vereinfachen
3. Sie hat einen weitreichenden Bekanntheitsgrad

Vereinfacht lässt sich Google Meet in einer Client-Server Architektur darstellen, auch wenn an dieser Stelle angemerkt werden muss, dass durch die Menge an Nutzern, die Vielzahl an Integrationen und die Stabilität der Software davon auszugehen ist, dass es sich nicht um eine naive Architektur der Server bzw. Server-Infrastruktur handelt.

Im Rahmen dieser Einsendeaufgabe wird die Annahme getroffen, dass Google Meet der folgenden Architektur entlang aufgebaut ist:

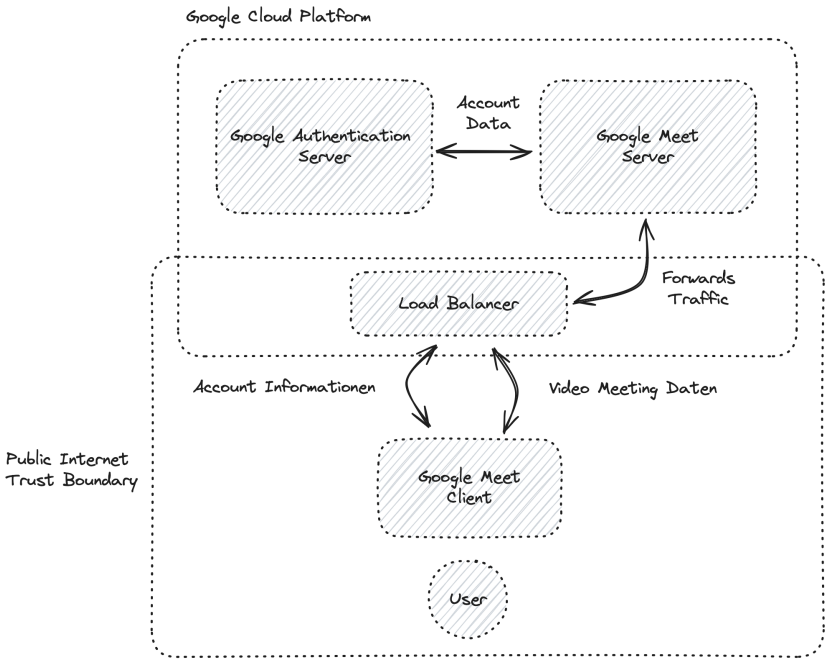

Abbildung 1: Anwendungsarchitektur Google Meet (stark vereinfacht)

Im Rahmen dieser Aufgabe wird von einer monolithischen Anwendungsarchitektur von Google Meet ausgegangen in einem zeitlich angemessenen Rahmen in der Lage zu sein eine hypothetische Gefahrenanalyse durchführen zu können. Der Google Meet Server in der beistehenden Grafik übernimmt die Verwaltung der Sitzungen, die Account-Authentifizierung und die Aushandlung der WebRTC-Sitzungen zwischen verschiedenen Nutzer.

2 Auswahl der Gefahrenanalyse Software

Um eine automatisierte Gefahrenanalyse durchzuführen stehen mittlerweile (Stand Mai 2024), eine Vielzahl an verschiedenen Tools zur Verfügung. Nachfolgend sind einige beliebtesten bzw. bekanntesten, kostenlosen *Threat-Modelling*-Programme aufgelistet:

Kostenlose Software

Da diese Einsendeaufgabe von geringem Umfang ist und keine weiteren Anwendungsfälle für eine ausgereifte Threat-Modelling Software absehbar sind beschränke ich den Vergleich verschiedener Software-Lösungen auf kostenlose bzw. Open-Source Programme.

MTMT – Microsoft Threat Modeling Tool

MTMT ist das Threat Modeling Tool der Firma Microsoft. Es ist ein weit verbreitetes Werkzeug zur Modellierung von Software und der Automatisierten Analyse dieser Modelle. Da der Anbieter Microsoft ist, besteht leider die Anforderung MTMT auf einem Windows-System auszuführen.

Das MTMT ist weiterhin ein sehr umfangreiches Werkzeug, das ebenfalls im professionellen Kontext Einsatz findet – dies resultiert in einer komplexen Oberfläche mit vielen Möglichkeiten und ausgereiften Konzepten.

Leider musste ich durch die Gegebenheit, kein Windows-System zu betreiben, die Analyse des MTMT hier beenden, da es mir leider nicht möglich war eine ausführbare Version für Unix-Systeme zu finden ohne Virtualisierung zu verwenden.

OWASP ThreatDragon

OWASP ThreatDragon ist eine Software der Organization OWASP (kurz für *Open Worldwide Application Security Project*). Die Software ist kostenlos und Browser basiert. Sie enthält Funktionen für die Diagramm-Zeichnung und anschließende Erstellung eines Sicherheits-Reports. Die Arbeitsstände müssen allerdings immer heruntergeladen werden und in lokalen Dateien gespeichert werden – dies ist eine relativ umständliche Arbeitsweise, da anders als bei nativer Software nicht eine Datei durchgehend geöffnet und bearbeitet werden kann. Außerdem bietet ThreatDragon nur einen begrenzten Umfang an erkannten Sicherheitslücken bei einer offensichtlich fehlerhaften Softwarearchitektur.

AWS ThreatComposer

AWS ThreatComposer ist eine Open Source Software der Firma Amazon. Sie ist ebenfalls Browser basiert, allerdings etwas ausgereifter und nutzerfreundlicher als die OWASP alternative. Dennoch gibt es auch hier einige Unannehmlichkeiten für Nutzer: Die Benutzeroberfläche ist unklar (e.g. Diagramme sind nur zur Dokumentation hinzuzufügen, nicht zur automatisierten Auswertung). Es gibt keine klare Datenverwaltung bzw. keinen klaren Datenexport und der allgemeine Funktionsumfang wirkt etwas eingeengt.

Threagile

Threagile ist kostenlose Open-Source Software zur Gefahrenanalyse ohne eine Benutzeroberfläche. Dies ist eine Eigenheit im Vergleich zu den anderen vorgestellten Programmen die hauptsächlich über das Desktop- oder Web-Interface zu bedienen sind. Threagile lässt sich über eine .yaml Datei bedienen die einem vorgegebenen Schema entsprechen muss. So lassen sich im Quelltext Komponenten (und ihre Beziehungen), Datenbanken, Verschlüsselungen, Datenflüsse, verwendete Technologien und vieles mehr Dokumentieren.

Der klare Vorteil von Threagile liegt in der Möglichkeit diese YAML-Datei bzw. die analysierte Architektur ohne weiteres in git oder anderen VCS (kurz für *Version Control Systems*) zu versionieren. Bei anderer proprietär Software ist dies zwar grundsätzlich Möglich aber da VCS in der Regel auf Klartext-Dateien ausgelegt sind sind die Möglichkeiten begrenzt bzw. parallel im Team an einer Datei zu arbeiten.

Des Weiteren ist es einfach mittels Threagile eine Gefahrenanalyse auf der YAML-Datei auszuführen und der generierte Report hat eine hohe Qualität (optisch, strukturell und inhaltlich).

2.1 Begründung der Auswahl

Im Rahmen dieser Einsendeaufgabe habe ich mich unter den oben dargestellten Abgrenzungen der verschiedenen Softwarelösungen für die Nutzung von Threagile entschieden. Threagile ist einfach zu bedienen, reproduzierbar (mittels Docker), eine nützliche Resource für zukünftige berufliche Situationen.

3 Ergebnis der automatisierten Gefahrenanalyse

Der automatisierte Report von Threagile hat basierend der oben dargestellten, stark vereinfachten, Anwendungsarchitektur folgende Risiken identifizieren können.

4 Detaillierte Gefahrenanalyse

Analyze some of the threats in more detail.

5 Priorisierung der Gefahren

6 Zusammenfassung