Richtlinie zum „Software-Sprint“

Bitmask VPN-Client für Android

Schlussbericht

Zuwendungsempfänger:

Richard Johannes

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS17S35** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Im Rahmen der Förderung stand die Weiterentwicklung und Stabilisierung eines VPN-Clients für das LEAP Softwaresystem im Fokus. Der bereits als Beta-Prototyp entwickelte Client namens Bitmask ermöglicht es, eine verschlüsselte Internetverbindung aufzubauen und Internetzensur zu umgehen, ohne spezielles technisches Wissen dafür zu benötigen. Bitmask trägt damit einen wesentlichen Teil zum Schutz der Privatsphäre seiner Nutzer*innen bei.

Primäres Ziele war es, die Software für den produktiven Masseneinsatz vorzubereiten.

Dafür sollten sicherheitsrelevante Verbesserungen vorgenommen werden, wie die Aktualisierung aller Bibliotheken (u.a. OpenVPN unnd OpenSSL), der Implementierung der Protokolle IPv6, UDP und obfs4.

Des weiteren sollten die Bedienbarkeit der Software vereinfacht werden. Dafür sollten im Wesentlichen die Oberfläche im Materlal Design überarbeitet und der Batterieverbrauch reduziert werden.

Um die Softwarequalität zu steigern, sollten Unit und Insturmentation Tests geschrieben werden sowie die Aufnahme der Tests in den Continuous Integration (CI) Prozess vorgenommen werden.

Um Nutzer*innen eine vertrauenswürdige alternative Quelle zum Google App Store zu bieten, sollte die Distribution der Software auf den open-source App Store F-Droid erweitert werden.

Eine verbesserte Entwicklungsdokumentation sollte den Entwicklungseinstieg in das Projekt erleichtern.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Die Zielgruppe für Bitmask sind Menschenrechtsorganisationen, Journalist*innen und Aktivist*innen, speziell solche, die unter repressiven politischen Regimen arbeiten. VPNs sind dabei elementar, aber aktuelle Lösungen oft unsicher und inkompatibel z.B. mit Skype. Populäre kommerzielle VPNs sind in solchen Teilen der Welt meist geblockt. Bitmask komplementiert dabei Tor, mit höherer Sicherheit als andere VPN-Lösungen, aber ohne die hohe Latenz von Tor.

Bitmask ermöglicht der Zielgruppe, einen uneingeschränlten Zugang zu Informatinen und sozialen Medien zu erhalten, sich vor digitalen Abhörmaßnahmen zu schützen und sicherer zu kommunizieren, ohne dafür besonderes technisches Wissen erlangt zu haben.

Ausführliche Darstellung der Ergebnisse

Das wesentliche Ziel die Software für den produktiven Einsatz vorzubereiten wurde erreicht, wenn auch einige wenige Meilensteine im Laufe der Entwicklung neu bewertet und angepasst werden mussten.

Den offensichtlichsten Fortschritt der Software ist aus Anwender*innenperspektive in der Überarbeitung der grafischen Oberfläche und der allgemeinen Stabilisierung der Software zu erkennen. Unter Beachtung der Material Design Guidelines von Google wurden sämtliche Oberflächen, die Navigation und der Screenflow überarbeitet. Dabei wurden wesentliche Teile der Softwarearchitektur refaktoriert, alte Fehlerquellen behoben und mit Hilfe von Tests abgesichert. Durch den Ausbau der Continuous Integration konnte sichergestellt werden, dass zukünftig im Rahmen der Entwicklung Regressionsfälle minimiert werden und das Schreiben von Tests fester Bestandteil des Entwicklungsprozesses wird. Mit diesem prozessualen Fortschritt kann die Softwarequalität dauerhaft gesteigert werden. Anhand der Google Developer Console, einem Werkzeug zur Verwaltung und Analyse eigener Apps im App Store, lässt sich nachweisen, dass in Folge dessen die Rate der Abstütze bei der täglichen App-Nutzung von 18,11% auf 0,76% reduziert werden konnte.

Ein weiteres erfolgreich abgeschlossenes Ziel ist die Reduktion des Batterieverbrauches. Implementiert wurde ein Mechanismus, der den Hintergrunddatenverkehr nach einigen Minuten Inaktivität des Smartphones blockiert und erst bei der Aktivierung des Smartphones wieder erlaubt. Dies entspricht dem Konzept eines sogenannten Doze-Modes.

Im Bereich der Softwaresicherheit wurden Fortschritte durch die Aktualisierung der Bibliotheken OpenVPN, OpenSSL, mBedTLS und der Basissoftware ics-openvpn erreicht. Darüber heraus wurden potentielle Sicherheitslücken durch das Forcieren des zur verschlüsselten Kommunikation genutzen Transportprotokolls

TLS 1.2 auch auf alten Android Plattformen geschlossen. Mithilfe des im Rahmen der Förderung überarbeiteten Certificate Pinnings wird der Angriffsvektor für ein Mitlauschen Dritter minimiert. In Bezug auf die Vermeidung von IP Leaks wurde ein Mechanismus implementiert, der verhindert, dass Datenverkehr außerhalb des VPNs bei noch nicht eingerichtetem Bitmask Profil stattfinden kann.

Schwierigkeiten ergaben sich bei der Implementierung des Obfuskierungsprotokolls obfs4. Die Implementierung wurde nicht angegangen, da sich die nötigen Änderungen des Backends nicht während der Förderungszeit realisieren ließen. Hier erwies es sich als problematisch, dass sich die Förderung ausschließlich auf den Android Client bezog und die Entwicklung des Backends auf der unentgeltlichen und freiwilligen Arbeit Dritter beruhte. Eine ähnliche Problematik entstand bei der Implementierung von IPv6. Während durch die Aktualisierung der Basissoftware ics-openvpn eine generelle IPv6 Kompatibilität bei Bitmask Android erreicht werden konnte, befindet sich diese Funktion jedoch nicht im Produktivbetrieb, da IPv6 Verbindungen seitens des Backends noch blockiert werden.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Mit der Weiterentwicklung des Bitmask Clients für Android hat die Zielgruppe eine einfach zu bedienende und stabil laufende VPN-Anwendung erhalten, die Sicherheitslücken durch individuelle Konfigurationsfehler ausschließt.
Quelloffener Code ist ein Schlüsselaspekt für sichere und vertrauenswürdige Software, da nur so versteckte Hintertüren ausgeschlossen werden können. Dies gilt insbesondere für VPN-Anwendungen, auf die sich die Anwender*innen im besonderen Maße verlassen können müssen, stabil und überwachungsfrei zu funktionieren. Da es für Bitmask kein dedizierten Geschäftsplan gibt, hilft die Offenlegung des Quellcodes außerdem, die Anwendung als Community Projekt durch die freiwillige Arbeitskraft vieler weiterentwickeln zu können. Der Aufbau und die Pflege einer solchen Community ist dafür essentiell und eine wichtige Aufgabe für die Zukunft.
Wichtige technische Weiterentwicklungen werden die Implementierung von Pluggable Transports, insbesondere von obfs4, sein, um Internetzensur noch effektiver umgehen zu können als es mit der bisherigen Lösung möglich ist. Darüber hinaus lassen sich noch viele mögliche Verbesserungen finden, z.B. die Implementierung einer verschlüsselten Datenbank, in der verschiedene VPN-Profile abgespeichert werden können, ein robusteres Verhalten bei Ausfällen einzelner VPN-Server eines Providers oder weitere Verbesserung im Kontext der User Experience.
Persönlich habe ich während der Förderungsphase viel gelernt, von dem ich zukünfig in meiner Arbeit profitieren kann. Dazu zählen der Ausbau der Continuous Integration, der Umgang mit der Containersoftware Docker und ein vertieftes Verständnis von OpenVPN.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Implentierung des Obfuskierungsprotokolls Obfs4 konnte nicht stattfinden, da es keine Testumgebung (kein ensprechendes Backend) gab, in der die Implementierung hätte getestet werden können.

Die Erhöhung der OpenVPN keep-alive Timeouts zur Reduktion des Batterieverbrauches hatte nicht den gewünschten Effekt und wurde verworfen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Anwendung ist sowohl im Google Playstore als auch in F-Droid unter dem Namen Bitmask zu finden. Der Source code befindet sich unter https://github.com/leapcode/bitmask_android. Weiterführende Informationen befinden sich auf der Bitmask Webseite https://bitmask.net.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es waren während der Förderungsphase keine Anpassungen der Arbeits- oder Kostenplanung.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab während der Förderungsphase Beiträge anderer Entwickler im Bereich der grafischen Oberfläche und bei Übersetzungen. Diese habe ich überprüft und nach dem Revisionsprozess ggf. integriert.