„Let's Encrypt“

Von Gilbert Brands und Bernd Roellgen

In unserer elektronischen Welt der Handys und des Internets sind die meisten Informationen ohnehin nicht zu verbergen. Wer wann wo gewesen ist, lässt sich aus dem Bewegungsprofil des Handys entnehmen, und wer welche Webseite besucht hat, geht aus den Verbindungsdaten hervor. Der einzige Schutz wäre Nichtbenutzung, aber wie es auf anderem Gebiet so schön heißt: *„Kein Alkohol ist auch keine Lösung“*. Was jedoch zu schützen ist, sind die Inhalte der Kommunikation. Konsequente Verschlüsselung vermiest Lauschern aller Art zumindest dieses Geschäft.

Verschlüsselung findet allerdings im Prinzip heute noch nicht statt: außer bei Besuchen von Webseiten oder in wenigen Chats läuft alles unverschlüsselt ab, und auch in den Bereichen, in denen etwas passiert, wird nur ein Teil der Information verschlüsselt. Vollends widersinnig wird die Sache, wenn Webseiten mit starker Verschlüsselung werben, um anschließend dem Nutzer den Extrakt des Seitenbesuchs in einer völlig unverschlüsselten Email nochmals mitzuteilen.

Das Verschlüsselungsstichwort beim Surfen im Internet heißt HTTPS oder SSL/TLS. Versprochen wird Sicherheit in dem Sinne, dass man wirklich mit der Webseite verbunden ist, die man aufrufen wollte, und auch keiner mitlesen kann. Das System beruht auf nicht fälschbaren elektronischen Ausweisen (Zertifikat genannt), die von einem Aussteller gegengezeichnet werden. Wenn man die Einstellungen seines Browsers öffnet, findet man irgendwo im Bereich „Advanced Options“ einen Punkt „Certificates“ und dort wiederum unter „Authorities“ die Zertifikate, die zur Gegenzeichnung verwendet werden. Spätestens beim Durchsehen dieser Liste sollte aber klar werden: Sicherheit bringt das nicht, zumindest nicht die versprochene. Die Verbindungen werden verschlüsselt, mehr aber auch nicht. Das System bringt hauptsächlich Geld in die Kassen der Authorities.

Warum das so ist, lässt sich leicht nachvollziehen: jede Auhority verfügt über mindestens ein Stammzertifikat, und schon die Liste der Authorities umfasst mehr als 50 Anbieter. Um die Kunden zu bedienen, reichen die Stammzertifikate nicht aus: mit ihnen werden nur so genannte Zwischenzertifikate signiert, die wieder in dritter Instanz die Seitenzertifikate signieren. Mancher Anbieter bringt es locker auf mehr als 50 Zwischenzertifikate. Was wird also garantiert außer der Verschlüsselung? Nichts – es sei denn, der Nutzer macht sich die Mühe, bei jeder Webseite die Zwischen- und Stammzertifikate zu identifizieren und in den Geschäftsbedingungen der Anbieter, meist ebenfalls mehr als 100 Seiten, nachzuschauen, was die eigentlich garantieren, außer der die Webseite einen größeren €-Betrag für die Signatur auf den Tisch legen musste.

Browser akzeptieren nur Webseiten, die bei den Anbietern eingekauft haben. Haben sie das nicht, gibt es zumindest auffällige Warnungen, die vom normalen Nutzer nicht zu beurteilen sind, sofern die Verbindung zur Webseite nicht grundsätzlich abgelehnt wird. Geplant ist, nicht nur ausschließlich Verschlüsselungen auf der Grundlage dieser Anbieter zuzulassen, sondern auch bislang unverschlüsselte Seiten einzubeziehen: alle Webseitenbetreiber sollen gezwungen werden, sich bei den Anbietern gegen Geld zu bedienen.

Aber nicht nur diese Machenschaften, das System selbst ist zweifelhaft. Zu OpenSSL, der meist verwendeten Software, heißt es

> *OpenSSL <http://openssl.org/> as a library at first glance is complicated, and then you realise that a lot of the documentation <http://openssl.org/docs/ssl/ssl.html> seems to be incomplete or missing.*

Wie verbaut das System ist, zeigt folgende Beobachtung: es ist durchaus zulässig, eine eigene Authority zu definieren und deren Zertifikate in die Liste zu importieren. Zumindest mit den Browsern, in denen diese Stamm- und

Zwischenzertifikate installiert sind, sollte alles reibungslos funktionieren. Tut es aber nicht unbedingt. In einem Versuch mit dem freien Anbieter http://www.CAcert.org kam heraus:

- Firefox unter Windows 10 akzeptiert ein so signiertes Serverzertifikat als Ausnahme. Das gleiche gilt für die MicroSoft-Browser.
- Firefox unter Ubuntu-Linux akzeptiert ein solches Zertifikat grundsätzlich nicht, egal ob man die Stamm- und Zwischenzertifikate importiert oder nicht. Es wird keine Verbindung mit der Webseite aufgebaut.
- Chrome unter Ubuntu-Linux akzeptiert das Zertifikat unter der Voraussetzung, dass man Stamm- und Zwischenzertifikat auf einem anderen Weg, beispielsweise mittels des Verwaltungsprogramms „Kleopatra", importiert und bearbeitet.

Drei Browser – drei Verhaltensweisen. Mozilla, der Hersteller von Firefox, meint dazu

> *.., because different browsers use different SSL stacks with different capabilities to find the missing certs, and it also depends on what other sites you have browsed to in that session and so whether the intermediate is in the cache.*

Was soll man von einem System halten, das nach Kommentaren der Macher nicht festen Regeln folgt, sondern von den Launen der Programmierer und vom völlig unverhersehbaren Verhalten der Nutzer abhängt? Auch an anderer Stelle ist OpenSSL in der Vergangenheit mit Schlampigkeiten aufgefallen: so scheinen die privaten Schlüssel deutlich weniger zufällig gewesen zu sein als notwendig, und auch zwischen verschiedenen Prozessen wiesen die Zufallszahlen Korrelationen auf. Mit den Leuten diskutieren zu wollen ist allerdings kaum möglich. Da (angeblich) alles freiwillig passiert, reagiert man auf Kritik (wie bei anderen OpenSource-Paketen auch) mimosenhafter als die üblichen Märchenprinzessinnen, und wer sich davon unbeeindruckt zeigt, muss mit verbalem Bashing rechnen. Aber man kann auch fragen, ob da wirklich Schlampigkeit am Werk war. In der englischen wikipedia findet man folgende Anmerkung zu OpenSSL:

> *"The project has a budget of less than $1 million a year and relies in part on donations. Steve Marquess, a former CIA consultant in Maryland started the foundation for donations and consultancy contracts and garnered sponsorship from the United States Department of Homeland Security <https://en.wikipedia.org/wiki/United_States_Department_of_Homeland_Security> and the United States Department of Defense <https://en.wikipedia.org/wiki/United_States_Department_of_Defense> . [3] <https://en.wikipedia.org/wiki/OpenSSL#cite_note-wsj-3> "*

Glücklicherweise ist man als Entwickler nicht an OpenSSL gebunden. Mit BotanSSL steht (mindestens) eine weitere SSL/TLS-Bibliothek in C++ zur Verfügung, die besser dokumentiert ist und manche der OpenSSL-Unzulänglichkeiten nicht aufweist. Aber es stellt sich die Frage, was passiert, wenn der Branchenführer sich an gewissen Stellen nicht an die Norm gebunden fühlen sollte, sei es nun aus Gründen der „künstlerischen Freiheit" der Programmierer oder aufgrund nicht zurückweisbarer Wünsche der Finanziers.

Zurück zum Problem des für normale Internetnutzer nicht durchschaubaren Zertifikatwesens. Ob um die Geldruckmaschinen der Authorities etwas einzudämmen oder aus was für Gründen auch immer hat sich die Zertifikat-Plattform „Let's Encrypt" etabliert, hinter der sich eine „Internet Security Research Group" verbirgt, die sich hauptsächlich aus Vertretern der Internet-Industrie zusammensetzt (es wäre sicher interessant, Leute und Unternehmensbeziehungen zu untersuchen, um die Motivation für dieses Projekt besser verstehen zu können; möglicherweise stößt man da auf ähnliche Überraschungen wie bei OpenSSL). Sie will globale Verschlüsselung ermöglichen, zunächst im HTTP-Bereich, aber im Text wird auch der E-Mail-Bereich angesprochen, wenn auch technische Details bislang vollständig fehlen. Genauso wichtige weitere Bereiche wie Telefonie oder Chat werden allerdings gar nicht erst angesprochen. Das Konzept basiert auf dem normalen X.509-Zertifikathandling, allerdings sind die Zertifikate im Gegensatz zu denen der Authorities kostenlos.

Noch scheint das Projekt nicht sonderlich weit gediehen zu sein. Die Bedienung ist einfach: man lädt ein so genanntes CertBot-Programm und teilt diesem mit, man möchte gerne ein Zertikat haben. Alles andere läuft automatisch, was schon ein sehr großer Fortschritt gegenüber dem heutigen, selbst für Server-Administratoren komplexen Verfahren ist. Der CertBot erstellt ein Zertifikat und lässt dieses in einem automatisierten Verfahren von einem Zentralserver signieren. Da anscheinend alle Browser-Hersteller kooperieren, werden diese Zertifikate anschließend auch problemlos akzeptiert. Ein echter Fortschritt? Wenn man von der einfachen Bedienung absieht, konzeptionell eher ein deutlicher Rückschritt. Wieso?

Derzeit ist anscheinend nur das Verfahren für die Erstellung der Server-Zertifikate geregelt. Das Zertifikat wird automatisch erstellt, an einen Zentralserver gemeldet, und dieser überprüft im Gegenzug, ob unter der angegeben URL ein Server zu erreichen ist, der über den privaten Schlüssel verfügt. Ist das der Fall, wird das Zertifikat gültig signiert. Weitere Kontrollen existieren nicht. Modellvorstellungen, wie das Konzept mit E-Mails funktionieren soll, sind noch nicht zu finden. Die wichtigsten Konsequenzen aus der automatischen Zertifikatausgabe nach dem derzeitigen Muster:

1. Das eigentliche Sicherheitskonzept hinter den Zertifikaten – dass der Inhaber einer URL auch tatsächlich derjenige ist, der er vorgibt zu sein – wird damit fallen gelassen. Es ist z.B. ohne Weiteres möglich, sich vorübergehend eine URL zu besorgen, sich dafür ein Zertifikat erstellen zu lassen, und anschließend darauf zu warten, dass irgendjemand die URL übernimmt und man versuchen kann, irgendwelchen Unfug zu treiben. Versprechen heutige Authorities auch noch irgendwelche mit 50.000 € oder mehr dotierte Garantien, dass sie sich von der Seriosität des Inhabers überzeugt haben – bei Let's Encrypt gibt es keine mehr. Der normale Nutzer kann aber nur mit sehr viel Aufwand zwischen „Let's Encrypt“- und teuren Standardzertifikaten differenzieren, oder mit anderen Worten: de fakto ist der letzte Rest der versprochenen Sicherheit dahin.

2. Die Erstellung von gültigen Zertifikaten ist an die Kontrolle über den Server gebunden, da der CertBot nur von Administratoren aufgerufen werden kann. Wer seine Seiten irgendwo hostet, kommt folglich an solche Zertifikate selbst nicht dran, sondern muss beim Provider „bitte-bitte“ machen oder hoffen, dass der auf den Zug aufspringt und das für ihn ohne Aufforderung übernimmt.

Man mag nun argumentieren, die Beschreibungen der Modelle seien noch nicht ausgereift oder es würden noch Work-Arounds für das eine oder andere Problem entwickelt. Beispielsweise könnte der Zentralserver ja nur ein Zertifikat für eine Webseite zulassen und bei erneuter Beantragung nach dem in 1. erwähnten Angriffsmodell das alte zurückziehen. Damit würde aber nur das Tor zu massiven Denial-of-Service-Angriffen geöffnet, denn ein Angreifer mit entsprechenden Mitteln könnte beliebig Zertifikate laufender Server ungültig werden lassen. Wie man es dreht oder wendet: ein automatisiertes Zentralsystem versagt genau bei den Angriffsszenarien in epischer Breite, für deren Verhinderung das Zertifikatsystem ursprünglich einmal entwickelt worden ist.

Zusammengefasst: Let's Encrypt ist so notwendig wie ein Pickel auf der Nase, da die Sicherheit eher geringer ist als wenn jeder selbst seine Zertifikate erstellt und signiert. Mit solchen selbst signierten Zertifikaten kann man nach den aktuellen Standardprotokollen heute ebenfalls arbeiten, indem man solche Zertifikate als Ausnahmen zulässt, aber die Browser reagieren darauf mit für den Nutzer deutlich zu heftigen Meckereien oder, wie im Fall des Ubuntu-Firefox und möglicherweise der zukünftigen Standardreaktion, mit einer protokollwidrigen Arbeitsverweigerung. Let's Encrypt dient bestenfalls zur Rettung für das als inzwischen nur noch bedingt taugliche Zertifikatkonzept, und mit der eigenwilligen Interpretation, welche Zertifikate sicher sind und welche nicht, wird der Internetnutzer gezwungen, völlig entmündigt alles einer anonymen Organisation zu überlassen, die letzten Endes doch für nichts garantiert oder gar – aber das ist jetzt Verschwörungstheorie – für die Interessen derjenigen arbeitet, vor denen man sich schützen will.

Jeder Polizist rät, sein Haus und sein Eigentum selbst zu sichern, um Gaunern das Handwerk zu erschweren. Warum darf das nicht im Internet gelten? Weil irgendwann mal OpenSSL konstruiert wurde und jemand beleidigt

sein könnte, wenn man die inzwischen unzulänglichen Konzeptteile über Bord wirft? Statt wie bei Let's Encrypt weiterhin auf eine Zentralinstanz zu setzen und dem Nutzer eine wirksame Eigenverantwortung zu entziehen, sollte man ihm nicht besser mehr Kontrolle und Verantwortung übertragen und ihn sein Eigentum selbst schützen lassen? Die Zertifikate, bei denen man sicher sein MUSS (es sind nämlich gar nicht so viele, wie immer suggeriert wird), dass tatsächlich die angegebene Person oder Institution dahinter steckt, kann man auch selbst verwalten und muss nicht hoffen, dass eine anonyme Zentralinstanz keine falschen Daten ausliefert. Das entspräche dem Konzept der *Elektronischen Identität*, die wir anderswo entwickelt haben. Die Bedienung wäre nicht komplizierter als bei Let's Encrypt: ein EIBot erstellt die notwendigen Zertifikate und verwaltet sie. Das Konzept bräuchte noch nicht einmal neue Software-Entwicklungen: weist man die Browser an, selbst signierte Zertifikate nicht in schreiend bunten Warnfarben als „gefährlich", sondern schlicht als „Elektronische Indentitäten" zu kennzeichnen und den Rest dem EIBot zu überlassen, wäre das vermutlich in 20 Programmzeilen zu realisieren. Auch E-Mails, Telefonie, Chats oder gar abgeschlossene Sicherheitsbereiche wie ein Unternehmen oder der zukünftige autonome Straßenverkehr sind problemlos mit dieser Technologie einzubinden.

Wie es weitergeht, bleibt abzuwarten. Weiterhin alles den US-Amerikern überlassen, bei denen sich am Ende herausstellt, dass doch wieder Homeland Security in irgendeiner Form dahinter steckt? Oder selbst einmal etwas in die Hand nehmen? Eine Alternative wäre natürlich noch dieses Kommunikationsmodell aus den 1920er Jahren: