Überwachung

und die Kunst der digitalen Tarnung

Michael Anders

April 2021

Für meine kinder.

Ich habe dieses Buch in großer Bewunderung für Edward Snowden, für Hans und Sophie Scholl und für all die anderen Menschen geschrieben, die Recht von Unrecht unterscheiden können und die mutig genug sind, auch im Machtbereich des Monsters die Wahrheit laut auszusprechen.

Der Sinn dieses Buches:

Dieses Buch ist für IT-Laien geschrieben, die ihren privaten Computer benutzen, um mit Freunden in Kontakt zu bleiben, mit Fremden in Kontakt zu kommen oder einfach nur möglichst sicher im Internet zu surfen. Gegenüber einem am Arbeitsplatz genutzten, vom Unternehmen administrierten System eröffnet das privat genutzte System größere Risiken, aber auch wesentlich größere Freiheiten und Chancen. Es geht in diesem Buch darum, diese Freiheiten und Chancen zu nutzen, um die persönliche elektronische Kommunikation erheblich besser gegen Zugriff durch staatliche oder kriminelle Organisationen zu schützen, als das üblicherweise am Arbeitsplatz geschieht.

Über den Autor:

Dr. Michael Anders ist Physiker. Nach dem Studium der Physik an der Universität Gießen und an der University of Washington in Seattle, der Promotion in Angewandter Physik und Forschungstätigkeit im Bereich Quantenelektronik und Rastersondenmikroskopie am Forschungszentrum Jülich war er bei der BASF AG in Ludwigshafen tätig. Im Jahre 1993 wechselte er zurück in den akademischen Betrieb und wurde Dozent an der Fachhochschule Wedel. Für viele Jahre war er Studiendekan des Studienganges Wirtschaftsingenieurwesen und Betriebsratsmitglied der privaten Hochschule in Wedel. Für einen breiten Hörerkreis von Studenten aus Studienfächern von BWL über Physik- und Wirtschaftsingenieurwesen bis zu diversen Informatikstudiengängen leitete er viele Lehrveranstaltungen in den Bereichen Statistik, Mathematik, Physik, Fertigungstechniken der Elektronik und Mikrosystemtechnik sowie Seminare zu den Themen Kryptografie, Anonymität und praktische Verschlüsselungstechnik.

Im Jahr 2011 entwickelte er das quelloffene Verschlüsselungswerkzeug Academic Signature, das mit Elliptischer-Kurven-Algebra Verschlüsselung sowie die Erstellung und Verifikation digitaler Signaturen und von Zeitstempeln ermöglicht. Das Programm wurde unter der GNU[1] General Public License als freie Software veröffentlicht (https://www.academic-signature.org). Neben dem wesentlich weiter verbreiteten GnuPG ist es das einzige quelloffene, flexibel einsetzbare eigenständige Werkzeug für die hybride Verschlüsselung beliebiger Dateien.

Vorwort zur zweiten deutschen Auflage dieses Buches:

Liebe Leser, nach der ersten deutschen und der ersten englischen Ausgabe dieses Buches folgt nun die zweite Auflage der deutschen Version. Satz, Schriftbild und Struktur der englischen Version waren professioneller gelungen als bei der ersten deutschen Version. Außerdem hatte ich für die Übersetzung das Buch inhaltlich überarbeitet, erweitert und aktualisiert. Diese Verbesserungen möchte ich nun auch in einer neuen deutsche Ausgabe verfügbar machen.

Die Arbeit an diesem Buchprojekt und den beiden Vorläufern war von Anfang an niemals kommerziell motiviert. Ich möchte Sie ermächtigen, ausbilden und schulen. Es wird dem Leser[2] hoffentlich Vergnügen bereiten, sicherlich jedoch vom Leser auch Anstrengungen erfordern, die zahlreichen Übungen dieses Buches zu bearbeiten. Obwohl zwar die meisten Menschen grundsätzlich daran interessiert sind, digitale Übergriffe abzuwehren, würden viele davor zurückschrecken, dieses Buch durchzuarbeiten, um die notwendigen digitalen Abwehrtechniken zu erlernen. Solchen Menschen dürfte eher mit einem Buch gedient sein, das eine gewisse Empörung und Klage über die allgegenwärtige Überwachung zum Ausdruck bringt, das einfache zu lesende Ratschläge bietet, die nie eingeübt werden müssen, aber leider in der Praxis nicht besonders wirkungsvoll sind. Es gibt viele solcher Bücher, und sie verkaufen sich in großer Zahl.

[1] https://de.wikipedia.org/wiki/GNU

[2] Bitte entschuldigen Sie, wenn ich meist der Einfachheit halber die männliche Form der Anrede wähle. Ich bin mir sehr wohl bewusst, dass viele sachkundige Leser nicht männlich sind und es viele hervorragende Kryptografinnen, Hackerinnen und Software-Entwicklerinnen gibt.

Dieses Buch ist anders. Obwohl Sie einige theoretische Abschnitte einfach nur sorgfältig lesen und verstehen müssen, erfordert die Arbeit mit den meisten anderen Teilen Ihren hochgefahrenen Computer neben dem Buch, einen zwischen dem Computerbildschirm und den Buchseiten hin- und herschweifenden Blick und Ihre Finger auf der Tastatur. Ich möchte Sie ermutigen, die Behauptungen aus diesem Buch mit den Ergebnissen eigener Internet-Recherchen zu vergleichen, die Ergebnisse mit Ihrem System zu reproduzieren, freie quelloffene Software zu Ihrer digitalen Verteidigung zu installieren, zu benutzen und hoch entwickelte Kommunikationsmethoden einzuüben. Verlassen Sie sich nicht einfach auf Aussagen aus diesem Ratgeber, sondern überprüfen Sie immer wieder die Richtigkeit der Angaben. Experimentieren Sie mit Ihrem System, um Behauptungen zu verifizieren oder zu falsifizieren.

Die in diesem Buch vorgestellten Methoden sind von Natur aus defensiv. Doch die Beherrschung der Techniken zur verdeckten Internetkommunikation kann Ihnen eine gewisse Macht verleihen. Nutzen Sie diese Macht niemals, um jemandem durch anonyme Drohungen, persönliche Entlarvung, Beleidigungen, Diebstahl, Internetbetrug, Erpressung, Stalking oder durch digitale Hilfe für Verschwörungen zu gewalttätigen Handlungen zu schaden. Respektieren Sie stets die digitale Privatsphäre und bewahren Sie die digitale und physische Unversehrtheit aller Mitmenschen. Unterstützen Sie schutzbedürftige Menschen in der digitalen Welt.

Jeder darf sich natürlich um das Erlernen der digitalen Tarnung bemühen, das Buch ist für jeden offen. Dennoch wünsche ich mir, dass religiöse und politische Extremisten, insbesondere diejenigen aus dem fremdenfeindlichen rechten Sektor, von diesem Buch fernbleiben. Alles in allem sind Computer und das Internet großartige Errungenschaften mit dem Potenzial, der Menschheit zu dienen. Sie ermöglichen, über alle Grenzen hinweg zu kommunizieren, aus Unterdrückung auszubrechen und die Welt zu einem besseren Ort zu machen.

Testleser einer Vorabversion haben mich gebeten, darauf hinzuweisen, dass man das Buch auch mit großem Gewinn lesen kann, ohne die Übungen zu absolvieren.

An dieser Stelle möchte ich herzlich Dr. Reinhard Maaß für die sorgfältige Durchsicht des Manuskriptes und viele wertvolle und hilfreiche inhaltliche Anregungen danken.

Michael Anders,
Klein Nordende, Deutschland, im April 2021.

Rechtliches:

Zweck dieses Buches ist, den Leser durch Argumente von der Wirksamkeit bestimmter Schutzmaßnahmen gegen Überwachung zu überzeugen. Er soll durch die Ausführungen im Buch zu fundierten eigenen Entscheidungen befähigt werden. Wenn er die im Buch vorgestellten Werkzeuge

und Verfahren nutzt, tut er das aber auf eigene Verantwortung und nach eigener Abwägung und Entscheidung.

Die Verwendung aller im Buch vorgestellten Techniken und Werkzeuge ist nach meinem Kenntnisstand in Deutschland legal. Dies ist aber nicht in allen Nationen der Fall. Bitte erkundigen Sie sich nach der rechtlichen Situation in Ihrem Heimatland und nutzen Sie im Buch vorgestellte Techniken und Werkzeuge nicht in einem Land, in dem deren Nutzung verboten ist. Ich empfehle Ihnen schweren Herzens, sich in diesem Fall den demokratiefeindlichen, menschenverachtenden Überwachungspraktiken des betroffenen Landes zu unterwerfen.

In einigen demokratischen europäischen Ländern ist Einsatz und Verbreitung wirksamer Verschlüsselung ohne Regierungslizenz verboten. Mir ist aber nichts davon bekannt, dass in unseren demokratischen Nachbarländern der Versuch unternommen würde, dieses Verbot auch durchzusetzen.

Ich habe alle in diesem Buch enthaltenen Informationen mit der gebotenen Sorgfalt zusammengestellt. Trotzdem kann ich Fehler oder veraltete Informationen nicht ausschließen. Ich stelle hiermit klar, dass ich unter keinen Umständen für Schäden haftbar gemacht werden kann, die durch unpräzise Informationen oder missverständliche Formulierungen in diesem Buch entstehen. Außerdem übernehme ich keine Verantwortung für die Verfolgung des Lesers durch die örtlichen Behörden, falls sie ihn bei der Anwendung von Techniken erwischen, die im Aufenthaltsland des Lesers als ungesetzlich oder unerwünscht gelten. Ebenso rate ich Ihnen dringend davon ab, während der Lektüre dieses Buches frisch gekochten Kaffee zu trinken. Sie könnten sich verbrühen.

Selbstverständlich bin ich für Hinweise meiner Leser auf fehlerhafte Informationen in diesem Buch oder die Notwendigkeit einer Aktualisierung veralteter Passagen dankbar. Software, die in diesem Buch zur Installation vorgeschlagen wird, darf nur für private Zwecke verwendet werden. Für eine kommerzielle Nutzung ist möglicherweise die Zustimmung des Inhabers der entsprechenden Lizenz erforderlich. Die Namen bestimmter Soft- und Hardware oder Markennamen, die in diesem Buch genannt werden, können ohne ausdrückliche Erwähnung dem Marken- oder Patentschutz unterliegen.

Herausgeber und Vertrieb: epubli GmbH, Berlin, www.epubli.de

Hinweis: *Alle im Folgenden in der Papierversion dieses Buches wiedergegebenen Internetadressen sind unter:*
https://www.academic-signature.org/buch/digitale-tarnung.php
komfortabel sortiert und anklickbar gelistet.

Inhaltsverzeichnis

1. Die Sicherheit privater Computer

1.1 Beruflicher und privater Umgang mit dem Computer

Am Arbeitsplatz wird sich die IT-Abteilung im Idealfall gut um die IT-Sicherheit der Arbeitsplatzrechner gegen Kriminelle, aber im Regelfall leider weniger gut um die Vertraulichkeit der beruflichen Kommunikation kümmern. Zur Herstellung von Sicherheit gegen Kriminelle wird die IT-Abteilung die Rechte der Mitarbeiter auf den bereitgestellten Systemen stark beschränken und den Mitarbeitern vernünftige und - besonders im Passwortbereich - auch teilweise absurde Pflichten auferlegen. Wer kennt nicht Anweisungen, etwa drei unterschiedliche Passwörter für drei verschiedene Systeme mit jeweils Sonderzeichen, Großbuchstaben und Zahlen in gewisser Mindestlänge zu nutzen, mindestens alle 60 Tage zu ändern, aber ganz bestimmt nirgends zu notieren?

Man hat als Mitarbeiter im Gegenzug aber das beruhigende Gefühl, nicht für die Sicherheit und Funktionalität der IT verantwortlich zu sein. Bei der kommerziellen IT-Sicherheit im Unternehmensumfeld liegt notwendigerweise der Fokus auf ganz anderen Stellen als bei der in diesem Buch behandelten sicheren Kommunikation im privaten Umfeld. Ich möchte jetzt diese Unterschiede herausarbeiten.

Bei der IT-Sicherheit im Unternehmen ist das Hauptproblem, die in diesen Dingen unmotiviertesten und am wenigsten kundigen Mitarbeiter von der Gefährdung der Unternehmens-IT durch Dummheiten abzuhalten, die Möglichkeiten für diese zu begrenzen, massiven Schaden anzurichten und notfalls bei solchen Schäden wenigstens möglichst früh alarmiert zu werden. Die unkundigen und, als Kollateralschaden, auch alle anderen Mitarbeiter müssen in ihren Möglichkeiten beschränkt und in ihren Arbeitsabläufen kanalisiert werden. Sie müssen davon abgehalten werden, beispielsweise in Outlook berüchtigte Mailanhänge des Prinzen aus Nigeria anzuklicken. Durch regelmäßige Systemupdates und rigorose Blockade nicht für notwendig erachteter Kommunikationsmodi (Blockade von Ports in der Firewall) muss gleichzeitig die Anzahl der für Angreifer nutzbaren Softwarefehler in Grenzen gehalten werden. Westliche staatliche Dienste werden wider besseres Wissen als Bedrohung der vertraulichen Kommunikation meist vernachlässigt. Die Gefahr

der Informationslecks durch illoyale Unternehmensinsider wird als wesentlich bedrohlicher eingestuft. Deshalb würde in diesem Kontext die Verfügbarkeit von Ende-zu-Ende-Verschlüsselung für kommunizierende Mitarbeiter eher als Bedrohung der Unternehmenssicherheit denn als Schutz gegen Industriespionage gesehen. Das sind wenig günstige Rahmenbedingungen für einen effektiven Schutz der Mitarbeiterkommunikation.

Wenn man sich aber, wie in diesem Ratgeber, mit der Sicherheit privater digitaler Kommunikation befasst, ist man in einer komfortableren Situation als im Unternehmensumfeld. Wer dieses Buch in die Hand nimmt, ist hoch motiviert und an der Sicherheit der eigenen Kommunikation interessiert. Ich kann beim Leser zwar anfangs nicht auf allzu große IT-Kenntnis oder gar kryptografische Fachkenntnis, dafür aber auf Interesse und geistige Beweglichkeit vertrauen und kann Experimentierfreude begrüßen. Das ist eine ausgezeichnete Ausgangsposition.

Beim Umgang mit dem privaten Computer sind Sie selbst in der Pflicht, durch umsichtiges Verhalten die Risiken zu minimieren, Opfer von Internetkriminalität zu werden und Schäden durch Angriffe in Grenzen zu halten. Zur Umsicht gehört, keine Speichermedien mit unklarer Vorgeschichte mit dem System zu verbinden und vernünftigen Umgang mit Passwörtern zu pflegen. Weiterhin sollte man ein System, mit dem Bankgeschäfte oder Einkäufe getätigt werden oder diesbezüglicher E-Mail-Verkehr betrieben wird, nicht zum ungeschützten Surfen auf "Blinki-Bunti-Schmuddelseiten" verwenden. Wenn man dies verinnerlicht und sich ähnliche Restriktionen auferlegt, wie sie im beruflichen Umfeld erzwungen werden, kann man auf einem privaten System eine dem beruflichen Umfeld vergleichbare Sicherheit erreichen. Man könnte darüber hinaus sogar vielleicht noch die üblichen Ratschläge zur regelmäßigen Datensicherung befolgen und einen aufmerksamen Blick auf die Logs des Systems, vielleicht sogar auf den Umfang des ein- und ausgehenden Datenverkehr haben und bei Auffälligkeiten sofort einschreiten.

Solche Maßnahmen der allgemeinen umsichtigen Verwendung des Computers werden in diesem Ratgeber aber nur am Rande thematisiert. Hier lege ich den Fokus auf die Vertraulichkeit Ihrer elektronischen Kommunikation. Bei den später folgenden Ausführungen unterstelle ich, Ihr privates System sei sicher und staatliche oder kriminelle Organisationen hätten ohne Ihre ausdrückliche Zustimmung keinen Zugriff auf Ihren Computer.

1.2 Das Angebot der Softwarekonzerne für private Computernutzer

Die Maßnahmen zur allgemeinen Systemsicherheit erfordern eine gewisse Mühe und ein gewisses Verständnis auf Nutzerseite. Um dem Nutzer die Mühe und die meist ungeliebte Suche nach Verständnis zu erleichtern, tendieren die Hersteller der gängigen kommerziellen, für Privatnutzer gedachten Betriebssystemen dazu, die Systeme immer mehr gegenüber dem Nutzer zu verschließen und für Fernwartung aufzubereiten. Diese Fernwartung läuft dann ohne Mitwirkung und Information des Nutzers im Hintergrund ab. Eine solche Fernwartung ohne Nutzerkontakt wird aber niemals die Sicherheit und Flexibilität herstellen, wie man sie bei einem durch einen kundigen Menschen administrierten System erreichen kann.

Die Fernwartbarkeit am Nutzer und Besitzer vorbei bringt leider auch eine starke Versuchung für Softwarehersteller, diesen unkontrollierten Zugang zum privaten System für allerhand lästige Marketingaktivitäten und Handel mit abgezogenen Nutzerdaten zu missbrauchen. Die Marketingaktivitäten sind hierbei der für uns Nutzer offen sichtbare Teil.

1.3 Das Ringen um Zugang zu Ihren privaten Daten

Für eine Minderheit der Nutzer, die die volle Kontrolle über ihr System und ihre Daten behalten wollen, ist die Gefahr des Machtmissbrauchs, die zunehmende Beschränkung der Rechte am eigenen System und die allzu bereitwillige Kooperation der Softwarekonzerne mit den Nachrichtendiensten ihrer Heimatländer ein großes Ärgernis.

Ein Beispiel: Mitte Oktober 2017 wurde öffentlich bekannt, dass in vielen sogenannten TPMs (Trusted Platform Module) eine fehlerhafte Bibliothek für die Generierung von RSA-Schlüsseln[1] verwendet wurde[2].

Der Fehler führt zu unsicheren RSA-Schlüsselpaaren. Es handelte sich aber nicht um einen Fehler im Zufallszahlengenerator.

Das TPM ist ein in Desktop Computern und Notebooks eingebauter Chip, der auch zur Prüfung digitaler Signaturen vor der Ausführung von heruntergeladenen Installations- oder Updatedateien verwendet wird. Die fehlerhafte Bibliothek liegt in der Firmware; das ist auf dem Chip selbst vorliegende, gegen Manipulation von außen gesondert geschützte Software.

Der Fehler erhielt die offizielle Kennung CVE-2017-15361 und den Namen ROCA, Return of Coppersmith's Attack. Auch Infineon TPMs wurden 5 Jahre lang mit dem fehlerhaften Programm ausgeliefert. Nun muss man wissen, dass zur Herstellung eines RSA-Schlüssels zwei lange Primzahlen gefunden werden müssen. Dazu wird, ausgehend von einer ungeraden Zufallszahl entsprechender Länge, meist durch doppeltes Hochzählen und jeweiliges Anwenden eines probabilistischen Primzahltests, die nächste Primzahl gesucht und gefunden. Dies wird zweimal ausgeführt. Das Produkt der zwei gefundenen Primzahlen zusammen mit einer frei gewählten, meist deutlich kleineren Zahl, dem sogenannten öffentlichen Exponenten, ist der öffentliche Schlüssel. Kenntnis der beiden Primfaktoren ist der private Schlüssel.

Wenn nun der Zufallszahlengenerator, wie verbreitet wurde, nicht betroffen war, stellt sich folgende Frage: Was hat die Software der fehlerhaften Chips für ungewöhnliche Kriterien, nicht die jeweils nächsten Primzahlen nach einer Zufallszahl, sondern andere, handverlesene, zu leichter knackbaren Schlüsseln führende Primzahlen auszuwählen? Die von Infineon verbreitete Erklärung, so hätten die benötigten Primzahlen schneller erzeugt werden können, erscheint mir absurd. RSA-Schlüssel werden höchstens ein paar Mal in der Lebenszeit des Chips errechnet und dies dauert im Chip, wenn es korrekt durchgeführt wird, vielleicht eine Minute. Ich vermute, dass bewusst eine Hintertür für staatliche Akteure eingerichtet war.

Als die Sicherheitsforscher der Masaryk Universität in Brünn, der britischen Sicherheitsfirma Enigma Bridge und der Ca' Foscari Universität in Venedig die Lücke entdeckten, musste auf Seiten der staatlichen "Bedarfsträger" mutmaßlich rasch gehandelt werden, um den Schaden zu begrenzen. Selbstverständlich waren die betroffenen TPMs vom amerikanischen NIST[3] und vom deutschen BSI bei deren Einführung als sicher zertifiziert worden.

Auch ohne besondere Kenntnisse in Kryptografie kann jeder verstehen, dass eine vorhergehende Einschränkung der Auswahl auf wenige spezielle Primzahlen dem Angreifer das Ermitteln der tatsächlich im privaten Schlüssel verwendeten Primzahlen erleichtert, wenn dem Angreifer die Art der Einschränkung bekannt ist. Bei jedem seriösen Audit einer Implementation des RSA-Algorithmus würde man als einen der ersten Punkte klären, wie die verwendeten Primzahlen bestimmt werden und ob tatsächlich aus der vollen Primzahlmenge zufällig ausgewählt wird. (Auf alles andere als ein klares "Ja" zur letzten Frage würden normalerweise einige Augenbrauen sehr weit hochgezogen werden.) Die Tatsache, dass das offensichtlich nicht geschehen ist, ist sehr beunruhigend. Bezüglich der staatlichen IT-Sicherheitsbehörden BSI und NIST muss man sich nun entscheiden, ob man die beteiligten Mitarbeiter für Saboteure oder für inkompetent hält. In einem Artikel in Zeit-Online wurde sehr vorsichtig die zweite Variante angedeutet. Der Autor des Artikels bezeichnete den Vorgang als peinlich für das BSI[4]. Ein weiteres Nachhaken vonseiten des investigativen Journalismus erfolgte leider nicht.

[1] RSA ist ein asymmetrisches Kryptosystem, das auf Seite 127 näher erläutert wird.

[2] https://crocs.fi.muni.cz/public/papers/rsa_ccs17

[3] NIST ist das US-amerikanische National Institute of Standards and Technology, eine in etwa der deutschen PTB entsprechenden Behörde für die Pflege von Maßeinheiten und Standards mit zusätzlicher Zuständigkeit für IT-Belange.

[4] https://www.zeit.de/digital/datenschutz/2017-10/infineon-verschluesslung-personalausweis-tpm-bsi-zertifiziert

Nun müssen Sie, liebe Leser, für sich den hier als Beispiel dargestellten Vorfall selbst bewerten und einordnen. Ich persönlich neige dazu, die Mitarbeiter des BSI und des US-amerikanischen NIST nicht gleichzeitig für inkompetent zu halten und vermute Sabotage.

Natürlich müssen Sie diese Einschätzung und meine kritische Sicht auf das Agieren der Behörden BSI und NIST nicht teilen. An dieser Stelle möchte ich ausdrücklich alle Leser dieses Buches, unabhängig von ihrer Sicht auf solche Ereignisse, willkommen heißen. Bei den Lesern, die der von meiner Meinung abweichenden Auffassung zuneigen, die offiziellen Stellen und großen Firmen hätten in aller Regel recht, seien ehrlich und es seien viele unberechtigte Verschwörungstheorien in Umlauf, möchte ich mich vorauseilend für gelegentliche Wertungen in meinen Ausführungen entschuldigen. Im Schlusskapitel 6 dieses Buches, nach dem Abschluss der eher politisch neutralen technischen Passagen, werde ich meine persönlichen Wertungen aber erneut deutlich und unmissverständlich formulieren. Es könnte allerdings selbst in dem technisch orientierten Hauptteil des Buches passieren, dass mir hier und da wertende Adjektive in den Text hineingeraten sind, die Ihnen vielleicht nicht gefallen. Bitte sehen Sie mir das nach. Letztlich hat mir meine persönliche Betroffenheit über den Umgang unserer politischen Führungen mit den Enthüllungen von Edward Snowden den Anstoß zum Verfassen dieses Buches gegeben. Es ist aber primär ein technisches Buch.

Leider ist die Sicht auf die Arbeit unserer westlichen Nachrichtendienste und auf das Agieren der Internetkonzerne ziemlich polarisiert. Der eine betrachtet mich und meine Mitstreiter als Haufen wirrer Verschwörungstheoretiker. Die Seite, zu der ich mich zugehörig fühle, ist dagegen in Gefahr, jeden Kritiker quelloffener und freier Software und jeden konzern- oder behördennah argumentierenden Diskussionsteilnehmer als faschistoiden Bespitzelungsverfechter zu sehen. Wenn man auf dem Gebiet der Technik virtuell die Klingen kreuzt, kann man sich aber über technikzentrierte Diskussionen auch wieder politisch näherkommen. Wir Abhör- und Datensammelgegner müssen uns in die Gedankenwelt unserer Angreifer hineinversetzen, um in der Abwehr gut zu sein. Außerdem müssen wir ständig unsere Feindbilder auf den Prüfstand stellen. Unangemessen überbordendes Misstrauen macht unsere Abwehr nicht besser.

Falls Sie, lieber Leser, aber eher zur anderen Seite gehören und uns Überwachungsgegner für unangemessen misstrauische Anarchisten halten, sollten auch Sie gelegentlich gedanklich in unsere Haut schlüpfen. Auch Sie sollten Ihre Feindbilder hinterfragen. Wie fühlen Sie sich damit, von Behörden überwacht zu werden, die es als Geheimnis behandelt wissen wollen, welchen Umfang und welche Intensität diese Überwachung hat? Vielleicht hinterfragen Sie sogar für einen Moment kritisch, ob die eine oder andere zunächst von Ihnen begrüßte staatliche Maßnahme nicht in Wirklichkeit tatsächlich den Weg auch in Ihre Unfreiheit und in eine Diktatur ebnen könnte. Auf jeden Fall heiße ich Sie, so wie Sie sind, als Leser willkommen und auch Sie sollten vom technischen Teil dieses Buches profitieren können.

Zum Schluss möchte ich diejenigen Leser begrüßen, die ich von Beginn an als Adressaten vor Augen hatte. Sie sind intelligente Computerlaien, haben aber durchaus Interesse an Computerdingen und dem Internet. Sie sind nicht ideologisch festgelegt und Ihre Bewertung des eingangs dieses Abschnittes vorgestellten Fehlers im TPM Ihres Computers ist noch im Fluss. Sie sind sich aber gewisser Fakten bewusst: Es gab die Snowden-Enthüllungen im Jahr 2013. Westliche staatliche Dienste waren außer Rand und Band. Ob sie sich heute demokratiekonform verhalten oder es verdeckt heute noch schlimmer treiben, wissen wir nicht. Die Vorsicht legt aber nahe, sich gegen den letzteren Fall zu wappnen. Weiterhin gibt es als Bedrohung feindliche Nachrichtendienste und die stetig wachsende, ganz normale Internetkriminalität. Sie möchten deshalb die Fähigkeit zu sicherer, verdeckter Kommunikation erwerben und später vielleicht auch weitergeben. Herzlich willkommen, genau für Sie habe ich dieses Buch geschrieben.

Unabhängig davon, wie ihr Interesse für dieses Buch begründet ist, werden alle Leser mit etwas Sorgfalt und der Suche nach Verständnis mit Ihrem privaten Computer ein Niveau an Selbstbe-

stimmung und Sicherheit der Kommunikation erreichen können, das bei fremdadministrierten oder gar hersteller-fern-administrierten Systemen undenkbar ist. Abgesicherter elektronischer Kontakt mit schillernden Kommunikationspartnern und Webseiten wird möglich, der am Arbeitsplatz mit gutem Grund off Limits, sehr weit off Limits ist. Weiterhin ist ein Schutz der Vertraulichkeit der elektronischen Kommunikation gegen Kriminelle und sogar staatliche Akteure möglich, der im beruflichen Umfeld niemals erreichbar ist, meist dort von der Unternehmensleitung gerade nicht erwünscht ist und häufig genug vom Unternehmen aktiv verhindert wird.

1.4 Sichere elektronische Kommunikation in einer Demokratie

Für Bürger einer freiheitlichen Demokratie (oder Herrschende in einer Diktatur), deren Privatsphäre vom Staat respektiert wird, ergibt sich ein einfaches Bild. Anonymität ist nicht notwendig und wirksame Verschlüsselung kann offen und selbstbewusst eingesetzt werden.

Es muss lediglich auf Basis eines unter voller Kontrolle des Nutzers stehenden Systems ein Werkzeug zur starken Verschlüsselung installiert und fachgerecht verwendet werden. Die heute verbreiteten Verschlüsselungswerkzeuge wurden für diesen Rahmen entwickelt. Die Bewahrung von eventuell bestehender Anonymität spielte bei der Entwicklung keine Rolle. Bei einem Einsatz in feindlicher Umgebung brechen oder gefährden sie die Anonymität der Nutzer.

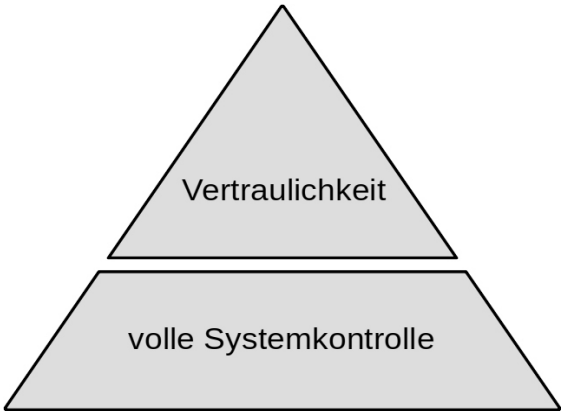

Abbildung 1.1: Pyramide freier Bürger für sichere elektronische Kommunikation

In den westlichen Demokratien haben wir lange geglaubt, das Streben nach der Vervollständigung der in Abb. 1.1 dargestellten Pyramide der Freien reiche aus, d. h., nutze Linux und verschlüssele Mails mit GnuPG, chatte über Jabber-Server.

Leider wird bereits diese simple Vorgehensweise häufig als kompliziert dargestellt, von vielen Bürgern auch so empfunden und deshalb abgelehnt. Selbst unter diesen vereinfachten Bedingungen haben die Mehrzahl der Firmen und Einzelpersonen bereits leichtfertig die Basis der Pyramide vernachlässigt und sich, obwohl nach eigener Einschätzung frei, der Ausspähung durch externe IT-Unternehmen, der staatlichen Dienste der Sitze dieser Unternehmen und unbekannter Datenhändler unterworfen. Sie nutzen proprietäre Betriebssysteme, deren Quellcode der Allgemeinheit gegenüber nicht offengelegt wird und missachten damit die Basis einer sicheren elektronischen Kommunikation.

1.5 Die heutige Situation

Spätestens seit den Snowden-Enthüllungen im Frühsommer 2013 wissen wir, dass wir uns in einem Übergangsprozess von freien Bürgern zu Digitaluntertanen befinden, der schon relativ weit fortgeschritten ist. Leider sind kaum Tendenzen erkennbar, dass dieser Prozess verzögert oder gar

aufgehalten würde. Er schreitet stetig weiter voran und erfordert, dass wir uns als Privatpersonen aufwendig in einem ausgefeilteren Muster als nach der Pyramide der freien Bürger (siehe Abb. 1.1) gegen den Abhörstaat schützen müssen. Nur dann werden wir auch zukünftig als einfache Bürger mehr als nur die Illusion einer sicheren Kommunikation genießen können.

Eine unvollständige Strategie könnte zunächst nur die Verschlüsselung der Kommunikation in den Vordergrund stellen. Laut eigener Aussage bzw. eigenen Drohungen der Nachrichtendienste, die im Nachgang zu den Snowden-Enthüllungen ganz offen ausgesprochen wurden, interessieren sich die Dienste besonders für diejenigen Bürger, die ihre Kommunikation verschlüsseln (deshalb solle man das nicht tun). Naheliegend und den Diensten zuzutrauen wäre es, dieses Interesse durch Verwanzen unserer Computer zu befriedigen. Dies kann ein Dienst recht einfach bewerkstelligen, wenn die Zielperson und deren Internetzugang dem Dienst bekannt ist. Staatliche Nachrichtendienste unterhalten Sammlungen von Sicherheitslücken aller verbreiteten Betriebssysteme, über die sie in diese Systeme Malware einbringen und damit Daten aus- und einschleusen können. Diejenigen, die noch nicht öffentlich bekannt sind, werden Zero Day Exploits genannt. Die neu gegründete Behörde ZITiS soll in Deutschland so die Verwanzbarkeit unserer Systeme durch den Staat sicherstellen. Ein staatlichen Stellen bekannter missliebiger Bürger hat kaum eine Chance, sich gegen solche Angriffe des eigenen Staates zu schützen. Belastbare Anonymität im Internet, bevor der Bürger bei den Behörden als Dissident oder Querulant gebrandmarkt ist, ist der einzige Schutz.

Als Reaktion auf die Snowden-Enthüllungen sind, besonders für Messenger Dienste, verschlüsselte (nicht anonymisierende) Software-Varianten breit in den Markt eingeführt worden und einige werden publikumswirksam von staatlichen Behörden und Diensten wegen deren Verschlüsselung als Ärgernis bezeichnet.

Bei genauem Hinsehen gibt es für mich aber bei allen weitverbreiteten kommerziellen Produkten Anlass zu Misstrauen. Ein bestimmter Messengerdienst namens Signal macht praktisch alles richtig und wird auch von der Mehrzahl der Experten empfohlen: quelloffen, nur standardisierte Verschlüsselungsalgorithmen, Finanzierung durch Spenden und staatliche Zuwendungen, regelmäßige Audits, teilweise reproduzierbarer Build-Prozess, einfach in der Benutzung. Aber genau dieser Messengerdienst läuft zwingend über zentrale Server, ist inkompatibel mit Anonymität und legt die Metadaten gegenüber dem Betreiber des Servers offen. Der Serverbetreiber sichert die vertrauliche Behandlung dieser Daten zu, ist aber Partner in einer Zusammenarbeit mit WhatsApp und unterliegt der US-Jurisdiktion …

Dennoch, wer sich nicht wirklich durch den eigenen Staat oder die US-Administration bedroht fühlt und niemals Anonymität braucht oder die Beschäftigung mit der Sicherheit der eigenen Kommunikation als lästig empfindet, möge solche Dienste nutzen und sich sicher fühlen. Aber auch solche Computernutzer sollten sich ernsthaft zumindest um die Basisstufe in der Kommunikationspyramide der freien Bürger (siehe Abb. 1.1) bemühen.

Dieser Ratgeber ist vor allem für die anderen Menschen gedacht, die die Kontrolle über ihre Daten und über die Sicherheit ihrer elektronischen Kommunikation selbst in der Hand behalten möchten und die auch ein Grundinteresse an den Prinzipien der Anonymisierung und der Verschlüsselung haben. Dieses Buch hat den Zweck, dem Leser nachhaltig und in den Grundzügen produktunabhängig Verständnis über Anonymisierung und Verschlüsselung, insbesondere über deren Zusammenspiel, zu vermitteln und ihn zu guten eigenen Entscheidungen zu befähigen.

1.6 Der Preis digitaler Selbstbestimmung

An dieser Stelle möchte ich betonen, dass Unabhängigkeit und Selbstbestimmung wie immer, so auch in diesem Gebiet, einen Preis haben. Wir müssen für ein Mehr an Autarkie ein Stück weit auf die Bequemlichkeit unseres sonst extrem arbeitsteiligen Lebens verzichten und uns auch in ungeliebten Tätigkeiten üben. Dass dies im analogen Leben gilt, ist eine Binsenweisheit. Wer im Analogen das Rebellenleben von Robin Hood im Sherwood Forest führen will, muss neben

Bogenschießen auch Knöpfe annähen und Unterhosen waschen können. Genauso muss, wer sich als digitaler Rebell sehen möchte, selber Workarounds für Macken des Computers finden, das System zur Not über die Konsole[5] administrieren und in gewissem Maße auch selber Systemfehler bereinigen können. Wer diese Tätigkeiten auf Dauer an Personen delegieren muss, denen er (oder sie) nicht vertrauen kann, verliert die Kontrolle über sein System und über seine Daten.

Für mich ist der in der Vorübung VSx1 auf Seite 35 nur kurz angetippte Konsolenbetrieb das ungeliebte digitale Pendant zum analogen Waschen von Unterhosen im Sherwood Forest. Ich versichere Ihnen aber, dass ich alles darangesetzt habe, in diesem Buch den Anteil solchen Gefummels auf das kleinstmögliche Maß zu beschränken. Ein in Computerdingen kundiger vertrauenswürdiger Freund, der nicht nur Arbeiten abnimmt, sondern auch anleitet, aber Ihnen nicht einfach seine Meinung überstülpt, könnte Ihnen hier eine sehr große Hilfe sein. Überschätzen Sie aber nicht die Kenntnisse Ihres Computer-Wizard-Freundes in Kryptografie. Belastbare Kenntnisse in diesem Gebiet sind rar gesät.

Aus eigener Kraft kundig zu recherchieren, die Qualität von Quellen einzuschätzen und die Ergebnisse am eigenen System umzusetzen, ist dagegen eine unverzichtbare Schlüsselkompetenz. Ich werde mich deshalb im Ablauf der Übungen dieses Buches zum Ende hin immer mehr von der Angabe detaillierter Schritt-für-Schritt-Anleitungen zurückziehen und den Übenden mehr und mehr zu eigenständigem Recherchieren und Umsetzen animieren.

Wem eine Liste heute empfohlener und heute zu vermeidender Anwendungen und Apps wichtig ist und wer ohne allzu tiefe Suche nach Verständnis viele Datenlecks schnell schließen möchte, dem ist sicherlich mit dem exzellenten Ratgeber zur "Digitalen Selbstverteidigung"[6] vom Schweizer Pendant des CCC schneller geholfen als mit diesem Buch. Weiterhin gibt es den noch schneller nutzbaren, aber etwas oberflächlicheren Ratgeber[7] der US-amerikanischen Electronic Frontier Foundation EFF[8]. Sie werden damit allerdings keinen vollen Schutz und kein Verständnis erreichen. Manche der dort empfohlenen Services verweigern sogar den Dienst, wenn Sie sie über Tor[9] anonymisiert nutzen wollen. Der Leitfaden der EFF ist sehr gut, doch für meinen europäischen Geschmack verlässt er sich ein wenig zu sehr auf US-Unternehmen, auch wenn Alternativen aus dem informellen Sektor vorhanden sind. Auf jeden Fall ist die Open-Source-Eigenschaft eines nichtkommerziellen Verschlüsselungswerkzeugs, das der Benutzer leicht aus dem heruntergeladenen Quellcode kompilieren und installieren kann, beruhigender als das Open-Source-Etikett auf einem Android Signal-Client, der über den Google-Playstore aus der Binärdatei installiert wird. Mit europäischen Augen sieht man US-Organisationen misstrauischer, als es US-Bürger tun.

1.7 U-Boot-Kommunikation

Die ideale Kommunikation, zu der mit diesem Ratgeber angeleitet werden soll, lässt sich gut mit einem Bild aus der Seefahrt veranschaulichen. Ein mächtiger Gegner (ein staatlicher Nachrichtendienst, kurz der "Datenkrake") überwacht die Oberfläche des Meeres (das Internet). Ein ungekennzeichnetes U-Boot taucht auf (Ihr Computer geht anonym online), setzt eine Boje mit einer perfekt verschlüsselten Nachricht aus und taucht wieder ab (geht offline). Ein nicht gekennzeichnetes U-Boot taucht eine gewisse Zeit später auf (der Computer Ihres Partners geht anonym online), nimmt die Boje mit der Nachricht auf und taucht wieder ab.

Der mächtige Gegner kann - wenn er sie rechtzeitig findet - die Boje untersuchen, beschädigen oder entfernen und mit der Entfernung seine Entdeckung und die Überwachung offenlegen. Er kann

[5] siehe den entsprechenden Eintrag im Glossar zu Konsole/Terminal auf Seite 188

[6] https://www.digitale-gesellschaft.ch/ratgeber/

[7] https://ssd.eff.org/

[8] https://www.eff.org/

[9] Tor ist das derzeit leistungsfähigste Anonymisierungsnetzwerk für Internetnutzer und wird in diesem Buch erst später in Abschnitt 3.3.3 auf Seite 56 detailliert behandelt.

aber weder erkennen, nach welchem Verfahren die auf der Boje befindliche Nachricht verschlüsselt wurde, noch ob es sich überhaupt um ein Chiffrat (eine verschlüsselte Nachricht) handelt. Er weiß nicht, wer die Nachricht gesendet hat, kennt den Inhalt der Nachricht nicht und hat keine Information darüber, wer die Nachricht empfangen hat oder empfangen soll. Er hat einzig die Information, dass vermutlich an ihm vorbei kommuniziert wurde oder werden soll.

Ich verfolge mit diesem Ratgeber das Ziel, eine solche perfekt geschützte Kommunikation zu erreichen, sozusagen mit Ihnen zusammen den für intelligente Laien erreichbaren Gipfel der sicheren elektronischen Kommunikation zu erklimmen. Wenn Sie dann später Ihre eigene Alltagskommunikation absichern wollen, spielt natürlich auch der Komfort eine Rolle und Sie werden im Alltag für sich selbst Ihren persönlichen Kompromiss zwischen Komfort und Sicherheit, je nach Intensität der Bedrohung, finden können. Bei Bedarf sollen Sie aber wissen, wie es wieder ganz nach oben geht; Sie waren schon mal da und kennen den Weg.

Ich sehe auf dem Weg zu einer dem Seefahrtsbild analogen, perfekt geschützten elektronischen Kommunikation vier abgrenzbare Fertigkeitsstufen, die aufeinander aufbauen und jeweils die Beherrschung der darunter liegenden Fertigkeitsstufen voraussetzen. Diese vier Stufen sind in Abb. 1.2 in Form einer Pyramide illustriert.

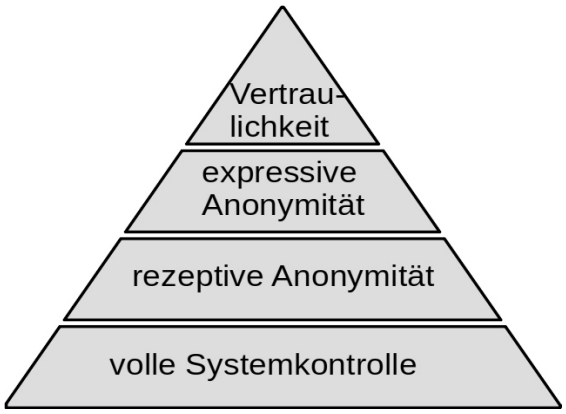

Abbildung 1.2: Pyramide der sicheren elektronischen Kommunikation, gesichert gegen einen Abhörstaat. Kurz hier auch "Pyramide des Dissidentenschutzes" genannt.

Im Folgenden geht es in diesem Ratgeber um das Streben nach Vervollständigung dieser Pyramide der gegen den Abhörstaat gesicherten elektronischen Kommunikation. Ich nenne diese Pyramide auch die "Pyramide des Dissidentenschutzes".

1.8 Training für digitale Tarnung

Wir werden die in Abb. 1.2 gezeigte Pyramide des Dissidentenschutzes systematisch Stufe für Stufe aufbauen, von der vollen Systemherrschaft an der Basis über zwei Ausprägungen der Anonymität bis hin zur Vertraulichkeit an der Spitze. Im Gegensatz zur verbreiteten Lehrmeinung behaupte ich, dass verlässliche Anonymität eine notwendige Voraussetzung für eine robuste Privatsphäre ist. Mittel zum Erzwingen der Privatsphäre, wie etwa Verschlüsselungswerkzeuge, dürfen die zugrunde liegende Anonymität nicht gefährden oder brechen.

Leider sind die meisten der heute verfügbaren Verschlüsselungswerkzeuge und alle Werkzeuge, bei denen die Verschlüsselung und einfacher Datentransfer über vorkonfigurierte Serververbindungen als festes Bündel vorliegt, mit der Anonymität nicht kompatibel (z. B. Messenger wie Signal oder WhatsApp). Bedauerlicherweise bezeichnen führende Sicherheitsexperten in den USA solche gebündelten Werkzeuge als modern und überlegen gegenüber flexiblen, selbstbestimmten Verfah-

ren[10]. Bis zu einem gewissen Grad wird diese Meinung sogar von der Organisation EFF vertreten, die sonst sehr hilfreiche Ratschläge für das Wahren der Privatsphäre gibt. Die "überlegenen Bündel" schützen jedoch nicht vor unserer jeweiligen Regierung, zumindest nicht vor der Regierung der USA.

In der im Winter 2020/21 wieder einmal neu aufgelegten Diskussion um ein Verbot der sicheren Verschlüsselung - diesmal in der EU und mit dem Blick auf Messenger Dienste - tut sich das deutsche Innenministerium mit zahlreichen Vorschlägen und Anregungen hervor, die meist auf eine eher mittelmäßige Kompetenz in Kryptologie schließen lassen, um es vorsichtig auszudrücken. Allerdings zeigte sich an einer Verlautbarung, dass man im BMI eine Schwachstelle der "überlegenen Bündel" (= Messenger Dienste) sachlich richtig ausgemacht hat. In dem in der Fußnote[11] referenzierten Artikel des Nachrichtenportals heise.de heißt es:

Möglich sei ein Zugriff auch, wenn der Netzbetreiber "bei der Erzeugung oder dem Austausch von Schlüsseln mitwirkt und ihm dadurch die Entschlüsselung der Telekommunikation möglich ist", konstatiert das Ministerium weiter.

Dem muss man nichts hinzufügen. Es ist für die Sicherheit entscheidend, dass der Nutzer von Verschlüsselung die Erzeugung, Verteilung, Nutzung und Verwaltung der Schlüssel unter ureigener und manueller Kontrolle hat.

Die Fähigkeit zur sicheren elektronischen Kommunikation ist nur durch einen systematischen Lernprozess zu erreichen, der nicht durch einfaches Durchlesen eines Ratgebers absolviert werden kann. Das geht so wenig, wie man durch Durchlesen eines Buches über das Klavier das Klavierspielen erlernen kann. Umfangreiche Übungsarbeit ist unerlässlich. In diesem Leitfaden behandle ich jede der vier Ebenen der Pyramide des Dissidentenschutzes, indem ich zunächst den theoretischen Hintergrund darlege. Dann werfen wir einen Blick auf einige kostenfreie Open-Source-Werkzeuge, die es uns jeweils ermöglichen, die auf der entsprechenden Ebene gesetzten Ziele zu erreichen. Danach stelle ich eine Auswahl von Übungen vor, die Ihnen praktische Erfahrung vermitteln. Das noble Vorhaben, die Dinge wirklich zu tun, anstatt nur darüber zu plappern, mag allerdings umständlich und anstrengend erscheinen. Aber jeder, der behauptet, Sie könnten mühelos Sicherheit Ihrer Online-Kommunikation erreichen, lügt. Auf jeden Fall eröffnet Ihnen jede einzelne der vier erklommenen Stufen neue Möglichkeiten, sich der Überwachung zu entziehen und der digitalen Unterdrückung zu widerstehen. Im Übrigen stärkt das Erreichen jedes Niveaus auch Ihre Widerstandskraft gegen die verbreitete Internet-Kriminalität. Im abschließenden Abschnitt zu jeder der vier Leistungsstufen werde ich Hinweise auf die wichtigsten Angriffsmodi und die bedrohlichsten Schwachstellen geben.

In den Anleitungen dieses Buches beschränke ich mich auf Werkzeuge für die Desktopsysteme Linux und Windows. Viele von diesen Werkzeugen könnten aber auch auf Apple Computern genutzt werden. Bezüglich Smartphonesystemen besitze ich bei Weitem nicht genug Erfahrung, um zur Basis der Pyramide, der IT-Selbstbestimmung, auf diesen Systemen anleiten zu können. Diese werden in diesem Ratgeber deshalb nicht von mir behandelt.

[10] Ein solches flexibles Verfahren könnte etwa darin bestehen, GnuPG-Verschlüsselung frei mit selbst gewählten Transportmitteln zu kombinieren.

[11] Meldung vom 9.12.2020 -> https://www.heise.de/news/Crypto-Wars-Bundesregierung-verteidigt-EU-Linie-zur-Entschluesselung-4984108.html

2. Systemherrschaft auf Ihrem Computer

2.1 Üben Sie volle Kontrolle über Ihr System aus

Ein System steht unter der vollen Kontrolle des Nutzers, wenn folgende Kriterien erfüllt sind:

1. Der Nutzer kann den Quellcode des Betriebssystems selbst einsehen oder durch eine Vertrauensperson seiner Wahl prüfen lassen.
2. Es können keine Änderungen am Zustand des Systems ohne die ausdrückliche Zustimmung des Nutzers vorgenommen werden.
3. Der Nutzer kann jede von ihm gewünschte Änderung am System vornehmen.

Zu Punkt 1

Proprietäre, nicht quelloffene Betriebssysteme wie Windows oder Apples macOS verletzen dieses Kriterium. Wegen der weiten Verbreitung von Windows unter privaten Nutzern werde ich diesen Ratgeber dennoch auch für Windows vorsehen. Der Windows-Nutzer sollte sich aber darüber im Klaren sein, dass er auf diesem System zwar grundsätzlich die Techniken zur "U-Boot-Kommunikation" erlernen kann, er aber zumindest gegen Microsoft und US-Dienste niemals belastbare Sicherheit erreichen wird. Der Windows-Nutzer kann also anhand dieses Ratgebers die höheren Stufen der Pyramide für sich ausloten, kann aber letztlich nur die ganze Pyramide aufbauen, wenn er (oder sie) schließlich zur Basis der Pyramide zurückkehrt und zu Linux oder einem anderen quelloffenen Betriebssystem migriert.

Solange der Nutzer keine Möglichkeit hat, den Quellcode des Betriebssystems einzusehen oder unabhängig vom Hersteller einsehen zu lassen, kann der Hersteller des Betriebssystems grundsätzlich, auch verdeckt, nach Gutdünken mit dem Nutzer verfahren. So könnte der Hersteller Zugangskanäle eingerichtet haben, mit denen er beliebige, z. B. sensible persönliche Daten aus dem System abfließen lassen oder z. B. unerwünschte Überwachungsprogramme oder kompromittierende Dateien in das System einbringen kann. Man sollte davon ausgehen, dass solche Zugänge tatsächlich vorhanden sind.

Vorsicht, selbst wenn das Betriebssystem quelloffen ist, gibt es in vielen im Rechner verbauten

Chips lokale Chipsoftware, die sogenannte Firmware, deren Quellcode meist nicht offen liegt. Der eingangs vorgestellte Fehler "ROCA", den wir in Abschnitt 1.3 ausführlich behandelt haben, liegt in solcher Firmware des TPM vor. Auch die Firmware kann herstellerseitig oder durch Hackerangriffe bösartig manipuliert sein. Allerdings sind Angriffe auf die Firmware ohne Beihilfe und Wissen des Chipherstellers deutlich schwieriger auszuführen als auf das Betriebssystem des Rechners. In die Firmware eingebrachte Malware ist aber sehr schwer zu entdecken und wäre auch durch Zurücksetzen des Betriebssystems in den Auslieferungszustand nicht zu entfernen. Dies macht dort angesiedelte Attacken besonders attraktiv für mächtige staatliche Angreifer.

Zu Punkt 2

Regelmäßige Systemupdates sind aus Sicherheitsgründen unverzichtbar, um jeweils neu erkannte Sicherheitslücken zu schließen oder andere Softwarefehler zu heilen. Aus Gründen der Bequemlichkeit wünschen sich viele private Nutzer, dass sie damit nicht behelligt werden und dieses automatisch weitestgehend im Hintergrund abläuft. (Wir haben diesen Punkt bereits in Abschnitt 1.2 berührt.) Die Hersteller proprietärer Betriebssysteme gehen davon aus, dass alle privaten Nutzer das so wünschen, setzen dies in der Regel so um und veranlassen solche Aktionen ohne ausdrückliche Zustimmung des Nutzers.

Aus Sicherheitsgründen ist diese Vorgehensweise aber entschieden abzulehnen, weil damit ein Kontrollverlust einhergeht. Nur dann, wenn man der Benevolenz und der Kompetenz des Betriebssystem-Herstellers für jetzt und in alle Zukunft absolut vertrauen würde, könnte die Hinnahme dieses Kontrollverlustes einigermaßen gerechtfertigt werden. Man kann aber als selbstverständlich voraussetzen, dass man niemals in einer Kunde/Verkäufer-Beziehung dem Verkäufer absolut und für alle Zeiten bedingungslos vertrauen kann. Customer Lock-In ist das Mindeste, auf das man sich bei so einem nachlässigen Verhalten einstellen kann. Weiterhin geht aus den Snowden-Enthüllungen hervor, dass alle großen Softwarekonzerne bereitwillig mit US-Nachrichtendiensten kooperiert hatten. Man muss also damit rechnen, dass man nach solch einem Kontrollverlust sein System dem Zugriff durch staatliche Dienste widerstandslos ausliefert.

Seriös konfigurierte Betriebssysteme erzwingen in Standardkonfiguration, dass jedes Updatepaket ausdrücklich vom Nutzer zur Installation freigegeben werden muss und informieren den Nutzer bei der Nachfrage verständlich über den Zweck und die Art des angebotenen Updates. Linux-Desktop-Systeme in der Standard-Einstellung tun das in aller Regel in vorbildlicher Weise.

Zu Punkt 3

Der Nutzer muss jede gewünschte Software, die auf dem gegebenen System lauffähig ist, auch installieren und betreiben können. Es ist nicht Sache eines Softwarekonzerns zu entscheiden, was auf dem System des privaten Nutzers und Besitzers installiert werden darf und was nicht. Ein bekanntes Betriebssystem treibt diese inakzeptable Restriktion auf die Spitze und zwingt den Nutzer zu allerhand schmutzigen Tricks, um diesem sogenannten "Walled Garden", dem ummauerten Garten, zu entkommen. Sinnigerweise werden diese Tricks "Jailbreak" genannt - der Gefängnisausbruch.

Ebenso muss der Nutzer nicht benötigte und unerwünschte Software vom System sauber und einfach entfernen können. Die Soft- und Hardwarehersteller tendieren dazu, Systeme für Privatnutzer vor dem Verkauf mit allerhand sogenannter "Crapware" zu bestücken. Sie tun dies zum Eigenmarketing oder schlicht gegen Bezahlung. Aus Nutzerperspektive ist Crapware die Menge unerwünschter Programme, die er nur mehr oder weniger schwer vom System entfernen kann. Eine nur kurze Zeit lauffähige Version von Microsoft Office auf Windows-PCs ist noch eine der milderen Varianten dieser Unsitte und kann mit einiger Mühe auch durch Nutzer mittlerer Expertise fast restlos entfernt werden. Besonders die mobilen Systeme werden häufig mit einer überbordenden

Menge an Crapware ausgeliefert, die ohne einen Jailbreak (Apple) oder ohne Rooting[1] (Android) gar nicht zu entfernen ist. Sogar ein Herausgeber einer Linux-Distribution hat dieses inakzeptable Verfahren entdeckt, Einnahmen zu generieren: Ich habe selbst schon in Ubuntu (eine populäre Linux-Distribution) einen mit großer Penetranz wiederholt auftauchenden "Music Store" aus meinem Rechner entfernen müssen. In solchen Maßnahmen zeigt sich die Respektlosigkeit der Softwarekonzerne gegenüber ihren privaten Kunden, die auch bestimmte Rückschlüsse auf die Intensität des Eintretens für den Schutz der Kundendaten gegenüber staatlichen Stellen nahelegt. Seien Sie auf der Hut im Umgang mit Softwarekonzernen, als deren Kunde Sie sich fühlen möchten!

Der Nutzer und Eigentümer eines Gerätes muss unerwünschten Datenverkehr mit dem Hersteller des Betriebssystems unterbinden können. Besonders seit der Einführung von Windows 10 ist dies für dieses Betriebssystem nicht mehr gewährleistet. In Situationen, die das aus Sicherheitsgründen unbedingt erfordern, kann der Nutzer dies dennoch durch hartes Abschalten des Netzwerk-Interface oder, etwas weniger rustikal, durch Einschalten des sogenannten "Airplane Modus" erreichen.

2.2 Sichere Installation von Software

Das Recht, beliebige Änderungen am System vorzunehmen, ist nur ein Teil der benötigten Ressource. Man muss die Änderungen, in diesem Fall die Installation neuer Software, auch sicher vornehmen können.

Neue Software oder Softwareupdates kommen heute üblicherweise über das Internet zu Ihrem Computer. Sie sind deshalb während der Reise durch das Netz durch Einsicht und ggf. Manipulation durch Kriminelle, die Marketingmafia, staatliche Nachrichtendienste und andere digitale Wegelagerer bedroht. Manipulation solcher übertragenen Daten ist ein klassischer Weg, gegen den Willen des privaten Nutzers offen oder verdeckt die Kontrolle über den Rechner des Nutzers zu erlangen. Auch die deutsche Bundesregierung hat sich entschlossen, solche digitale Wegelagerei betreiben zu wollen und dafür am 6. April 2017 die Behörde ZITiS[2] gegründet.

Manipulationsschutz von Dateien im Transfer mussten die Hersteller von Betriebssystemen (Windows, macOS, Linux, ...) schon lange für ihre Systemupdates genauso gewährleisten, wie ihn jetzt der private Computernutzer gewährleisten muss, der als Administrator seines Systems nach eigener Entscheidung Software beziehen und installieren will.

Im "ganz normalen PC" liegen für diesen Zweck bereits kryptografische Sicherungen der Softwareindustrie vor (z. B. Authenticode[3] von Microsoft), die aber für den Benutzer verdeckt oder mit sehr sparsamen Informations- und Eingriffsmöglichkeiten meist automatisch im Hintergrund arbeiten. Diese Sicherungen sollen gegen gewöhnliche Kriminelle schützen, dienen aber auch den Interessen von Wirtschaft, Regierungen und leider wohl auch den Interessen von Geheimdiensten. Sie sind bei kommerziellen Betriebssystemen der Kontrolle der privaten Nutzer faktisch entzogen. Deshalb muss für selbstbestimmten Umgang mit solchen kryptografischen Sicherungen ein System unter ureigener manueller Kontrolle des Nutzers neu verankert und aufgebaut werden. Bei Linux-Desktop-Systemen ist standardmäßig ein für den privaten Nutzer manuell nutzbares, quelloffenes kryptografisches Sicherungswerkzeug vorhanden (GnuPG[4]).

Ich werde im folgenden Text, besonders wenn kryptografische Eigensicherung noch nicht vorhanden ist oder von einem Softwareherausgeber nicht unterstützt wird, die Verwendung von HTTPS[5] und der kommerziellen Public-Key-Infrastruktur (PKI)[6] empfehlen, wie Sie in gängigen Browsern unterstützt wird. Diese bieten eine gewisse Sicherheit.

[1] Rooting eines Mobilgerätes ist das Hacken durch den Nutzer, um sich Kontrolle über das Gerät zu verschaffen.
[2] https://www.zitis.bund.de/DE/Home/home_node.html
[3] siehe: https://docs.microsoft.com/en-us/windows-hardware/drivers/install/authenticode
[4] https://de.wikipedia.org/wiki/GNU_Privacy_Guard
[5] siehe: https://de.wikipedia.org/wiki/Https
[6] siehe: https://de.wikipedia.org/wiki/Public-Key-Infrastruktur oder den Eintrag im Glossar.

Aber Vorsicht, HTTPS bietet Sicherheit gegen Feld-, Wald- und Wiesenkriminelle, wird aber von kompetenten Akteuren, wie z. B. auch staatlichen Nachrichtendiensten, regelmäßig unterlaufen. In einer geleakten Dokumentensammlung der CIA namens Vault7[7] warnt ein CIA-Experte die eigenen Agenten, sich niemals auf die Sicherheit von HTTPS zu verlassen. In vielen Staaten und in vielen Firmen würde HTTPS-Sicherung routinemäßig ausgehebelt. Beim Ausschleusen von Daten aus erfolgreich infizierten Zielcomputern sollten die allgemeinen Standardsicherungen und automatisierten Verschlüsselungen lediglich als "blending layer"[8] betrachtet werden, also sozusagen als Dekoration. Wirksame Verschlüsselung müsse man (zusätzlich) selber vornehmen. Ich hätte das nicht schöner formulieren können.

2.3 Einsatz der digitalen Signatur

Abbildung 2.1: Paar aus privatem und öffentlichem Schlüssel
Links: privater Schlüssel. Muss geheim gehalten werden. Dient zum Entschlüsseln von Chiffraten und zum Signieren von Dokumenten.
Rechts: öffentlicher Schlüssel. Muss gegen Manipulation geschützt veröffentlicht werden. Dient zum personalisierten Verschlüsseln und zum Verifizieren der Signatur.

An dieser Stelle sollen ganz kurz die Grundprinzipien der asymmetrischen Kryptografie skizziert und die digitale Signatur holzschnittartig erläutert werden. Das Grundprinzip ist, entgegen dem landläufigen Vorurteil, recht einfach. Allerdings wird, durch kommerzielle Interessen geleitet, häufig daraus ein kaum durchschaubares Gestrüpp aufgebaut und im Markt etabliert. Wer sich einen Eindruck von diesem Labyrinth verschaffen will, der möge sich nach den in den nächsten Absätzen folgenden kurzen Bemerkungen über die digitale Signatur im Internet über Microsoft's "Authenticode"-Verfahren informieren.

In der asymmetrischen Verschlüsselung hat ein Nutzer ein Schlüsselpaar (siehe Abb. 2.1). Das Paar besteht aus einem privaten Schlüssel und einem zugehörigen öffentlichen Schlüssel. Es ist rechnerisch einfach, den öffentlichen Schlüssel aus dem privaten Schlüssel zu bestimmen. Es ist jedoch rechnerisch nahezu unmöglich, den privaten Schlüssel aus dem öffentlichen Schlüssel oder anderen Daten zu berechnen, die der Besitzer des Schlüssels öffentlich preisgibt. Der Besitzer muss den privaten Schlüssel vor fremdem Einblick verbergen und den entsprechenden öffentlichen Schlüssel verbreiten. Er macht ihn seinen Kommunikationspartnern fälschungssicher zugänglich. Eine geeignete Art der Verbreitung wäre etwa die Veröffentlichung auf einer HTTPS-gesicherten Webseite, auf die nur der Eigentümer des Schlüssels Schreibzugriff hat.

Mit dem öffentlichen Schlüssel eines Kommunikationspartners kann man gezielt für den Partner eine Datei so verschlüsseln, dass ausschließlich der Partner sie mit seinem privaten Schlüssel wieder entschlüsseln kann.

Mit Ihrem eigenen privaten Schlüssel können Sie eine gültige digitale Signatur einer Datei erstellen. Dies ist ohne Zugriff auf den privaten Schlüssel nicht möglich. Jeder, der den entsprechenden öffentlichen Schlüssel kennt, kann die Gültigkeit des Tripels aus Datei, digitaler Signatur und öffentlichem Schlüssel überprüfen (siehe Abb. 2.2).

Im Fall eines eigengesicherten Software-Updates verfügt man über den öffentlichen Schlüssel des Herausgebers und hat das Update und die Signatur des Updates über das Netz heruntergeladen.

[7] siehe: https://en.wikipedia.org/wiki/Vault_7

[8] siehe z. B.: https://wikileaks.org/ciav7p1/cms/page_14587109.html und suchen Sie in dieser Seite nach den Worten "blending layer".

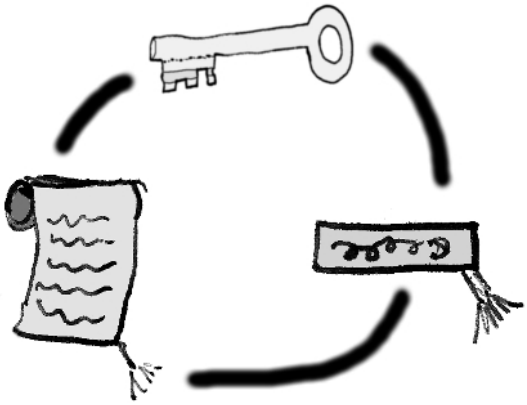

Abbildung 2.2: Das Tripel öffentlicher Schlüssel, Datei und digitale Signatur kann gültig sein (die Signatur ist korrekt) oder ungültig. Datei und digitale Signatur werden zusammen zum Verifizierer transportiert. Der öffentliche Schlüssel des Herausgebers sollte vom Verifizierer verantwortungsvoll von selbst gewählter Quelle, zu selbst bestimmter Zeit und auf selbst bestimmtem Weg beschafft worden sein.

Wenn die Prüfung von Signatur und Update-Datei gegen den öffentlichen Schlüssel erfolgreich verläuft, kann man sicher sein, dass der Inhaber des zugehörigen privaten Schlüssels das Update exakt in dem Zustand signiert hat, wie es jetzt beim Nutzer vorliegt. Auf jedes einzelne Bit genau.

Manipulation durch digitale Wegelagerer, etwa die Umwandlung in einen Trojaner bei der Reise durch das Internet, kann dann ausgeschlossen werden. Bei erfolgreich geprüfter Signatur kann die heruntergeladene Software installiert werden, falls dem Herausgeber der Software selbst vertraut werden kann.

Nach der Beschreibung dieses einfachen manuellen Verfahrens, das vom Benutzer gesteuert wird, möchte ich nun auf den Windows-Automatismus Authenticode eingehen, bei dem die Nutzerbeteiligung vermieden wird.

Bei Microsofts Authenticode-Automatismus ist der Installationsdatei eine digitale Signatur des Herausgebers angeheftet sowie ein X509-artiges "Zertifikat", das den durch eine "Certification Authority" signierten öffentlichen Schlüssel des Herausgebers enthält, dessen Signatur seinerseits wiederum zuerst automatisch geprüft wurde. Dabei ist der öffentliche Schlüssel der Certification Authority in der Datei "certmgr.msc" im System hinterlegt. Bei positivem Ausgang der Prüfung wird automatisch die eigentliche Signatur des Herausgebers gegen die Installationsdatei geprüft. Parallel fragt das System über das Internet bei einer externen Datenbank ab, ob das Zertifikat inzwischen zurückgerufen wurde... Sie bekommen ein Bild des letztlich durch Ausschluss des Nutzers begründeten Gestrüpps.

Bitte beachten Sie, dass ein häufig von Herausgebern freier Software zur Installationsdatei mitgelieferter Hashwert[9] wenig Sicherheitsrelevanz hat und eine digitale Signatur des Herausgebers nicht ersetzen kann. Jeder digitale Wegelagerer könnte zu einer von ihm manipulierten Installationsdatei einen Hashwert berechnen und angeben. Leider hat sich diese Erkenntnis bei einigen Herausgebern freier Software immer noch nicht eingestellt. Deshalb muss gelegentlich bei fehlender Signatur einer Installationsdatei versucht werden, mit etwas umständlichen Ersatzverfahren das sicherheitstechnisch Bestmögliche aus einer angegebenen Checksumme, d. h. aus einem angegebenen Hashwert, zu machen.

[9] Siehe Glossareintrag zum Begriff Hashwert.

2.4 Werkzeuge für Systemherrschaft auf Ihrem Computer

Wie auch in den folgenden Abschnitten zu den anderen Schichten der Pyramide des Dissidentenschutzes (siehe Abb. 1.2) wird im Folgenden quelloffene und freie Software als Werkzeug vorgestellt, mit dem wir das Ziel der aktuellen Schicht erreichen können. An dieser Stelle betrachten wir Software, mit deren Hilfe wir dem Ziel der vollen Systemherrschaft nahekommen können.

2.4.1 Quelloffene Betriebssysteme

Das am weitesten verbreitete freie und quelloffene Betriebssystem für private Computer ist Linux. Ich möchte deshalb hier nur auf Linux-Varianten eingehen.

Es gibt viele freie Linux-Distributionen für private Desktops, Notebooks oder Netbooks. Diese Distributionen enthalten jeweils den Linux-Kernel und eine Sammlung von Software-Modulen, die für ein arbeitsbereites System benötigt werden, wie z. B. einen Mailer, einen Browser, ein Office-Paket, einen Desktop-Organizer und so weiter. Darüber hinaus bieten die Distributionen Zugang zu einem sogenannten Repository, das ein Universum aller Arten von freier Software enthält, die Sie mit einem einzigen Mausklick zum Herunterladen und zur sicheren Installation auswählen können. Sie können Quellen für diese Linux-Distributionen leicht mit Ihrer bevorzugten Suchmaschine finden. Es ist sicher eine freie Distribution dabei, die Ihren Wünschen für ein Desktopsystem entspricht. Sie sollten für jetzt oder später eine Migration zu einem dieser Systeme erwägen, falls Sie noch mit Windows oder einem anderen proprietären Betriebssystem arbeiten. Hier sind, ohne Anspruch auf Vollständigkeit, einige aufgelistet: Ubuntu, Kubuntu, Edubuntu, Lubuntu, Xubuntu, Debian, Mint, Knoppix, Trisquel, Elementary OS, Fedora, Mandriva, openSUSE, Raspbian ...

Sie finden Beschreibungen und Links leicht in Wikipedia[10] oder anders sortiert in der englischsprachigen Wikipedia[11] [12]. Ich persönlich mag Lubuntu[13] und Xubuntu[14], weil diese Distributionen der Windows-Version, mit der ich ursprünglich vor vielen Jahren auf dem Computer digital gehen gelernt habe, in der Bedienung immer noch recht ähnlich sind. Wenn Sie als Windows-Nutzer viel mit WindowsXP gearbeitet haben und - wie ich - das Wischen auf bunten Brettchen verabscheuen, könnten Ihnen diese Distributionen ebenfalls liegen.

Für alle neueren Computer sollten Sie die 64 Bit Version des Betriebssystems auswählen. Hardware, die mehr als eine Handvoll Jahre alt ist, braucht möglicherweise noch eine 32 Bit Version.

Sie können für die jeweilige Linux-Distribution eine Live-CD/DVD oder einen Live-USB-Stick erzeugen. Diese Live-Medien erlauben Ihnen, an Ihrem System auszuprobieren, ob auf Ihrer Hardware wirklich alles auf Anhieb problemlos läuft und ob Ihnen die jeweilige Standard-Oberfläche der Linux-Distribution gefällt. Sie erlauben dies an der Festplatte Ihres Computers vorbei, ohne dass Sie Änderungen an Ihrem System vornehmen müssen. Die Download-Seiten der jeweiligen Distributionen bieten in der Regel hervorragende Schritt-für-Schritt-Anleitungen dazu, wie Sie solche Live-Medien vorbereiten und nutzen können. Wenn Ihnen die vom Live-Medium aus betriebene Linux-Distribution gefällt, können Sie die Installation auf dem primären Speichermedium Ihres Computers mit einem Mausklick aus dem Live-System heraus starten.

[10] siehe: https://de.wikipedia.org/wiki/Linux-Distributionen
[11] siehe: https://en.wikipedia.org/wiki/Linux_distribution
[12] siehe: https://en.wikipedia.org/wiki/Lightweight_Linux_distribution
[13] https://lubuntu.net/
[14] https://xubuntu.org/getxubuntu/

2.4.2 Manueller Umgang mit digitalen Signaturen: GnuPG

Signaturen nach dem OpenPGP-Standard[15] sind die verbreitetste Form von durch den Nutzer frei handhabbaren digitalen Signaturen. Deshalb werden Sie die freie Software GnuPG benötigen, in der dieser Standard implementiert ist. Bei Windows müssen Sie GnuPG installieren, bei Linux-Distributionen ist GnuPG üblicherweise bereits enthalten.

Teil 1, sichere Installation von GnuPG

An dieser Stelle begegnen wir als typischer Windows-Nutzer einem Dilemma. Wir müssen, noch ohne die Möglichkeit zu kryptografischer Eigensicherung, durch digitale Signaturen eine Software zur kryptografischen Eigensicherung möglichst sicher installieren. Ich hadere damit, Ihnen die sowohl für den Autor als auch für den Leser unelegante, verstrickte Beschreibung der Behelfssicherung hier zu präsentieren. Eine digitale Signatur ist, wenn sie gut implementiert wird, einfach, zweifelsfrei sicher und transparent. Das hier beschriebene Behelfsverfahren ist weder einfach noch transparent, und bei der Sicherheit gibt es durchaus Luft nach oben. Ich habe dem Behelfsverfahren hier dennoch viel Raum und Mühe gewidmet, weil es leider noch immer auch an vielen anderen Stellen in der Praxis benötigt wird. Mich versöhnt dabei aber ein wenig, dass Sie als Leser bei der mühsamen Umsetzung des Behelfsverfahrens durchaus Ihr IT-Repertoire erweitern und trainieren werden. Gehen wir es also an!

Gefahren bei unsicherer Softwareinstallation ohne Eigensicherung

Die Installationsdatei könnte bei ihrem Weg durch das Internet von einem Angreifer mit einem Virus oder Trojaner beladen werden. Bei der anschließenden Installation mit dieser Datei würden wir unser System infizieren. Jede Station auf der Datenstrecke, die an der Übertragung beteiligt ist, könnte einen solchen Angriff durchführen. Ein ressourcenstarker Angreifer könnte sogar einen möglicherweise bestehenden HTTPS-Schutz der Daten brechen oder umgehen.

Der Angriff kann wie ein Schrotschuss auf jeden Internetnutzer zielen, der die Installationsdatei herunterlädt, und jeden Download mit Malware vergiften. Oder es könnte sich um einen gezielten Angriff speziell auf Ihre Person und Ihren Computer handeln. Die Gefahr des Schrotschuss-Angriffs ist in unserer gegenwärtigen Situation tragbar. Viele andere Internetnutzer haben bereits die Möglichkeit zu kryptografischer Eigensicherung, die wir erst nach der Installation von GnuPG haben werden. Diese anderen Nutzer würden durch Prüfen der digitalen Signatur der Installationsdatei (siehe Abschnitt 2.3 auf Seite 24) die Manipulation der Datei durch den Angreifer sofort bemerken und den Herausgeber der Software darauf hinweisen. Der Angriff würde auffliegen und Gegenmaßnahmen würden ergriffen werden, die den Angreifer gefährden oder bloßstellen könnten. Spätestens nach ein paar Tagen wäre der Spuk vorbei.

Gefährlicher ist in dieser Situation für uns die zweite Variante, ein gezielter Angriff auf Sie persönlich (oder auf eine kleine Gruppe von Zielpersonen) und damit auf Ihren Computer, der unter Ihrer IP-Adresse auf Internetressourcen zugreift. Der Angreifer würde alle von ihm transportierten Installationsdateien inspizieren und nur dann mit Malware vergiften, wenn Ihre IP-Adresse als Ziel des Datenstromes angegeben ist. Ein solcher Angriff könnte über Monate oder gar Jahre scharfgeschaltet sein, ohne allgemein bemerkt und bekämpft zu werden.

Sie könnten einwenden, dass ein Angreifer kaum wissen kann, unter welcher IP-Adresse Sie im Netz unterwegs sind. Ich versichere Ihnen aber, dass es viele Datenbanken gibt, in denen tagesaktuell Ihre IP-Adresse Ihrem Namen und Ihrer Identität zugeordnet ist. Die Marketing-Mafia treibt Handel mit solchen Daten. Bei jedem Login, etwa bei einem automatisch erfolgenden Google-Login, kennt der Dienstanbieter Ihre Identität aus den Kontendaten und kann diese mit

[15] Der OpenPGP-Standard ist ein von der Internet Engineering Task Force (IETF) gepflegter Standard für Verschlüsselung und digitale Signaturen und kann unter der URL https://www.openpgp.org/about/standard/ eingesehen werden.

der IP-Adresse verknüpfen, über die Sie den Login ausführen[16]. Wenn die Marketing-Mafia Ihnen auf Sie persönlich zugeschnittene Werbung aufdrängen kann, kann Ihnen auch eine feindliche Organisation personalisierte Attacken aufdrängen. Alle benötigten Daten könnte der Angreifer sogar auf dem grauen Werbemarkt kaufen.

Wir wollen nun genau diesen gezielten Angriff auf Sie erschweren, indem wir Ihre Download-Aktivität auf zwei IP-Adressen verteilen, von denen die zweite Ihnen ganz frisch von Ihrem ISP neu zugewiesen worden ist. Eine gezielte Attacke auf Sie müsste dann sehr schnell auf die neue IP-Zuordnung reagieren, um erfolgreich zu sein. Lediglich tagesaktuelle Adressdaten würden dem Angreifer nicht mehr genügen. Es müssten mindestens minutenaktuelle Daten sein. Ein nach Zuweisung der neuen IP-Adresse noch auszuführender Teil Ihres Downloads sollte daher klein sein, damit er schnell erfolgen kann. Deshalb würden Sie im zweiten Teil nicht die große Installationsdatei, sondern nur einen digitalen Fingerabdruck dieser Datei erneut herunterladen, den sogenannten Hashwert (siehe auch den entsprechenden Eintrag im Glossar). Dies verursacht eine viel kleinere "Bugwelle" im Datenozean des Internets und ist damit weniger deutlich sichtbar für Überwacher und Angreifer als ein erneuter Download der großen Installationsdatei.

Sie haben an dieser Stelle des Kurses also nun die Option, bewusst die Gefahr eines auf Sie persönlich gezielten Angriffes zu tolerieren, ein kurzes Stoßgebet gen Himmel zu schicken und "einfach so" die Installationsdatei herunterzuladen und ohne Prüfung auszuführen. Das wäre die einfachste Vorgehensweise und könnte bei momentan noch moderatem Ehrgeiz und Fertigkeitsniveau auf Ihrer Seite sogar vernünftig sein. In diesem Fall bitte ich Sie aber, die folgenden Ausführungen trotzdem aufmerksam zu verfolgen und die dabei benötigten Fertigkeiten wenigstens teilweise einzuüben (Hashberechnung und Handhabung der eigenen IP-Adresse) und nur für Sie sehr schwierige Teile zu überspringen. Die andere Option ist, die Ärmel hochzukrempeln, Konzentration zu sammeln und die zugegebenermaßen komplexe, behelfsmäßig gesicherte Installation anzugehen.

Vorbereiten des Systems für die Behelfssicherung

Im letzten Abschnitt habe ich den Einsatz von Hashwerten bereits angesprochen. Vielleicht verfügen Sie bereits über ein Werkzeug zur Bestimmung der Hashwerte von Dateien. Dann wäre die behelfsmäßig geschützte Installation möglich. Einen Hashwert kann man sich als digitalen Fingerabdruck einer Datei vorstellen. Im Gegensatz zu einem echten Fingerabdruck ändert jedoch die kleinste Änderung an der Datei, wie z. B. ein einziger Bit-Flip an einer beliebigen Stelle, den Hashwert in eine völlig andere Zahl, die keine Ähnlichkeit mit dem vorherigen Wert aufweist.

Vermutlich steht auf Ihrem System bereits Software für die Berechnung von Hashwerten zur Verfügung. Wenn Sie mit Linux arbeiten, können Sie in einer Konsole (siehe auf Seite 188 im Glossar und Abschnitt 2.5.1 auf Seite 35) das Kommando:

`sha256sum <pfad zur datei> > checksumme.txt`

eingeben. Die Textdatei "`checksumme.txt`" würde dann den berechneten Hashwert enthalten. Für andere Hash-Algorithmen können Sie entsprechend das Kommando `in sha512sum`, `sha1sum` oder z. B. `md5sum` abändern. Im Windows command prompt wäre das entsprechende Konsolenkommando:

`certutil -hashfile <pfad zur datei> SHA256 > checksumme.txt`.

Sollte Ihr System die obigen Konsolenbefehle noch nicht unterstützen oder Sie jetzt noch nicht mit der Konsole arbeiten wollen, empfehle ich Ihnen als ersten Schritt direkt die im letzten Abschnitt als erste Option genannte ungesicherte Installation von GnuPG. Laden Sie die Installationsdatei der Software von deren HTTPS-geschützter Homepage herunter und installieren Sie sie ohne Eigensicherung. Bei der HTTPS-geschützten GnuPG-Homepage ist das Risiko einer Infektion durch Malware nach meiner Einschätzung geringer als auf einer Webseite zum Download eines

[16] Recherchieren Sie zu "IP Targeting advertising" oder besuchen Sie beispielsweise die URL: https://www.dbswebsite.com/blog/ip-targeting-101-smart-display-advertising/

beliebigen freien Windows-Programmes zur GUI-unterstützten Hashberechnung.

Wenn Sie dennoch jetzt die Behelfsprozedur für Eigensicherung durchführen wollen, aber noch keine Hashwerte berechnen können, wird Ihnen die Suchmaschine Ihrer Wahl sicherlich ein Link zu einer dafür geeigneten GUI-basierten Software liefern. Suchen Sie in diesem Fall z. B. nach "calculate hash values on Windows". Die Software sollte mindestens die längst veralteten, bereits gebrochenen, aber gerne genutzten Hashes MD5 und SHA1, sowie die noch als sicher betrachteten Hashes SHA256 und SHA512 berechnen können. Weiterhin sollte der Herausgeber der Software Ihnen vertrauenswürdig erscheinen. Weil Sie zu diesem Zeitpunkt vermutlich noch nicht über den Tor-Browser[17] verfügen, empfiehlt es sich, bei der Suche nach Seiten mit Referenzen zu solcher freier Windows-Software Adblocker, Tracker Blocker, Virenschutz etc. in voller Breite zu aktivieren, um das digitale Ungeziefer des Windows-Biotopes ein wenig auf Abstand zu halten.

Download der GnuPG-Dateien

Wir fahren nun mit der Installation von GnuPG fort. Wenn Sie es noch nicht getan haben, dann booten Sie bitte jetzt Ihren Rechner und öffnen Sie einen Internet-Browser. Sie finden Werner Kochs GnuPG in der Version für Windows bei der in der Fußnote angegebenen URL[18]. Ich finde die auf der Webseite zu bearbeitende Aufforderung zu einer Zahlung oder Spende ein wenig aufdringlich. Tragen Sie 0 Euro ein und klicken Sie auf "Donate & Download". Wenn Sie später regelmäßig GnuPG nutzen sollten, können Sie gerne zu dieser Stelle zurückkehren und Werner Koch und seine Mitstreiter finanziell unterstützen.

Diejenigen Nutzer, die sich für die ungesicherte Option entschieden haben und an dieser Stelle weder digitale Signaturen prüfen noch Hashwerte berechnen können, sollten einmal tief Luft holen, ein kurzes Stoßgebet gen Himmel schicken und die Installation "einfach so", ungesichert durchführen.

Diejenigen anderen Nutzer, die zwar nicht digitale Signaturen prüfen, aber schon Hashwerte berechnen können, sollten das Behelfsverfahren fortsetzen und auf Hashwerte zur Absicherung zurückgreifen. Laden Sie zur GnuPG-Installationsdatei auch den SHA256-Hashwert der Datei herunter. Zunächst überprüfen Sie mit eigener Hashberechnung, ob die heruntergeladene Datei tatsächlich den angegebenen und heruntergeladenen Hashwert liefert. Wenn dies der Fall ist, war der Download zumindest konsistent. Aber ein mächtiger Angreifer, der den Datenverkehr überwacht und auch HTTPS-Verkehr öffnen kann, z. B. ein staatlicher Nachrichtendienst, könnte immer noch böswillig gezielt für Sie sowohl das Download-Paket als auch den Hashwert durch ein anderes passendes Paar ersetzen, das implantierte Malware enthält.

Behelfsprüfung der GnuPG-Dateien

Sie können wie folgt vorgehen, um die zuvor beschriebenen, speziell gegen Sie gerichtete Angriffe zu erschweren. Finden Sie zunächst Ihre gegenwärtige IP-Adresse heraus, die Sie für die letzten Downloads verwendet haben.
Hinweis: Diese IP-Adresse identifiziert Sie persönlich. Selbst wenn Sie keine statische, sondern eine sich häufig ändernde, dynamisch zugewiesene IP-Adresse haben, weiß zumindest Ihr Internet Service Provider (ISP), zu welchem Zeitpunkt Ihr Haushalt welche seiner IP-Adressen verwendet hat. Der ISP könnte mit solchen Informationen handeln oder sie an Behörden weitergeben. Falls die Behörden Sie bereits im Visier haben, könnten sie sogar beim ISP die Information über die Ihrem Haushalt zugewiesenen IP-Adressen abonniert haben.

Greifen Sie zur Tastatur Ihres Systems und schreiten Sie zur Tat: Ihre bevorzugte Suchmaschine antwortet sicher mit einer reichhaltigen Linkliste, wenn Sie nach "find my IP address free" suchen.

[17] Der Tor-Browser ist ein spezieller Browser, der jeden Internetzugriff über das Anonymisierungsnetzwerk Tor routet, um ein Tracking des Internetnutzers so weit wie möglich zu erschweren. Wir behandeln Tor in diesem Buch erst in Abschnitt 3.3.3 auf Seite 56.

[18] https://gpg4win.org/download.html

Ich habe es gerade getan und habe die in den Fußnoten[19] [20] angegebenen URLs als zwei der ersten drei Ergebnisse gefunden. Gehen Sie zu einer solchen Webseite und notieren Sie sich die IP-Adresse, die die Seite für Ihren Internetzugang anzeigt. Dann sollten Sie etwas Zeit verstreichen lassen und die Download-Seite von GnuPG mit einer anderen IP-Adresse erneut aufsuchen. Wie das geht, lesen Sie im nächsten Absatz.

Im Normalfall erhalten Sie von Ihrem ISP eine dynamisch zugewiesene IP-Adresse. Um für den zweiten Zugriff auf die Webseite eine andere dynamisch zugewiesene IP-Adresse zu erhalten, können Sie Ihren Heimrouter ausschalten, einen Kaffee zu kochen und eine Kaffeepause machen. Irgendwann später schalten Sie Ihren Router wieder ein. Ihr ISP hat Ihrem Anschluss nun normalerweise eine neue Adresse zugewiesen. Sie können dies überprüfen, indem Sie auf einer der zuvor gefundenen Webseiten Ihre neue, nun aktuelle IP-Adresse abfragen.

Wenn Sie eine Leitung mit feststehender IP-Adresse haben, oder wenn Sie generell etwas höhere Sicherheit wünschen, können Sie über ein VPN erneut auf die Webseite von gpg4win zugreifen. Vorsichtige und fleißige Nutzer, die diesem Kurs etwas vorgreifen wollen, können z. B. hier[21] eine Liste freier VPNs erhalten und ein dort gelistetes VPN nutzen. VPNs werden in diesem Buch aber erst an späterer Stelle, in Kapitel 3 über rezeptive Anonymität ab Seite 45, detailliert behandelt.

Mit der neuen IP-Adresse für Ihren Zugang laden Sie nun rasch und ohne Umwege den SHA256-Hashwert für die GnuPG-Installationsdatei erneut herunter. Überprüfen Sie, ob er mit dem zuvor heruntergeladenen Wert identisch ist. Wenn dies der Fall ist und Sie sich zuvor vergewissert hatten, dass dieser Wert für die heruntergeladene Installationsdatei tatsächlich korrekt ist, können Sie die Installationsdatei für GnuPG nun mit tolerierbarem Risiko auf Ihrem Windows-System ausführen.

Wenn das im letzten Absatz beschriebene Kontrollverfahren scheitert, sollten Sie alarmiert sein. In den allermeisten Fällen ist das Scheitern auf einen zufälligen Fehler in der Datenübertragung zurückzuführen. Es könnte aber auch ein Hinweis auf einen laufenden Angriff auf Ihren Computer sein. Sie sollten solche Hinweise niemals ignorieren. Brechen Sie in diesem Fall die Installation ab und überprüfen Sie Ihr System auf eine Infektion mit Malware. Führen Sie die gleiche Prozedur einige Tage später mit erhöhter Wachsamkeit erneut aus.

Bestehen Sie auf einem positiven Test! Wenn das Verfahren wiederholt fehlschlägt, installieren Sie die Software nicht. Dies gilt für jede Behelfssicherung einer Softwareinstallation. Die wiederholt fehlgeschlagene Sicherung und in Folge abgebrochene Installation einer Software ist besser als ein kompromittiertes System. Häufig könnte sich ein Angreifer vielleicht nicht direkt in Ihr System einhacken, aber er könnte durchaus in der Lage sein, Ihren Datenverkehr zu stören und so Ihre korrekte Softwareinstallation zu verhindern. Menschen reagieren in der Regel auf eine Funktionsstörung, indem sie Sicherheitsvorkehrungen fallen lassen, nur um das geplante Ziel doch noch zu erreichen. Durch Stören der Funktion könnte ein Angreifer Sie also dazu bringen, die Sicherheit so weit herabzusetzen, dass Ihr System für seinen Angriff doch noch verwundbar wird.

[19] https://whatismyipaddress.com/ip-lookup
[20] https://www.whatismyip.com
[21] https://www.vpngate.net/en/

Zusammenfassend: Behelfsverfahren für signaturlose Verifikation allein auf Basis der Angabe eines Hashwertes.

1. Installationsdatei und Soll-Hashwert herunterladen.
2. Übereinstimmung des Hashwertes der Installationsdatei mit dem Soll-Hashwert überprüfen.
3. Router ausschalten, Zeit verstreichen lassen (je mehr, desto besser) und Sitzung mit neuer IP-Adresse starten, ggf. sogar über ein freies VPN.
4. Soll-Hashwert erneut herunterladen und auf Übereinstimmung mit dem vorigen Soll- und Ist-Wert überprüfen.
5. Bei Übereinstimmung kann ein mäßig vorsichtiger Nutzer die Installationsdatei nun ausführen.

Schlussbemerkungen und Ausblick zur Sicherung von Softwareinstallationen

Wenn Sie später öffentliche Schlüssel eines Programmherausgebers herunterladen, sollten Sie ebenfalls das Herunterladen von Signatur (und Installationsdatei) vom Herunterladen des öffentlichen Schlüssels zeitlich und nach IP-Adresse entkoppeln. Beschaffen Sie sich am besten auch dann beides in getrennten Sitzungen. In diesem Fall brauchen Sie das aber für einen öffentlichen Schlüssel nur einmal zu tun und können ihn danach bei sich aufbewahren bzw. in GnuPG (oder später Academic Signature) einpflegen. Zum Umgang mit GnuPG werden Sie im Übungsteil einige Aufgaben und Anleitungen finden.

Der eine oder andere Leser wird einfach die GnuPG-Installationsdatei heruntergeladen und ohne Prüfung installiert haben, da das seinem bisherigen normalen Verhalten entspricht und das bei bisher fehlender Möglichkeit zur Berechnung von Hashwerten sogar hier angeraten wurde. Vielleicht haben Sie als ein solcher Leser bei diesem ungesicherten Verfahren tatsächlich noch nie eine ungewollte Manipulation Ihres Systems bemerkt und diese erfolgte, wenn überhaupt, allenfalls verdeckt. Aber genau hier und jetzt haben Sie ein mächtiges Werkzeug für manuelle kryptografische Eigensicherung und damit für abgesicherte Installationen erworben. Nun ist ein geeigneter Zeitpunkt, mit altem riskanten Verhalten zu brechen und zukünftig nur Installationsdateien auszuführen, die Sie durch digitale Signaturen authentifiziert haben.

Das Ungewollte passiert zwar selten, aber über viele Monate und ggf. Jahre passiert gelegentlich doch einmal eine Infektion mit Malware. Viele private Windows-Rechner haben im Laufe einer jahrelangen riskanten Benutzung offene oder stille Infektionen mit Malware erlitten. In einer Untersuchung[22] von Kaspersky im Jahr 2013 zeigte sich, dass etwa 5 % aller untersuchten Windows-Systeme trotz Virenschutz aktive Malware enthielten und 13 % der Windows-Computer ohne Virenschutz mit aktiver Malware infiziert waren. In der Untersuchung ging es nur um klassische Windows-Viren, die vom Kasperski-Virenschutz detektiert wurden. Ich vermute, dass eine erhebliche Dunkelziffer an Malware nicht erfasst wurde.

Bitte beachten Sie: Die überwiegende Mehrheit der Malware-Infektionen resultiert nicht aus dem auf Seite 27 beschriebenen gezielten Angriffstyp. Die Mehrheit gehört zu zwei anderen Kategorien. Die erste davon besteht darin, das Opfer auf eine bösartig programmierte Webseite zu locken, die Fehler in einem Script-Interpreter des Browsers ausnutzt. Die zweite besteht darin, naive Benutzer dazu zu bringen, speziell präparierte E-Mails im Outlook zu öffnen. Angreifer können diese E-Mails mit bösartig präparierten Office-Dokumenten als Mailanhang gespickt haben. Solche Office-Dokumente enthalten in der Regel Makros, die MS-Office-Fehler ausnutzen, um die Kontrolle über den Computer des Opfers zu übernehmen.

Die zuvor in diesem Abschnitt erwähnten raffinierteren, maßgeschneiderten Angriffe auf die Computer von ausgewählten Zielpersonen kommen nicht häufig vor. Aber diejenigen, die sich wie wir mit Verschlüsselung und Anonymisierung befassen, um die Kontrolle über unsere Daten zu behalten, können leicht in den Fokus staatlicher Nachrichtendienste geraten. Nachlässig oder gar

[22] https://eugene.kaspersky.com/2013/03/25/one-in-twenty-is-the-sad-truth/

nicht geprüfte Installationsdateien wären eine Einladung für Angriffe solcher Dienste.

Teil 2, erste Schritte mit GnuPG

Mit der Installation von GnuPG haben wir uns im vorigen Abschnitt bereits befasst. Diese Installation war nur für Benutzer von Windows notwendig. Nun, da wir über eine Installation von GnuPG verfügen, müssen wir Schlüssel in das Repository von GnuPG importieren oder auch generieren, um sie nutzen zu können. Dies ist sowohl für Windows- als auch für Linux-Nutzer erforderlich.

Absolvieren Sie jetzt bitte eine vorgezogene Übung. Erstellen Sie mit GnuPG ein OpenPGP-Schlüsselpaar aus öffentlichem und privatem Schlüssel, das mit Ihrer E-Mail-Adresse verknüpft ist. Wenn Sie Ihre bevorzugte Suchmaschine verwenden, werden Sie leicht hilfreiche Anweisungen für diese Aufgabe finden. Heutzutage sollte ein RSA-Schlüsselpaar mindestens eine Länge von 3072 Bit haben, um als sicher zu gelten. Es gibt einige grafische Benutzeroberflächen für GnuPG, die die Erstellung des RSA-Schlüsselpaares erleichtern. Viele versierte Computerbenutzer bedienen GnuPG jedoch gerne direkt über die textorientierte Konsole, anstatt eine grafische Benutzeroberfläche (GUI) als Vermittler zu verwenden.

Falls Sie sich noch nicht bereit fühlen, das Schlüsselpaar selbst und nach Ihren Internet-Rechercheergebnissen zu erzeugen, können Sie nun vorzeitig die Einführungsübungen 2.5.1 auf Seite 35, im Detail VSx1_I1 und VSx1_I2, durcharbeiten und dann gemäß VSx1_I5 durch den Befehl `gpg2 --gen-key` den Schlüssel über die Konsole erzeugen.

Später benötigen Sie den öffentlichen Schlüssel des Autors dieses Leitfadens, meinen öffentlichen Schlüssel. Im nächsten Schritt werden wir diesen öffentlichen Schlüssel in den Schlüsselspeicher Ihrer GnuPG-Implementierung importieren. So sieht der Schlüssel aus:

```
    ---BEGIN PGP PUBLIC KEY BLOCK----
Version: GnuPG v1.4.11 (GNU/Linux)

mQINBE8FgGEBEACllAF1ZZbngJcJbLxCsD/CvBIsdO32rdUoew/SI9S+4sp+Dl6uzWFER63c6ImwL
3Nr2lgw3Q9F7tZfPQcolvIFhjnvQ6Zp2yHe1ZyJkAZ8f4Pjg1wuMt2Ra979SDRSYPpEmWhvRkG+3o
rBFeqhUfNAGrZw/SdcgMNjbzsQqW2qresxY4wtEyDC7Y6ADBUmi8SCiW8EpWnoi2IZP+KPwvVAOL3
Jv4BXOLImWjGwYtGoV1OByQv2HTmrKUbX7M9GZWENPsUBLLLhRTFOhDt4wuXZrSmNhzu2tCqATEe0
sB7nyZz3Ou0Os3uS6o9hEZ2EqtRVkDTAwazQ9szPkAFIQRxtFpsmumOlSP8bjHNtDamht2bCe97lu
4PFEnghFibXnQioT5d4SVfR8e5av989mpmeKwLzm3wulrXpY6x+PjJOhayfYClPh3+C7qK8jrxOWu
PsLGHa6HHyNoys+/3cnHBRMDxla8NxzbLWOyUmDHGE9R99wXzJEqNE3VeGhT2t41XQkKWHR4TjSoQ
/2FOGu7ghgPIF6NDef42gh/n8J1uowu8AYEfvfqzJbCSy9wISWP4eGJYpSgFnvwJRvZk3foUvTXMW
FnhVfJ/X6BmYbEjq/ZKH7ob5 0q49wV89wMjCaSQ6pFmJr3l6/bDhkuWMJ/KVpYMKrvGqMPOFQ/ZA
+/w9owARAQABtDJQcm9mLkRyLk1pY2hhZWwgQW5kZXJzIChrZXlfMjAxMikgPGFuQGZoLXdlZGVsL
mRlPokCPgQTAQIAKAUCTwWAYQIbLwUJEswDAAYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQcM
XXdBxtaFrp2w/+IIJS6ihldDFyUmCQaN63dV/8vhOPRL4n1HL+Dltq8QiW5kptL5pcrpYWwQvX+f4
H9xnO3y/SOt2b/+VE6P1xYYxyiYc+sOlqUFnWqpOtdt4uwFHRUAh8uPQVURY7aVcZ3K8H5opR4A3H
GIVxXPArZbphPQNDL/OEsfGloAujCfBav/QeW4h2+LljaDslt/F3p+VDBPrwHtRBd4cCroZ326Y/X
NYfByopXWVqXVKtTL8J5vzt/UTqhbKYAlho4Z4Hu9MYmcX/VOQYS3s/ODCeQvhRJv/FbOvA+1mA78
bNDA/fDG5ByQ8GhRkVMxv1MOyH9TwEZnPWD7T+gpPyhfU+fqH3UQ4yr3ShmkSaLffVd9nlfcGHvNB
vI+U853v19s+YGaTKi9or6Y2YZ8lAQ1rJ6Rh4MPHnD7qkQfCR+ciE54J7fNmWgEXOhED1MKywt4dN
UPUzkVOKol7hHxCL/Bt+taDOB6il98dAJOdB7xlyi4+F2z+BmQE5wLwwkPeMYJiSy96uCZ65wpPjC
DUJIfnUEqc/dYzVrgJz6TBTZAiCI97OMDUj89/J1HWXOoPeTWDDvStOWNFSELuHXIb7vFnV+nPGsO
PhZANiKX7H1Pajb1PT5ypMhmsMJOculJuTclc583BPKT+oGsL7Khd5h5OL1Q44yl+8vtLhUSVw3uy
KX1E= =pw2U
-----END PGP PUBLIC KEY BLOCK-----
```

Öffentliche Schlüssel im OpenPGP-Format, wie der oben gezeigte, werden normalerweise in Radix64-Codierung dargestellt. Diese Codierung verwendet ausschließlich druckbare Zeichen, sodass wir sie problemlos als Ausdruck auf Papier darstellen oder über textorientierte digitale Übertragungskanäle transportieren können. Es wäre natürlich unsinnig, den oben gedruckten Schlüssel über das Keyboard abzutippen. Sie können den oben angegebenen öffentlichen Schlüssel

in elektronischer Form, fertig zum Copy&Paste, von der HTTPS-gesicherten Homepage von Academic Signature[23] kopieren. Er befindet sich unten auf der Webseite, auf die in der letzten Fußnote verwiesen wird.

Wieder können Sie Ihre bevorzugte Suchmaschine benutzen, um eine Webseite zu finden, die gut erklärt, wie man einen öffentlichen Schlüssel in den internen Schlüsselspeicher von GnuPG importiert. Suchen Sie zum Beispiel nach "import GnuPG public key". Die Verwendung eines Werkzeugs mit einer grafischen Benutzeroberfläche kann diesen Import erleichtern. Falls Sie sich noch nicht bereit fühlen, dies nach dem Ergebnis eigener Internet-Recherchen zu tun, können Sie wie folgt vorgehen:

1. Kopieren Sie den Schlüssel, wie im obigen Kasten abgedruckt, von der präsentierenden Webseite und fügen Sie ihn z. B. in eine neu erzeugte Textdatei mit dem Namen "anders_key.asc" ein.
2. Speichern Sie diese Textdatei in einem Verzeichnis Ihrer Wahl.
3. Öffnen Sie eine Konsole[24] und navigieren Sie in der Konsole zu diesem Verzeichnis.
4. Tippen Sie das folgende Kommando in der Konsole ein:
   `gpg2 --import anders_key.asc` .
   Wenn Sie einen anderen Namen für die Textdatei verwenden, die den in Radix64 codierten Schlüssel enthält, verwenden Sie natürlich diesen Namen anstelle des Platzhalters im obigen Beispiel-Konsolenbefehl.
5. Achten Sie auf die Rückmeldung von gpg und schließen Sie die Konsole wieder. Wenn die Antwort keine Fehlermeldung gewesen war, sollte sich der öffentliche Schlüssel jetzt im Schlüsselspeicher von GnuPG befinden.

Eine alternative, wesentlich unsichere Quelle für öffentliche Schlüssel könnte ein öffentlicher Schlüsselserver sein. Eine Suchmaschine Ihrer Wahl bietet Ihnen bei der Suche nach "public key server" o. Ä. sicherlich die in der Fußnote[25] angegebene Webseite an. Auf dieser Webseite werden viele öffentliche Schlüssel eingestellt. Den oben angegebenen öffentlichen Schlüssel können Sie dort bei entsprechender Suche auch leicht finden (und zusätzlich einige Karteileichen zu meinem Namen, auf die ich keinen Zugriff mehr habe). Alternativ können Sie meinen Schlüssel aber auch mit Angabe des sogenannten Fingerprints[26] direkt über das Terminal von einem Schlüsselserver beziehen. Das Kommando hierzu wäre
`gpg2 --keyserver keys.gnupg.net --recv 0x1C6D685A` .
Die Zuordnung zur Person bei Bezug über Schlüsselserver ist allerdings unsicher. Jeder könnte auf solchen öffentlichen Schlüsselservern beliebige Schlüssel einstellen. Es wäre beispielsweise kein Problem für Sie, dort den öffentlichen Schlüssel eines von Ihnen generierten Schlüsselpaars zum Namen Karl der Große, Josef Stalin oder Mutter Theresa einzustellen.

Experten mögen jetzt auf das sogenannte Web of Trust (WOT) verweisen, das die Zuordnung von öffentlichen Schlüsseln zu Personen absichern kann. Ich will hier nicht auf das bei fehlender Notwendigkeit von Anonymität hilfreiche WOT eingehen, da dessen Nutzung Ihr persönliches Beziehungsnetzwerk jedem fähigen Angreifer auf dem silbernen Tablett serviert und Anonymität nachhaltig verbrennt.

2.4.3 Manueller Umgang mit digitalen Signaturen: Academic Signature

Hinweis: Der Autor dieses Buches entwickelt, pflegt und verteilt das kostenfreie, quelloffene Programm Academic Signature. Es bestehen aber keinerlei wirtschaftliche Interessen des Autors

[23] siehe: https://www.academic-signature.org/academic_signature_key.html
[24] siehe den Eintrag im Glossar auf Seite 188 und 2.5.1 auf Seite 35.
[25] https://pgp.mit.edu/
[26] Der Fingerprint eines OpenPGP-Schlüssels ist eine Hexadezimalzahl, die sich aus dem gekürzten Hashwert des Schlüssels ergibt und diesen kurz und eindeutig kennzeichnet.

bezüglich dieser Software.

Academic Signature verwendet das fortschrittlichste der etablierten Kryptosysteme, die Elliptische-Kurven-Kryptografie[27] (ECC). Der Nutzer steuert das Programm manuell über eine grafische Benutzeroberfläche und kann damit sogenannte Negligible Adversary Advantage[28] [29] Chiffrate erstellen. Ein Angreifer kann solche Chiffrate nicht von statistischem Byte-Rauschen unterscheiden. Diese Chiffrate sind voll kompatibel mit Anonymität und erfüllen daher das Königskriterium der sicheren Verschlüsselung. Meines Wissens gibt es keine anderen freien, quelloffenen Werkzeuge, die es erlauben, solche Chiffrate mit hybrider Verschlüsselung (siehe Abschnitt 5.3) zu erstellen und die vollständig mit anonymer Verwendung kompatibel sind (und meines Wissens auch keine derartigen kommerziellen Werkzeuge).

Das im letzten Abschnitt behandelte GnuPG ist weit verbreitet, aber kann in der Benutzung etwas sperrig sein. Es verwendet nicht die modernsten Algorithmen sowie die nicht mehr ganz zukunftsfesten Kryptosysteme RSA und ElGamal. Leider ist außerdem ein GnuPG-Chiffrat für den Angreifer immer als solches zu erkennen und der Betrieb mit verdeckter Empfänger-ID ist wenig praktikabel. Weil GnuPG kaum mit anonymer Kommunikation verträglich ist, gibt es den dringenden Bedarf für ein anonymitätskompatibles Verschlüsselungswerkzeug wie Academic Signature.

Sie finden die Installationsdatei für Windows oder den leicht selbst compilierbaren Quellcode für Academic Signature als "Standard Tarball"[30] über die Suchmaschinen Google, Startpage, Bing, Yandex, Baidu,... oder direkt hier[31].

Laden Sie die Installationsdatei Ihrer Wahl herunter und auch meine GnuPG-Signatur der entsprechenden Datei. (Laden Sie für spätere Verwendung zusätzlich auch gleich die zweite Signatur der Installationsdatei herunter, meine ECDSA-Signatur.) Über meinen öffentlichen Schlüssel für GnuPG sollten Sie zu diesem Zeitpunkt bereits verfügen. Suchen Sie mit einer Suchmaschine Ihrer Wahl z. B. über die Suche "Verifikation of digital signatures using GnuPG"[32] eine Anleitung zur Verifikation von OpenPGP-Signaturen. Vielleicht werden Sie dabei auf diese Webseite[33] verwiesen.

Folgen Sie nun den Anweisungen auf der Webseite, die Sie unter den Suchergebnissen am geeignetsten gefunden haben. Überprüfen Sie das Tripel aus der Installationsdatei (oder dem Quellcode-Tarball), der entsprechenden OpenPGP-Signaturdatei und dem öffentlichen Schlüssel des Herausgebers (in diesem Fall meinem). Auf einem langsamen System kann die Überprüfung einige Sekunden dauern. Bei positivem Ausgang der Verifikation können Sie die Installation ausführen.

Die Installation der Binärdatei sollte auf Windows ca. 2-3 Sekunden dauern. Wenn Sie mit Linux arbeiten, sollte die Kompilation des Quellcodes in 3-5 Minuten beendet sein, die anschließende Installation erfolgt in einem Wimpernschlag. Beim ersten Aufruf von Academic Signature durchlaufen Sie eine Sequenz, in der u. a. Ihr Zugangspasswort gesetzt, der Zufallszahlengenerator sicher gestartet und Ihr erstes Schlüsselpaar generiert wird. Über ein grafisches Interface in Academic Signature können Sie nun zusätzlich zu den Academic-Signature-Funktionen auch auf GnuPG-Funktionen zugreifen, wenn GnuPG ebenfalls installiert ist. Mein öffentlicher ECC-Schlüssel wird automatisch mit dem Programm Academic Signature mitgeliefert und befindet sich nach der Installation bereits im gesicherten Ablagebereich von Academic Signature.

---

[27] siehe: https://de.wikipedia.org/wiki/Elliptic_Curve_Cryptography

[28] siehe: https://en.wikipedia.org/wiki/Advantage_(cryptography)

[29] siehe auch: https://en.wikipedia.org/wiki/Ciphertext_indistinguishability

[30] Der Tarball (Teerkugel) ist die verbreitete Form, ein Programm als Quellcode zu verteilen. Er ist ein komprimiertes Archiv aus allen benötigten Dateien, der in standardisierter Weise vom Nutzer zum ausführbaren Programm verarbeitet und dann installiert werden kann.

[31] https://www.academic-signature.org/

[32] Bei einer deutschsprachigen Suche werden Sie leider wenige geeignete Quellen finden.

[33] https://www.gnupg.org/gph/en/manual/x135.html

Prüfen Sie nun als erste informelle Mini-Übung auch die Installationsdatei mit meiner digitalen ECDSA-Signatur gegen meinen mitgelieferten öffentlichen Schlüssel. Manche Nutzer empfinden die Menüführung dazu als selbsterklärend, andere können auf die Videotutorials auf der Downloadseite zurückgreifen. Die Prüfung sollte erfolgreich verlaufen, schließlich hatten Sie ja bereits meine GnuPG-Signatur verifiziert.

Sie verfügen nun mit GnuPG und Academic Signature über zwei unabhängige Werkzeuge zum Erstellen und Überprüfen von digitalen Signaturen und können somit unabhängig von Automatismen Ihres Betriebssystems auf Ihrem Rechner kryptografische Eigensicherung betreiben.

2.5 Übungen zur vollen Systemherrschaft auf Ihrem Computer

In den letzten beiden Abschnitten sind Sie bereits auf einige informelle Übungsabschnitte gestoßen. Nun kommen wir zu den ersten formalen Übungen. In solchen Übungen sollten Sie einen eingeschalteten Computer neben dem Buch liegen haben, wenn Sie den Text lesen. So können Sie während des Lesens mit diesem Computer im Internet recherchieren und üben. Im Folgenden finden Sie die Sammlung formaler Übungen, mit denen Sie Ihre Fertigkeiten bezüglich der Basis der Pyramide des Dissidentenschutzes einüben können. Sie werden später ähnliche Abschnitte mit Übungen für alle anderen Schichten der Pyramide finden.

Alle formalen Übungen in diesem Buch beginnen mit einem gerahmten, kursiv gedruckten Element. Dieses Element gibt den Schwierigkeitsgrad, den geschätzten Zeitaufwand für die Übung, die Voraussetzungen und, falls erforderlich, besondere Anmerkungen zur Übung an. Gelegentlich kann es vorkommen, dass eingestreute Textelemente wieder kursiv gedruckt werden. Diese Passagen sind nicht Teil des engeren Übungsablaufs. Sie geben zusätzliche Informationen, die bei der Fehlersuche hilfreich sein könnten oder die von allgemeinem Interesse sind, um Ihre Ergebnisse einzuordnen. Sie werden feststellen, dass ich den Übungstiteln eine Kennung beifüge. In den Bezeichnungen dieses Übungsblocks steht VS für "Volle Systemkontrolle" und x für "exercise".

2.5.1 VSx1, Vorübungen

Schwierigkeitsgrad: mittel.
Übungszeit: 30-90 min.
Voraussetzungen: ein laufender Computer an Ihrer Seite.

I1)

Legen Sie auf Ihrem Rechner ein Arbeitsverzeichnis für diese Übungen an.

I2)

Konsolenübung (langweilig und mühselig, aber leider nötig).

Mit der Konsole können Sie unter Umgehung der grafischen Benutzeroberfläche, des GUI (= Graphical User Interface), auf Ihrem System Betriebssystemfunktionen und andere Programme aufrufen. Man muss sich diese Konsolen-Kommandos leider merken, sie haben aber jahrzehntelang Stabilität und werden nicht wie ein GUI häufig geändert.

Computer-Nerds lieben die Konsole. Das Verschlüsselungsprogramm GnuPG wird gerne über die Konsole betrieben. Bei havarierten Computersystemen mit zerschossenem GUI hat man manchmal nur noch über die Konsole Zugriff und kann manchmal nur noch über die Konsole die Havarie beheben oder wenigstens Daten retten.

- Recherchieren Sie im Internet nach einer Seite mit den für Ihr System gültigen Konsolen-Kommandos. (z. B. Suchmaschine: "console commands PowerShell", "console Command Prompt commands" oder "linux console commands".)

- Starten Sie eine Konsole (unter Windows "Command Prompt" oder "PowerShell"). Die Konsole befindet sich nach dem Start im sogenannten "Home-Verzeichnis" als aktuellem Verzeichnis.

Um Ihnen einen ersten Eindruck zu geben, liste ich hier einige nützliche Kommandos für die Linux-Konsole auf:

ls Erstellt ein Listing des aktuellen Verzeichnisinhaltes in der Konsole.

cd Dieses Kommando, gefolgt von einem Leerzeichen und der relativen Angabe des Zielverzeichnisses, wechselt ins angegebene Verzeichnis.

cd .. Das Kommando, gefolgt von einem Blank und zwei Punkten, wechselt in das übergeordnete Verzeichnis.

mkdir Dieses Kommando (make directory), nach einem Leerzeichen gefolgt von der Angabe eines bisher noch nicht existierenden Verzeichnisnamens, erzeugt dieses Verzeichnis.

rm Dieses Kommando (remove), nach einem Leerzeichen gefolgt von der Angabe eines Verzeichnisses oder eines Dateinamens, löscht dieses Verzeichnis oder diese Datei.

Sie werden viele weitere nützliche Konsolenkommandos auf den Seiten finden, die Sie bei Ihrer Internetrecherche gefunden haben. Wenn Sie unsicher über die Verwendung eines Kommandos sind, so rufen Sie es mit dem Parameter `--h` auf, also etwa `rm --h`. Dann wird in der Konsole ein Hilfetext ausgegeben.

- Navigieren Sie nun in der Konsole in das in I1) erzeugte Verzeichnis und listen Sie dort den (noch nicht vorhandenen) Inhalt.
- Navigieren Sie anschließend in der Konsole zurück in Ihr Home-Verzeichnis und listen Sie den Inhalt des Home-Verzeichnisses.
- Dann können Sie die Konsole wieder schließen.

I3)

Wählen Sie einen einfachen Texteditor für die folgenden Übungen aus, mit dem Sie Textdateien im ASCII-Format verfassen und editieren können. In der Regel finden Sie auf Ihrem System einen geeigneten Editor vor.

I4)

Verfassen Sie einen kleinen Text mit einer Handvoll Sätze und speichern Sie den im Arbeitsverzeichnis unter einem Dateinamen Ihrer Wahl ab.

I5)

Wenn Sie es bisher noch nicht getan haben, erzeugen Sie sich ein privat/öffentliches Schlüsselpaar für GnuPG (RSA) und eines für Academic Signature (ECC).

- Für GnuPG können Sie das in der Konsole mit dem Kommando `gpg2 --gen-key` tun.
- Für Academic Signature verwenden Sie dazu die GUI des Programms.

Im Folgenden gehe ich davon aus, dass Sie Academic Signature und GnuPG installiert und die einführenden Übungen I1-I5 bearbeitet haben.

2.5.2 VSx2, Digitale Signatur mit Academic Signature

Schwierigkeitsgrad: einfach.
Übungszeit: etwa 10 min.
Voraussetzungen: Übungen VSx1 I4, VSx1 I5 und die Installation von Academic Signature.
Hinweis: Diese Übung ist eine Voraussetzung für weitere Übungen.

1. Erstellen Sie eine digitale Signatur für die Textdatei aus der vorherigen Einführungsübung VSx1 I4 mit Academic Signature unter Verwendung Ihres neu erstellten privaten ECC-

Schlüssels. Wenn Sie Hilfe benötigen, werfen Sie einen Blick auf die Anleitungen und Tutorials auf der Homepage von Academic Signature.
2. Prüfen Sie anschließend das Tripel, bestehend aus Textdatei, Signaturdatei und Ihrem öffentlichen Schlüssel auf Konsistenz. Umgangssprachlich: Prüfen Sie die Signatur.
3. Invertieren Sie ein einzelnes Bit der Textdatei. Tun Sie dies beispielsweise, indem Sie einen Großbuchstaben in einen Kleinbuchstaben ändern oder umgekehrt. Speichern Sie die Datei und überprüfen Sie die digitale Signatur erneut. Die Überprüfung muss jetzt fehlschlagen.
4. Invertieren Sie das Bit erneut und stellen Sie damit den ursprünglichen Zustand der Datei wieder her, speichern Sie sie und überprüfen Sie die Gültigkeit der Signatur erneut.

2.5.3 VSx3, Digitale Signatur mit GnuPG

Schwierigkeitsgrad: mittel.
Übungszeit: etwa 20 min.
Voraussetzungen: alle Einführungsübungen und die Installation von GnuPG.
Hinweis: Diese Übung ist eine Voraussetzung für viele weitere Übungen.

1. Erstellen Sie eine digitale Signatur für die Textdatei aus der vorherigen Einführungsübung VSx1 I4 mit GnuPG und Ihrem neu erstellten privaten RSA-Schlüssel. Sie müssen dies nicht unbedingt über die Konsole tun. Sicherlich finden Sie im Internet Anleitungen und Tutorials, wie Sie dies durchführen können. Verwenden Sie Ihre bevorzugte Suchmaschine und wählen Sie eine gute Quelle für Hilfe aus. Überprüfen Sie anschließend die Signatur. Ich werde hier keine expliziten und detaillierten Anleitungen geben und Sie stattdessen ermutigen, sich auf Ihre Fähigkeiten und Ihre Fertigkeit bei Internetrecherchen zu verlassen. Ich wäre jedoch nicht überrascht, wenn Ihre Internetsuche Sie auf die in der Fußnote angegebene URL verweisen würde[34], und wenn Sie dort folgende Kommandos für Signatur und Verifikation finden würden:
   `--detach-sign` und `--verify` .
2. Invertieren Sie ein einzelnes Bit Ihrer Textdatei, z. B. indem Sie einen Großbuchstaben in einen Kleinbuchstaben ändern oder umgekehrt, speichern Sie die Datei und überprüfen Sie die Signatur erneut. Die Überprüfung muss jetzt fehlschlagen.
3. Invertieren Sie das Bit erneut und stellen Sie damit den ursprünglichen Zustand der Datei wieder her, speichern Sie sie und überprüfen Sie die Gültigkeit der Signatur erneut.

2.5.4 VSx4, Webseitenzertifikate

Schwierigkeitsgrad: einfach.
Übungszeit: etwa 15 min.
Voraussetzungen: keine.

Finden Sie eine Internetseite, auf die Sie geschützt mit HTTPS zugreifen können (z. B. Google.de oder das Internetportal Ihrer Bank). Dabei werden im System bereits vorhandene kryptografische Sicherungen im Hintergrund genutzt. In diesem Fall stellen die im Hintergrund arbeitenden Sicherungen sicher, dass sich keine gefälschte Webseite beispielsweise als Ihre Homebanking-Startseite ausgeben kann, etwa um Ihnen Zugangscodes oder TAN-Nummern zu entlocken.

Besuchen Sie die Seite und inspizieren Sie das verwendete Zertifikat. Ein Zertifikat ist in diesem Zusammenhang ein öffentlicher Schlüssel, reichlich garniert mit mehr oder weniger wichtigen Zusatzinformationen. Dieses Zertifikat und der Domainname der Webseite sind miteinander verknüpft.

[34] siehe: https://www.gnupg.org/documentation/manuals/gnupg/Operational-GPG-Commands.html

Die Domain ist die Webseite, deren Name typischerweise auf .org, .com, .gov, .de oder Ähnliches endet, und deren Unterseiten. Beim Aufbau einer verschlüsselten HTTPS-Kommunikation beweist die Domain dem Besucher, dass sie Zugriff auf den privaten Schlüssel hat, der zu dem im Zertifikat der Webseite angegebenen öffentlichen Schlüssel gehört. Das ist deren Authentisierung gegenüber dem Besucher.

Bei Firefox würden Sie zur manuellen Inspektion des Zertifikates auf das kleine Vorhängeschloss in der URL-Zeile klicken, dann (hier und heute bei englischsprachigem Menü) auf "Show connection details" und "more Information". Leider ändern die Entwickler diesen Zugangsweg häufig, sodass sicherlich schon ein anderer Zugangsweg aktuell ist, wenn Sie diese Zeilen lesen. Sie öffnen jedenfalls die Seiteninformation, wenn Sie sich erfolgreich durch die Menüs geklickt haben.

Wenn Sie einen anderen Browser benutzen, suchen Sie bitte komplett selbst nach diesen Informationen. Lassen Sie nicht locker, bis Sie zumindest den öffentlichen Schlüssel in menschenlesbarer Form, das verwendete Kryptosystem (RSA, ElGamal, ECC, ...) und die Bitlänge des Schlüssels gefunden haben.

2.5.5 VSx5, Authentifizierung über die Automatismen des Betriebssystems

Schwierigkeitsgrad: einfach.
Übungszeit: etwa 15 min.
Voraussetzungen: keine.
Hinweis: Diese Übung ist eine Voraussetzung für viele weitere Übungen.

Bitte installieren Sie eine zusätzliche, nützliche und freie Software für Ihr System. Wählen Sie eine Software, die zur Absicherung der Installation auf die automatischen kryptografischen Schutzvorrichtungen des Betriebssystems zurückgreifen kann. Ich schlage vor, in dieser Übung den kostenlosen Chat-Client Pidgin zu installieren.

Variante für Windows-Nutzer

1. Verwenden Sie eine Suchmaschine Ihrer Wahl, um die Download-Webseite der Software zu finden. Besuchen Sie die Download-Seite der Software mit einem Browser Ihrer Wahl.
2. Laden Sie die Installationsdatei herunter und speichern Sie sie auf Ihrer Festplatte. Führen Sie sie noch nicht aus. Möglicherweise müssen Sie die Systemeinstellungen ändern, um die automatische Ausführung heruntergeladener Installationsprogramme zu verhindern und eine feinere Steuerung der Installationsprozesse auf Ihrem System zu ermöglichen.
3. Starten Sie nun die Ausführung des Installationsprogramms. Notieren Sie alle Systemmeldungen und Anfragen während dieses Installationsprozesses einschließlich Ihrer Eingaben. Überlegen Sie sich, welche davon mit kryptografischen Schutzvorrichtungen zu tun haben könnten. Sammeln Sie alle Informationen, die Sie über die vom Betriebssystem verwendeten kryptografischen Verfahren in Erfahrung bringen können (Schlüssellänge, verwendete Hash-Algorithmen, verwendetes Kryptosystem). Legen Sie sich Erklärungen zurecht: Was war der eigentliche Sinn jeder sicherheitsrelevanten Anfrage oder Ausgabe?

Variante für Linux-Nutzer

1. Wählen Sie entweder den Befehlszeilen-Paketmanager oder einen grafischen Paketmanager für die Installation. Verwenden Sie den Manager, um Pidgin oder andere Software zu installieren, die Sie für diese Übung ausgewählt haben. Starten Sie dann den Download und die Installation.
2. Notieren Sie alle Systemmeldungen und Anfragen während dieser Installation einschließlich Ihrer Eingaben. Überlegen Sie sich, welche davon mit kryptografischen Schutzvorrichtungen zu tun haben könnten. Sammeln Sie alle Informationen, die Sie über die verwendeten

kryptografischen Verfahren in Erfahrung bringen können (Schlüssellänge, verwendete Hash-Algorithmen, verwendetes Kryptosystem). Legen Sie sich Erklärungen zurecht: Was war der eigentliche Sinn jeder sicherheitsrelevanten Anfrage oder Ausgabe? Ist eine der Ausgaben Ihrer Meinung nach unnötig? Fehlt Ihrer Meinung nach eine bestimmte Ausgabe oder Abfrage?

2.5.6 VSx6, Manuelle Authentifizierung einer Installationsdatei mit GnuPG

Schwierigkeitsgrad: einfach.
Übungszeit: bei erstmaliger Ausführung etwa 30 min, bei routinierter Ausführung 1-2 min.
Voraussetzungen: Installation von GnuPG und Pidgin, alle Vorübungen VSx1 und VSx3.
Hinweis: Diese Übung ist eine Voraussetzung für viele weitere Übungen.

Zu einem späteren Zeitpunkt werden Sie das OTR-Plugin[35] für die Verschlüsselung von Chats in Pidgin benötigen. Die Kanadischen Cypherpunks haben das OTR-Plugin entwickelt und pflegen es. Auf einem Windows-System installieren Sie das Plugin mit einer Installationsdatei, die Sie von der entsprechenden Webseite herunterladen können. Auf einem Linux-System ist das OTR-Plugin normalerweise in den Repositories enthalten und kann mit der in Linux verfügbaren automatischen Authentifizierung durch einen einfachen Mausklick heruntergeladen und sicher installiert werden. Für diese Übung sollten Linux-Nutzer jedoch, genau wie Windows-Nutzer, OTR manuell herunterladen und den Download manuell authentifizieren.

Die Verwendung der Primärquelle eines Softwarepakets, das vom Herausgeber mit dem privaten Schlüssel des Herausgebers signiert ist, ist immer eine engere und damit sicherere Interaktion als die Verwendung des Sicherheitsmechanismus des Betriebssystems. Letzteres würde einen Vermittler auf der Seite des Betriebssystems hinzufügen, der ein unnötiges zusätzliches Ziel für Angriffe oder Bestechung durch böswillige Organisationen sein könnte.

1. Finden Sie mit einer Suchmaschine Ihrer Wahl die offizielle Downloadseite des OTR-Plugins für Pidgin von den Cypherpunks.
2. Besuchen Sie die Webseite und laden Sie das Installationsprogramm und seine digitale Signatur auf Ihren Computer in ein Verzeichnis Ihrer Wahl herunter. Die OTR-Entwickler vermeiden Absicherung über den Windows-Schutz-Automatismus und ermöglichen so eine manuelle Überprüfung der gegebenen GnuPG-Signatur, ohne sich auf den Konzern Microsoft verlassen zu müssen.
3. Suchen und finden Sie in einer separaten Sitzung den öffentlichen Teil des Schlüssels, mit dem die Entwickler die Installationsdatei des OTR-Plugins signiert haben. Importieren Sie diesen öffentlichen Schlüssel in den Schlüssel-Vorrat von GnuPG.
4. Überprüfen Sie die Signatur des OTR-Plugins anhand des Installationsprogramms und des öffentlichen Schlüssels des Entwicklers. Nach erfolgreicher Überprüfung führen Sie das Installationsprogramm aus.
   Hinweis: Bei einem Fehlschlag wiederholen Sie den Vorgang. Bei wiederholtem Fehlschlag informieren Sie die Entwickler über das Authentifizierungsproblem. Seriöse Entwickler sind immer daran interessiert, herauszufinden, ob Benutzer auf solche Schwierigkeiten stoßen, die auf aktuelle böswillige Störungen hinweisen könnten.

[35] Das Akronym OTR steht für Off The Record.

2.5.7 VSx7, Manuelle Authentifizierung einer Installationsdatei mit Academic Signature

Schwierigkeitsgrad: einfach.
Übungszeit: bei erster Ausführung etwa 10 min, mit Routine etwa eine Minute.
Voraussetzungen: Installation von Academic Signature.
Hinweis: Diese Übung ist Voraussetzung für einige andere Übungen.

Ich nehme an, Sie haben das Programm Academic Signature bereits heruntergeladen und installiert. Am Ende der Installationsanleitung in Abschnitt 2.4.3 wurden Sie bereits gebeten, die digitale Signatur des Installationsprogramms zu prüfen. Wenn Sie das bereits getan haben, können Sie diese Übung überspringen. Ich halte es jedoch für sinnvoll, diese Übung als Teil dieses formalen Aufgabenblocks zur Verfestigung zu wiederholen.

Falls Sie meine digitale ECDSA-Signatur des Academic Signature Installationsprogramms noch nicht heruntergeladen haben, holen Sie das bitte jetzt nach.

- Bitte verifizieren Sie nun (nacheilend) mit Academic Signature, dass meine ECDSA-Signatur der Installationsdatei, die Installationsdatei und mein öffentlicher Schlüssel ein gültiges Tripel bilden. Man nennt das dann umgangssprachlich etwas unpassend eine "korrekte Signatur".

Sicherlich gelingt die Verifikation der Installationsdatei. Schließlich hatten Sie ja bereits die OpenPGP-Signatur verifiziert. Falls die Verifikation aber scheitert, haben Sie möglicherweise das Problem, dass inzwischen ohne Ihr Wissen auf Ihrem Rechner Dateien geändert wurden. Das wäre besorgniserregend.

2.5.8 VSx8, Installieren Sie einen freien Hex-Editor auf möglichst sicherem Weg

Schwierigkeitsgrad: einfach.
Übungszeit: etwa 10 min.
Voraussetzungen: keine.

Sie werden zu einem späteren Zeitpunkt einen sogenannten Hex-Editor benötigen. Ein Hex-Editor ist ein Programm zum Anzeigen und Bearbeiten von Dateien als Zeichenketten aus rohen Hexadezimalzahlen[36]. Solche Editoren benötigen Sie immer, wenn Sie einen ungefilterten Blick auf Dateien haben möchten, ungefiltert durch das Programm, das üblicherweise die Datei für Sie auf dem Bildschirm darstellt. Sie können damit die Informationen ansehen und editieren, die beispielsweise als Metainformationen mit jedem pdf-Dokument oder Word-Dokument mitreisen, welches Sie weitergeben. Insbesondere wenn der Hex-Editor gleichzeitig die Hex- und in einem zweiten Fenster die ASCII-Repräsentation zeigt, können Sie etwa bei E-Mails eine Fülle von Metadaten einsehen und ändern.

Leider werden solche Programme häufig von Menschen entwickelt und verbreitet, die keine besondere Sensibilität in Sicherheitsfragen haben. Daher haben Sie kaum eine Chance, einen Hex-Editor-Installer auf der Webseite eines Entwicklers zu finden, der durch die digitale Signatur des Entwicklers gesichert ist.

1. Suchen Sie bitte mit einer Suchmaschine Ihrer Wahl nach einem solchen Hex-Editor für das Betriebssystem Windows. Linux-Nutzer sollten das als Übung trotzdem für einen Windows-Hex-Editor tun - jeder Linux-Nutzer findet sowieso zahlreiche gute Hex-Editoren in seinen Repositories.
2. Suchen Sie sich aus den Angeboten einen freien Hex-Editor aus, der Ihnen gefällt und im Idealfall mit manuell zu prüfender Signatur ausgeliefert wird, oder wenigstens den Windows-Automatismus nutzt, oder allerwenigstens mit Checksummenangabe ...

[36] Hexadezimalzahlen sind Zahlen, die nicht im Dezimalsystem, sondern im Sechzehnersystem codiert sind.

3. Prüfen Sie die heruntergeladene Installationsdatei so gut wie möglich.
4. Wenn der Sicherheitscheck keinen Hinweis auf Manipulation zeigt, installieren Sie als Windows-Nutzer bitte den Hex-Editor.

2.5.9 VSx9, Verschlüsseln Sie mit Academic Signature und GnuPG

Schwierigkeitsgrad: mittel.
Übungszeit: etwa 40 min.
Voraussetzungen: Installation von Academic Signature und GnuPG, Übungen VSx6, VSx7 (2.5.6 und 2.5.7).
Achtung: *Wenn Sie sich im Machtbereich einer repressiven Regierung befinden, könnte es geboten sein, das in dieser Übung nahegelegte Verschicken des Chiffrates per nonymer (= nicht anonymer) Mail zunächst zu unterlassen und es lediglich vor der Dechiffrierung auf Ihrem System in ein anderes Verzeichnis zu verschieben. Andernfalls würden Sie sich als Benutzer wirksamer Verschlüsselung kenntlich machen und könnten unerwünschte Aufmerksamkeit staatlicher Stellen auf sich ziehen. In Deutschland besteht dieses Problem wohl nicht.*

Tun Sie so, als würden Sie als Bürger in Ihren Freiheitsrechten respektiert und als wäre für Sie das Verhalten nach der Pyramide der freien Bürger (Abb. 1.1) ausreichend und angemessen.

1. Bitte wählen Sie ein Dokument mit unverfänglichem Inhalt.
2. Wenn Sie noch keine privat-öffentlichen Schlüsselpaare für GnuPG (RSA) und Academic Signature (ECC) erstellt haben, tun Sie es bitte jetzt.
3. Verschlüsseln Sie das ausgewählte Dokument asymmetrisch (in der Tat mit hybrider Verschlüsselung) und erstellen Sie ein GnuPG- und ein Academic-Signature-Chiffrat[37] des unverfänglichen Dokuments. Verwenden Sie dazu Ihre jeweiligen öffentlichen Schlüssel.
4. Wenn Sie dies in Ihrem Staat riskieren können, senden Sie das GnuPG-Chiffrat als E-Mail-Anhang von einem Ihrer E-Mail-Konten an ein anderes von Ihnen kontrolliertes E-Mail-Konto.
5. Wenn Sie dies in Ihrem Staat riskieren können, senden Sie das Academic-Signature-Chiffrat als E-Mail-Anhang in die entgegengesetzte Richtung zwischen Ihren E-Mail-Konten.
6. Laden Sie die E-Mail-Anhänge jeweils in einen Ordner Ihrer Wahl herunter und entschlüsseln Sie die verschiedenen Chiffrate jeweils mit Ihren privaten Schlüsseln.

Für diese Aufgaben gibt es im Internet zahlreiche gute Tutorials und Anleitungen. Finden Sie einige davon mit Ihrer bevorzugten Suchmaschine. Solche Quellen können diese Übung erheblich erleichtern, wenn sie gut und für Sie passend sind. Verbringen Sie einige Zeit damit, auf eigene Faust zu recherchieren, um die Anleitungen zu finden, die Ihnen am geeignetsten erscheinen. Handeln Sie entsprechend Ihren Rechercheergebnissen.

Bei Academic Signature sollten die Tutorials auf dessen Webseite ausreichen und die Funktionen sollten in der grafischen Benutzeroberfläche selbsterklärend sein, bei GnuPG gilt das eher nicht. Ich würde mich aber nicht wirklich wundern, wenn Sie als Hilfeseite zu GnuPG die in der Fußnote[38] angegebene Seite fänden.

Dort würden Sie vielleicht das folgende Konsolen-Kommando für hybride Verschlüsselung finden:

```
gpg --encrypt 04F028F8 test.txt
```

und für die hybride Entschlüsselung:

```
gpg --decrypt --output test.txt test.txt.gpg
```

.

[37] Menüpunkt: "ECC-Krypto -> crypto_Funktionen -> ECC_Verschluesseln"
[38] siehe: https://www.gnupg.org/documentation/manuals/gnupg/

In der ersten Kommandozeile steht 04F028F8 für den Fingerprint des von Ihnen verwendeten Schlüssels. Er muss natürlich in der konkreten Übung durch den Fingerprint Ihres Schlüssels ersetzt werden. "test.txt" steht für den Dateinamen Ihres unverfänglichen Klartextes, und "test.txt.gpg" steht für den Namen des Chiffrates. In beiden Befehlszeilen müssten Sie die Zeichenfolge "test.txt" durch den Namen der von Ihnen für diese Übung gewählten Datei ersetzen. Bevor Sie die oben erwähnten Konsolenbefehle abschicken, sollten Sie in der Konsole zu dem Ordner navigiert haben, in dem sich die entsprechenden Dateien befinden. Im Entschlüsselungskommando von GnuPG brauchen Sie den Fingerabdruck des Schlüssels nicht anzugeben, da ein unverschlüsselter Verweis darauf im Chiffrat enthalten ist. Bitte beachten Sie, dass dies die Anonymität brechen kann, falls dieses Chiffrat in anonymisierter Kommunikation verwendet würde.

Hinweis: Viele Mailanbieter blocken Mails mit Anhängen, die sie nicht einordnen können, manche nehmen daran sogar ungefragt Änderungen vor. Gute Chiffrate können die Anbieter aber selbstverständlich nicht einordnen! Falls Mails mit solchen Chiffraten blockiert oder die Chiffrate beschädigt werden, sollten Sie Ihr Chiffrat in eine ZIP-Datei[39] *einpacken. Mails mit solchen Anhängen werden dann üblicherweise klaglos und unbeschädigt transportiert.*

Herzlichen Glückwunsch! Sie haben nun den ersten Übungsblock abgeschlossen.

2.6 Angriffe auf Ihre Systemkontrolle

Gleich im Anschluss an den Übungsteil folgen jetzt einige Gedanken über Verwundbarkeiten, Gefährdungen und Möglichkeiten des Angriffs. Dies werde ich in späteren Abschnitten auch für alle anderen Schichten der Pyramide des Dissidentenschutzes (siehe Abb. 1.2) tun. Hier dreht sich alles um Sicherheitsgefährdungen und regelrechte Angriffe auf Ihre Kontrolle über Ihren Computer.

Meine hier niedergeschriebenen Gedanken decken die Gefahren natürlich in in keiner Weise vollständig ab. Bitte bedenken Sie, dass es ein Universum von weit zahlreicheren Gefährdungen und Angriffsmöglichkeiten auf Ihre Systemkontrolle gibt, die ich hier nicht diskutieren kann.

2.6.1 Systemd (bei Linux)

Das hier beschriebene Problem ist kein offener Angriff auf Linux, sondern kann eher als ein verdecktes Sicherheitsrisiko angesehen werden.

Zur Grundphilosophie von Linux und als Grundunterscheidungsmerkmal zu Windows zählt die von UNIX geerbte modulare Struktur des Systems. In neuerer Zeit ist u. a. zur Verbesserung der Effizienz des Bootprozesses eine größere Systemkomponente Systemd[40] entwickelt worden. Systemd ist nominell zwar auch modular aufgebaut, aber durch viele intermodulare Abhängigkeiten faktisch doch ein monolithischer Block.

Bei vielen gängigen Linux-Distributionen wurde Systemd eingeführt. Traditionelle Linux-Entwickler werden mehr und mehr gezwungen, das möglicherweise ungeliebte Systemd zu unterstützen. Auf diese Weise entzieht Systemd einen immer größeren Teil der Linux-Welt dem Zugriff der Gemeinschaft freier Entwickler und die Pflege verbleibt in den Händen weniger. Es bilden sich damit Strukturen, die anfälliger für Kaperung durch eine zentrale Stelle sein könnten. Viele Linux-Enthusiasten sehen eine solche Machtkonzentration in der Linux-Welt sehr kritisch.

Außerdem steht der Vorwurf im Raum, Systemd würde die Unix-Grundphilosophie der Modularität durch "Feature Creep", d. h., das kontinuierliche Einsickern von immer mehr Funktionen in die Komponente Systemd, und durch "Software Bloat", das ist kontinuierliches Aufblähen des Umfangs und des Rechenzeitbedarfes, missachten. Über "Feature Creep" und "Software Bloat"

[39] Eine ZIP-Datei ist ein verlustlos komprimiertes Dateiarchiv, das mit vielerlei Programmen, beispielsweise mit Winzip, aus Dateien erzeugt und wieder in Dateien entpackt werden kann.

[40] siehe: https://systemd.io/ und https://de.wikipedia.org/wiki/Systemd

würde so Linux in eine schlechte Richtung gedrängt, deren Endpunkt man vom Betriebssystem Windows her zur Genüge kenne.

In jüngster Zeit wurde die Einführung von sogenannten Snaps[41] und Snapd aus ähnlichen Gründen kritisiert. Die herausgebende Firma Canonical hält einen Teil des für Snaps verwendeten Codes in geschlossenem proprietären Quellcode.

2.6.2 Verhalten von Hardware-Herstellern und Handel

Durch die Knebelung des Computer-Kunden werden viele gewinnversprechende Geschäftsmodelle für alle Beteiligten der Wertschöpfungskette rund um den PC möglich. Eines davon ist die Einnahme von hohen Gebühren für Softwarelizenzen bei inzwischen verschwindend geringem Aufwand für Verbesserung, Wartung und Distribution der Software.

Ein anderes Modell mit wachsender wirtschaftlicher Bedeutung ist die Vermarktung von ausgeschleusten Nutzerdaten. Diese Daten können leicht aus den Systemen der abhängig gemachten Privatkunden abgezogen werden.

Es gibt eine weitere Unsitte. Große Handelsketten oder die Computerhersteller selbst bekommen Gebühren für die Installation von Crapware auf den Systemen. Diesen Punkt haben wir bereits in Abschnitt 2.1 auf Seite 22 berührt. Crapware ist vom Kunden unerwünschte Software, die sich nur schwer oder gar nicht von neu gekauften Computern der Kunden entfernen lässt. Unternehmen nutzen diese Crapware als Marketing-Instrument, um Kunden zum Kauf weiterer Software anzuregen oder zum Nachkauf einer zunächst zeitlich begrenzten Nutzungszeit einer Software zu animieren.

Solche Geschäftsmodelle florieren dann, wenn Hersteller und Händler dem Kunden den Weg zu selbstbestimmter Arbeit an Computern mit durchgehend freier Software nach Kräften erschweren.

Es war zu erwarten und ist auch selbstverständlich schon eingetreten, dass Systeme bereits, mutmaßlich versehentlich, mit klar als solcher zu identifizierender Malware - Viren und/oder Trojaner - ausgeliefert wurden. Das habe ich selbst bei einem neuen Windows 10 System bereits einmal feststellen müssen. An einigen Stellen kann man solche Allianzen zulasten Dritter beobachten. Der Dritte ist regelmäßig der entmündigte Computernutzer.

Leider ist es in den letzten Jahren aus meiner Sicht aufgrund zahlreicher neu eingeführter Hindernisse erheblich schwerer geworden, einen frisch erworbenen Windows-Rechner zu einem Windows/Linux Dual Boot System aufzurüsten. Ich finde es besonders dreist, dass solche Hürden mit der Notwendigkeit besserer IT-Sicherheit begründet werden.

Rechner ohne vorinstallierte kommerzielle Betriebssysteme bekommt man heutzutage fast nur noch im spezialisierten Online-Versand. Aufgrund der Vorteile für Softwareanbieter und Hardwarehersteller treiben sie die Verzahnung zwischen Hardware- und Softwarekomponenten auf Kosten der Modularität und der Wahlfreiheit der Kunden voran. Das abschreckende Beispiel für Privatanwender, die versuchen, sich der digitalen Bevormundung zu entziehen, ist der beklagenswerte Stand der Dinge, in dem wir uns bezüglich der IT-Sicherheit bei Mobilsystemen heute schon weitgehend befinden. Diese Systeme mögen für wenig ambitionierte Nutzer einfach in der Bedienung wirken. Sie sind aber dazu konstruiert, der Nutzerkontrolle entzogen zu sein und den Nutzer in Abhängigkeit und Unwissenheit zu halten. Zugleich scheint genau dieses Beispiel die Vision der Akteure in der computerbezogenen Wertschöpfungskette zu sein.

2.6.3 Microsoft übt volle Kontrolle über Ihr Windows-Betriebssystem aus

Das Windows-Betriebssystem der Firma Microsoft war in der Vergangenheit ein relativ offenes Betriebssystem, das dem privaten Nutzer wenig Grenzen gesetzt hat. Im Zuge der Einführung neuer Versionen dieses Betriebssystem wurde das System immer mehr gegenüber dem Nutzer

[41] siehe: https://en.wikipedia.org/wiki/Snap_(package_manager)

verschlossen. Das Booten eines alternativen Betriebssystems auf Windows-Computern wird, wie bereits im letzten Abschnitt angesprochen, durch prohibitive Schikanen zunehmend erschwert.

Weiterhin ist im aktuellen Windows 10 außer durch hartes physisches Stilllegen der Interfaces oder Blockaden in einer externen Firewall die ständige Kommunikation des privaten Windows-Rechners mit Microsoft-Servern nicht mehr zu unterbinden. Mühsam manuell unterbundene Geschwätzigkeit des Systems erscheint nach einem Systemupdate manchmal perfekt wiederhergestellt.

Einem um seine Selbstbestimmung und den Schutz seiner Privatsphäre bedachten Nutzer eines privaten Computers scheint man heute die Migration zu Linux empfehlen zu müssen.

3. Rezeptive Anonymität

Wir werden uns jetzt auf die zweite Ebene der Pyramide des Dissidentenschutzes konzentrieren. Dieses Kapitel hat die gleiche Grundstruktur wie das Kapitel für die Basisschicht.

Zunächst werde ich den Begriff der rezeptiven Anonymität einführen und einen Blick auf die Theorie werfen. Dann werden wir uns freie und quelloffene Werkzeuge ansehen, die helfen, die Ziele dieser Schicht zu erreichen. Was folgt, sind wieder praktische Übungen, für die neben dem aufgeschlagenen Buch Ihr laufender Computer benötigt wird. Zum Abschluss des Kapitels schauen wir auf Schwachstellen und Möglichkeiten zum Angriff, damit wir Attacken besser verstehen und abwehren können.

3.1 Was ist rezeptive Anonymität?

Die Einführung der Begriffe rezeptive und expressive Anonymität in diesem Buch ist geleitet von den Unterschieden in den Arten von Datenströmen, deren Endpunkte schutzbedürftig sind. Der hier verwendete Ausdruck rezeptive Anonymität bedeutet, auf im Netz bereitgestellte Ressourcen lesend zugreifen zu können, ohne dabei einem ressourcenseitigen Überwacher des Datenverkehrs Informationen über die eigene Identität zu geben. Der ressourcenseitige Überwacher soll den beobachteten Datenverkehr keiner realen und keiner zu schützenden fiktiven Identität eines Internetnutzers zuordnen können. Zugleich soll einem Überwacher aus dem Umfeld des Nutzers keinerlei Information über die abgerufene Ressource zugänglich sein. Salopp gesagt geht es um Techniken, die beispielsweise in einer extrem prüden Gesellschaft einem Teenager erlauben würden, sich im Netz über Verhütung, Schwangerschaftsabbrüche oder ein homosexuelles Coming-out zu informieren, ohne eine Steinigung zu riskieren. Natürlich gibt es auch diejenigen, die sich in einer Diktatur für indizierte Kunst oder Literatur interessieren und nach einem Besuch entsprechender Webseiten nicht in einem Foltergefängnis landen möchten.

Rein technisch gesehen gibt es selbst bei rein lesendem Surfen im Web neben dem inhaltlich reichen eingehenden Datenverkehr immer auch ausgehenden technisch motivierten Datenverkehr. Dies kann z. B. der Handshaking-Verkehr zum Aufbau einer HTTPS-Verbindung sein. Solcher technische Datenverkehr enthält jedoch keine individuell geprägten Gedanken, Meinungen, Formate oder Sprachschnipsel. Er enthält keine Informationen, die es Überwachern in einer autoritären

Verwaltung erlauben würde, Rückschlüsse auf die Person vor dem Bildschirm und deren politische Einstellung zu ziehen.

Die hier eingeführte begriffliche Unterscheidung in rezeptive und expressive Anonymität beruht auf der von dem vor dem Bildschirm sitzenden Nutzer ausgehenden oder eben nicht ausgehenden Information. Bestimmend ist, ob individuell geprägte Formulierungen, Gedanken oder Meinungen über den Rechner in das Internet und in die Welt herausgehen oder ob nur gelesen wird. Im ersten Fall bezeichne ich den benötigten Typ Anonymität in diesem Buch als expressiv. Die expressive Anonymität wird im nächsten Kapitel behandelt und entspricht der nächsthöheren Stufe in der Pyramide des Dissidentenschutzes. Im letzteren Fall, den wir in diesem Abschnitt behandeln, nenne ich die Anonymität rezeptiv.

Bei lediglich rezeptiver Anonymität ist anonyme Interaktion ausgeschlossen. Auch ein individueller Login bei einem Mail-Konto einer fiktiven Identität oder einem Internetportal wäre eine solche Interaktion, müsste vermieden werden und wäre nicht mit rezeptiver Anonymität verträglich. Es gibt allerdings einen Lohn für diese Einschränkung. Die in diesem Kapitel behandelte rezeptive Anonymität ist wesentlich schwerer anzugreifen als expressive Anonymität. Der Angreifer kann den Datenverkehr eines bestimmten Nutzers nicht von denen aller anderen auch nur rezeptiv agierenden anonymisierten Nutzer unterscheiden. Der Angreifer kann den Datenverkehr also nicht einer zunächst fiktiven Identität zuordnen. Folglich kann er auch dieser fiktiven Identität keine zeitlichen oder inhaltlichen Verhaltensmuster zuordnen und analysieren. Der Angreifer kann dann nicht den Personenkreis einengen, den er verdächtigt, hinter der fiktiven Identität zu stehen. Bei expressiver Anonymität, die für Interaktion notwendig wäre, wäre genau dies eine sehr starke Waffe des Angreifers.

Politisch ist die Möglichkeit rezeptiver Anonymität in einer Demokratie unverzichtbar, damit einflussreiche Personen einer Gesellschaft (Geistliche, Politiker, Militärs, Wissenschaftler…), die kleineren oder größeren Schwächen, oft im sexuellen Bereich, im Internet nachgehen, nicht durch Nachrichtendienste des eigenen oder eines anderen Landes erpressbar werden.

Beide Formen der Anonymität (expressiv und rezeptiv) lassen sich im Gegensatz zu Vertraulichkeit ermöglichender Inhaltsverschlüsselung niemals aus eigener Kraft herstellen. Man benötigt dafür eine anonymitätsfördernde Infrastruktur und eine große Zahl an ebenfalls Anonymität suchenden Personen. Diese müssen gewisse Regeln beachten und ihre Datenverkehre für außenstehende Überwacher untrennbar miteinander mischen. Früher wurde gelegentlich (auch von mir) verbreitet, nur ein verschwindend kleiner Teil der Anonymitätssuchenden wolle politisch, moralisch, kulturell, freiheitlich oder aber auch kriminell motivierte Datenverkehre anonymisieren. Die überwiegende Masse der durch Tor anonymisierten Datenverkehre bestünde aus gewöhnlicher Pornografie. Neue Daten deuten allerdings darauf hin (siehe Abb. 4.5 auf Seite 115), dass der überwiegende Anteil tatsächlich zum Schutz vor politisch motivierter Überwachung durch Polizei und übergriffige Innenbehörden über Tor geleitet wird.

So hat sich beispielsweise die Anzahl der täglichen Tor-Nutzer in den USA in der Woche nach dem Mord an dem Afroamerikaner George Floyd[1] durch Polizisten und dem Beginn großer, durch behördliche Überwachungsmaßnahmen[2] begleiteter Demonstrationen der Black Lives Matter Bewegung[3] vom langzeitigen Mittelwert von 300 000 auf etwa 900 000 verdreifacht. Erst mit dem Beginn der heißen Wahlkampfphase Trump/Biden sank die Anzahl der täglichen Tor-Nutzer in den USA wieder auf den ursprünglichen Sockelwert.

Unsere Regierungen rechtfertigen Angriffe auf die Anonymität gerne mit dem Kampf für Moral, gegen Terrorismus und gegen besonders geächtete Formen der Pornografie, obwohl aus

[1] https://en.wikipedia.org/wiki/George_Floyd

[2] siehe z. B. https://www.cnet.com/news/house-dems-ask-fbi-others-to-stop-spying-on-black-lives-matter-protesters/

[3] https://en.wikipedia.org/wiki/Black_Lives_Matter

meiner Sicht die Überwachung politischer Dissidenten[4] [5], Kontrolle der Bevölkerung und besserer fiskalischer Durchgriff auf Finanzströme die wirklichen Ziele sind. Bei repressiven Regierungen ist die tatsächliche Motivation für Angriffe auf die rezeptive Anonymität, zur Machtvermehrung Erpressungsmaterial gegen Bürger sammeln zu können oder Bürger, über die bisher keine Einordnung der politischen Präferenz besteht, politisch als Freund oder als Gegner zu klassifizieren.

3.2 Geopolitische Aspekte der Anonymisierung digitaler Kommunikation

Viele naive Computernutzer überschätzen die Macht staatlicher Angreifer als Allmacht und akzeptieren sie fatalistisch. Andere fühlen sich als besonders versierte Nutzer und erliegen einer nahe liegenden Illusion. Weil sie selbst keine Chance zu massendatenbasierten Angriffen haben und deshalb diese auch gar nicht erst durchdenken, sehen sie sich diesen auch nicht ausgesetzt. Ein staatlicher Gegner kann und wird diese aber sehr wohl ausführen. Es ist ebenso wichtig wie schwierig, sich in die ausschließlich sehr mächtigen Angreifern zur Verfügung stehenden Möglichkeiten durch Zugriff auf Massendaten hineinzudenken. Nur so kann man als Überwachungsgegner einschätzen, welche Verteidigungsmaßnahmen wirksam sein könnten.

Wer Zugriff auf die Netzinfrastruktur hat und den gesamten Datenverkehr in einer Region/einer Nation sehen, analysieren und sogar in der Transportgeschwindigkeit modulieren kann, kann Anonymität sehr viel wirksamer angreifen als eine kleine Einheit mit nur sehr begrenzter Sicht auf den Verkehr und vielleicht nur Zugriff auf einige wenige Router im Internet.

Hier kommt das Konzept des Datenkraken ins Spiel. Ich verwende in diesem Buch den Begriff Datenkrake für Unternehmen wie Google, Alibaba, Facebook, Amazon, Microsoft und für technisch fortgeschrittene staatliche Nachrichtendienste wie die NSA oder das GCHQ[6], die zu einem solchen praktisch unbegrenzten Echtzeitzugriff in der Lage sind. Mit Zugriff auf die Massendaten und Infrastruktur einer Region kann über Modulationen[7] und Korrelationsanalysen theoretisch jeder Kommunikationsstrang, der komplett innerhalb dieser Region verläuft, zunächst markiert und letztlich nonymisiert werden. Inwieweit unsere westlichen Dienste praktisch dazu schon in der Lage sind, ist nicht öffentlich bekannt.

Anonymität lässt sich nur dann einigermaßen zuverlässig erreichen, wenn der Datenverkehr die Grenzen von Domänen überschreitet, deren Massendaten in Gänze einem einzelnen Angreifer zur Verfügung stehen. Punktuelle Zusammenarbeit der Dienste verschiedener Domänen ist unvermeidbar und wohl auch im allgemeinen Interesse. Sie ist aber unkritisch für den Schutz der Privatsphäre. So werden die Dienste Russlands und des US-Einflussbereichs sicherlich bei der Verfolgung des einen oder anderen Islamisten oder Drogenhändlers punktuell zusammenarbeiten. Das bricht aber nicht die Möglichkeit zur Anonymisierung domänenüberschreitenden Datenverkehrs, solange diese beiden politischen Domänen keine Massendaten teilen. Ich bin recht zuversichtlich, dass sie das auch nicht tun, weil die Auslieferung der eigenen Massendaten an eine andere politische Domäne einer Kapitulation gegenüber deren Diensten gleichkäme.

Nach meiner Einschätzung würden die Nachrichtendienste der Domänen "Five Eyes" (US, UK, Aus, NZ, Can) und enge Verbündete (inkl. Deutschland), Russland und enge Verbündete, China und asiatische Staaten mit kommunistischer Historie, "Singapur+Indonesien+Malaysia" und "US-kritisches Lateinamerika" misstrauisch genug gegeneinander sein, um NICHT Massendaten aus Ihren jeweiligen Regionen zu teilen. Vielleicht bilden auch westlich geprägte asiatische Staaten wie

[4] Siehe auch die Verfolgung des Journalisten Markus_Beckedahl des Internetportals netzpolitik.org durch deutsche Sicherheitsbehörden https://de.wikipedia.org/wiki/Netzpolitik.org

[5] Siehe die Verfolgung von Julian Assange und von Edward Snowden.

[6] https://de.wikipedia.org/wiki/Government_Communications_Headquarters

[7] Es handelt sich nicht um Modulation im fernmeldetechnischen Sinn, sondern um Meta-Modulationen etwa in der Latenz der Datenpakete eines Kommunikationsstromes. Man kann sich das Internet wie eine hochentwickelte, verzweigte Rohrpost für Datenpakete vorstellen.

Japan, Südkorea und Taiwan eine eigene, von den Five Eyes getrennte Domäne. Möglicherweise kommen noch auf ihre Souveränität bedachte Einzelstaaten wie Frankreich, Iran, Israel, die Schweiz oder Indien hinzu, die man dann bezüglich Internet-Massendaten auch als isolierte Domänen betrachten könnte.

Ich möchte betonen, dass ich diese Domänengrenzen nach meinem Bauchgefühl und meiner politischen Intuition gezogen habe und daraus für meinen Datenverkehr Schlussfolgerungen ziehe. Der Leser dieses Ratgebers mag eigene, von meinen abweichende Erfahrungen oder Einschätzungen haben, die vielleicht auch fundierter als meine Mutmaßungen sind. Selbstverständlich wird dieser Leser bei seiner Anonymisierung kritischer digitaler Kommunikation die eigene "Domänenlandkarte" zugrunde legen.

Wenn ich mit Leben oder Gesundheit auf die Anonymität eines Kommunikationsstranges angewiesen wäre, würde ich sicherstellen, dass dieser Kommunikationsstrang neben der Five Eyes Einflusszone mindestens eine andere politische Domäne durchquert, auch wenn der Strang letztlich nur zu meinem Nachbarn auf der anderen Straßenseite geht[8]. Man kann dann hoffen, dass der Weg der Datenpakete in dieser Nachbardomäne weit genug ist, damit dort genügend statistisch verrauschte Zeitverzögerungen stattgefunden haben, die dem Zugriff und der Analyse von Überwachern aus der ersten Domäne entzogen waren. Diese würden Korrelationsanalysen der eigenen staatlichen Angreifer massiv behindern.

Trotz des internationalen Charakters des Internets scheinen die Internetkulturen, die Besorgnis hinsichtlich der Achtung von Privatsphäre, das Vertrauen in die eigene Regierung und die Nutzung des Internets in den verschiedenen Regionen der Welt recht unterschiedlich zu sein. Das bei Weitem am häufigsten eingesetzte Anonymisierungsnetzwerk Tor[9] schränkt die Macht von Regierungen ein, aber Tor braucht Freiwillige, die Server als Tor-Knotenpunkte betreiben. Daher ist das Angebot an Tor-Knoten in den politischen Domänen der Welt recht ungleichmäßig verteilt. Die lokale Verfügbarkeit von technischem Fachwissen, privaten Geldmitteln und der Grad des Misstrauens gegenüber staatlichen Stellen bestimmen das Angebot an Tor-Knoten. Ich habe noch niemals einen australischen oder neuseeländischen Tor-Knoten nutzen können. Wir finden eine kleine Anzahl von Knoten (fast keine) in Asien oder Afrika. China blockiert die Nutzung von Tor (die Nutzer benötigen besondere Vorkehrungen, um die Blockade zu umgehen). Tor-Knotenpunkte sind in Europa und Nordamerika reichlich vorhanden, in Russland gibt es eine angemessene Anzahl, aber in Südamerika sind sie dünn gesät. Man kann also leider nicht in jeder politischen Domäne auf eine ausreichende Anzahl an Knoten des Tor-Netzwerkes zählen.

3.3 Anonymität durch untrennbares Vermischen von Datenverkehr

In der Regel werden wir einen Internet Service Provider (ISP) dafür bezahlen, dass er einen Zugang zum Internet für uns bereitstellt, der von unserer Wohnung aus genutzt werden kann. Natürlich haben wir damit eine personalisierte Geschäftsbeziehung und der ISP kennt unseren Namen, unsere Kontonummer etc. Der ISP kann jedes von uns eingespeiste Datenpaket oder jedes an uns ausgelieferte Datenpaket unserer Person zuordnen. Ein wohlmeinender, kompetenter ISP mit großer Kundenbasis könnte einen mächtigen Schutz für unsere Anonymität aufbauen, wenn er das wollte. Aber unser ISP kann leider nicht generell als verschwiegene, vertrauenswürdige Instanz betrachtet werden. Selbst wenn er das sein wollte, würde ihn die staatlicherseits verlangte und immer wieder gerichtlich umkämpfte Vorratsdatenspeicherung mit der Verpflichtung der Herausgabe unserer Daten an die Behörden daran hindern.

Wir müssen deshalb dafür Sorge tragen, dass unser ISP weder den Inhalt unserer Datenpakete lesen noch die wahre Destination der ausgehenden Datenpakete oder die wahre Herkunft der

[8] Wir werden u. a. in Abschnitt 3.5.8 auf Seite 77 einen Weg dazu einüben.

[9] Wir behandeln das Konzept des Anonymisierungsnetzwerkes Tor erst später in Abschnitt 3.3.3. Es wird auch kurz im Glossar vorgestellt.

eingehenden Datenpakete kennen kann. Dazu benötigen wir verschwiegene "Strohmänner" als Mittler für unseren Datenverkehr. Um auch Sicherheit gegenüber mächtigen Beobachtern zu erlangen, die Zugriff auf die Netzinfrastruktur und den gesamten Datenverkehr in unserem Land haben, muss unser Datenverkehr und der Datenverkehr anderer Nutzer bei diesen verschwiegenen Mittlern für Außenstehende untrennbar miteinander vermischt werden.

3.3.1 Ein einfacher Internetzugriff

Wir wollen zunächst einen flüchtigen Blick darauf werfen, wie wir üblicherweise eine Internetseite zugreifen. Nehmen wir an, es handele sich um die Seite der Internetversion Ihrer Lieblingszeitung. In meinem Fall wäre das die Internetversion der Wochenzeitung "Die Zeit". Als regelmäßiger Leser kennen Sie die URL Ihrer Lieblings-Onlinezeitung, in meinem Fall, https://www.zeit.de/. Wenn wir auf den Inhalt der Webseite zugreifen wollen, tippen wir die URL in die Kopfzeile unseres Webbrowsers ein. Bei "return" zum Abschicken der Anfrage ruft der Browser zunächst das Domain Name System (DNS) auf.

Das DNS empfängt die menschenlesbare URL, wählt die entsprechende IP-Adresse des Hosting-Servers aus und sendet diese IP-Adresse an unseren Browser zurück. Für den Fall, dass der Server mehrere Webseiten hostet, wird der gewählte Domainname für eine eindeutige Anfrage an den Hosting-Server im laufenden HTTP-Verkehr mit übermittelt. Zum Zeitpunkt, als ich die erste deutsche Ausgabe dieses Leitfadens schrieb (Frühjahr 2017), hatte `zeit.de` für mein System und meinen Netzzugang die IP-Adresse 217.13.68.251.

Als ich die erste englische Ausgabe dieses Buches verfasste, wiederholte ich die Anfrage im April 2020. Nun gab das DNS 217.13.66.41 zurück, ebenso wie 2017 die Adresse eines Servers in Braunschweig, der zu einer Firma namens Gaertner Datensysteme gehört. Zufälligerweise war dieselbe IP-Adresse im Frühjahr 2017 die letzte für das traceroute Protokoll als sichtbar konfigurierte Adresse gewesen. Während des Schreibens der 2. überarbeiteten deutschen Ausgabe im Oktober 2020 hatte der Zeit-Server die 217.13.69.39 beim gleichen ISP in Braunschweig. Anscheinend hat sich also, abgesehen von einer etwas schnelleren Übertragung[10], in der Zwischenzeit nicht viel geändert.

Ihre Internet-Anfrage enthält notwendigerweise Ihre eigene IP-Adresse. Der angesprochene Server muss natürlich wissen, wohin die Antwortdaten geschickt werden sollen. Dazu enthält Ihre Anfrage je nach Browser-Einstellung mehr oder weniger umfangreiche, für die Reaktion nicht benötigte Zusatzdaten, z. B. auf welcher Internetseite Sie sich vor dem Aufruf ggf. aufgehalten haben. Der gewöhnliche Internetverkehr ist sehr geschwätzig.

Bei dem sogenannten Routing der Datenpakete durch das Netz werden sowohl Ihre Anfrage als auch die Antwort des Servers über flexibel bestimmte Zwischenstationen, die Router, durch das Internet geleitet. Bei unverschlüsselten Daten könnte theoretisch jede einzelne der beteiligten Stationen die Daten (und damit auch Ihre Empfänger-IP-Adresse) einsehen und auch die Daten eigennützig modifizieren.

Jedem mit dem Internet verbundenen Computer, wie z. B. einem Router, wird eine IP-Adresse zugewiesen. Andere angeschlossene Systeme sprechen ihn über diese IP-Adresse an. Heutzutage, in Zeiten von Industrie 4.0 und dem Zeitalter der Buzzwords, scheint sogar jeder Blumentopf eine IP-Adresse zu benötigen, um ein guter Blumentopf zu sein. Sie können den Befehl "traceroute" ("tracert" in einer Windows-Konsole) verwenden, um eine Liste der IP-Adressen aller Router zusammenzustellen, die Ihre Datenpakete auf dem Weg zu Ihrem Ziel durchlaufen. Natürlich nur, wenn deren Administratoren sie als sichtbar konfiguriert haben. Es kann gute Gründe dafür geben, dass Routing-Stationen unsichtbar sein sollen. Eine verdeckte Station muss nicht unbedingt eine Abhör- und Überwachungsmaschine sein.

[10] 72 Millisekunden im April 2017 gegen 32 Millisekunden im Oktober 2020.

Werfen wir nun einen Blick auf den Datenpfad durch das Netz von meinem Notebook zum Server der Online-Zeitung. Ich öffne eine Konsole auf meinem Linux-Computer (im Frühjahr 2017) und tippe den Befehl `traceroute zeit.de` . Unten sehen Sie das Traceroute-Protokoll für den Weg von meinem Computer zum Webserver der Zeitung. Ich habe aus Datenschutzgründen einige IP-Adressen an meinem Ende des Datenpfades durch xxxxx.. ersetzt.

```
    traceroute to zeit.de (217.13.68.251), 30 hops max, 60 byte packets
1   gateway (xxxxxxx)  0.238 ms  0.568 ms  0.553 ms
2   xxxxxxxxxxxxxxxxxxxxxxxxxxx  31.696 ms  32.962 ms  35.155 ms
3   62.214.61.201 (62.214.61.201)  36.903 ms  38.626 ms  40.223 ms
4   62.214.37.114 (62.214.37.114)  67.778 ms  67.847 ms  67.828 ms
5   qsc.bcix.de (193.178.185.60)  54.672 ms  74.653 ms  74.668 ms
6   scmber11-et-1.qsc.de (87.234.14.172)  67.612 ms  43.897 ms  44.027 ms
7   scmham11-cg-0-0-1-0.qsc.de (87.234.14.138)  45.833 ms  47.937 ms  49.631 ms
8   scmham12-cg-0-0-1-1.qsc.de (87.234.14.136)  51.623 ms  53.531 ms  55.805 ms
9   scmdus12-cg-0-0-1-0.qsc.de (87.234.14.134)  57.473 ms  59.141 ms  61.065 ms
10   crmdus11-cg-10-0-2-0.qsc.de (87.234.14.167)  63.064 ms  65.021 ms  66.924 ms
11   suhlu-v9.gaertner.de (217.13.66.41)  70.434 ms  71.774 ms  73.946 ms
* * *....
```

Bei Station 11 ist wohl der Zeit-Server erreicht und mir als Anfrager wird nach Station 11 kein Einblick in den weiteren Pfad erlaubt. Hierbei befinden sich die durch traceroute ausgewiesenen Server nach Auskunft einer der vielen populären IP-Geolocation-Webseiten[11] an folgenden Orten:

```
62.214.61.201  (Versatel Server Duesseldorf)
62.214.37.114  (Versatel Server Duesseldorf)
193.178.185.60 (bcix eV   Berlin )
87.234.14.172  (qsc ag,Cologne)
87.234.14.138  (qsc ag,Cologne)
87.234.14.136  (qsc ag,Cologne)
87.234.14.134  (qsc ag,Cologne)
87.234.14.167  (qsc ag,Cologne)
217.13.66.41   (gaertner datensysteme gmbh & co. kg, Braunschweig)
```

Bitte haben Sie Verständnis dafür, dass ich für den qsc-ag-Pingpong in diesem Fall keine Erklärung liefern kann. Die im Servernamen (wie ‘‘`scmber11-et-1.qsc.de`’’) ausgewiesenen Kürzel ber, ham, dus, sehen nach Städtekürzeln aus, aber weitere Angaben dazu wären reine Spekulation von meiner Seite. Der Wert von einer Latenzzeit[12] im Umfang einer hohen zweistelligen Millisekundenzahl - eine knappe Zehntelsekunde - war ein gängiger Wert für Datenverkehre kleiner privater Internetnutzer innerhalb Zentraleuropas. Ende 2020, etwa drei Jahre, nachdem die abgedruckten Traceroute-Protokolle aufgenommen wurden, hat sich die typische Latenzzeit etwa halbiert.

Wir halten fest: in diesem Fall kann mindestens "Versatel", der "bcix eV", die "qsc ag" und die "gaertner gmbh" aufzeichnen, welche Artikel der Zeit der Inhaber meiner IP-Adresse (ich!) sich zu welcher Zeit angesehen hat. Jede der Stationen hätte bei bösem Willen auch die Datenpakete

[11] https://www.neustar.biz/resources/tools/ip-geolocation-lookup-tool

[12] Latenzzeit oder kurz Latenz ist in der Internetkommunikation die Zeitverzögerung zwischen Abschicken einer Anfrage an einen Server und dem Eintreffen der Reaktion.

manipulieren können. Wenn nun eine staatliche Stelle (oder auch ein Abmahnanwalt) bei meinem ISP nachfragt, wer zur fraglichen Zeit die in den Datenpaketen ausgewiesene IP-Adresse zugeordnet hatte, kann diese Stelle nach der Antwort darauf zurückschließen, welche Artikel der Zeit ich persönlich zu welcher Tageszeit gelesen habe und wie lange ich bei den einzelnen Artikeln verweilt hatte.

Hinweis: Zum Zeitpunkt des Schreibens der ersten deutschen Ausgabe dieses Leitfadens war der Zugriff auf "www.zeit.de" noch nicht durch HTTPS geschützt. "www.zeit.de" hat inzwischen einen HTTPS-Schutz für seinen Datenverkehr eingeführt. Damit wurde das unerwünschte Abhören oder Manipulieren der Datenpakete während der Übertragung erheblich erschwert. Nur noch mächtige Instanzen, die in der Lage sind, den HTTPS-Verkehr zu öffnen, z. B. staatliche Geheimdienste, könnten dies noch tun.

Bei zentralen Knoten, wie hier dem Berliner bcix, ist die Versuchung für staatliche Dienste besonders groß, sich Zugänge zu verschaffen. Der zu bcix vergleichbare, deutlich größere Dienst De-Cix hat als zentrale Ausleitungsstelle von Daten an den BND traurige Berühmtheit erlangt. Suchen Sie mit Ihrer Lieblingssuchmaschine nach: "De-Cix BND", lesen Sie dazu einen Artikel in Zeit Online[13], der hoffentlich heute noch für Sie, liebe Leser, verfügbar ist oder besuchen Sie die in den Fußnoten[14] [15] angegebenen englischsprachigen Seiten.

Als kritischer Nutzer des Internets möchte ich nicht, dass unkontrolliert viele Organisationen daraus, was ich wann und wie lange in der Zeitung lese, auf meine politische Einstellung schließen können. Und es gibt weitaus sensiblere Internetaktivitäten als das Lesen einer seriösen Zeitung. Es würde mich nicht wundern, wenn Sie als Leser dieses Ratgebers für sich die Situation ähnlich bewerten würden.

3.3.2 Der VPN-Server als verschwiegene Zwischenstation

Das Virtual Private Network (VPN)

Virtual Private Networks (VPNs) werden von Unternehmen häufig verwendet, um Mitarbeitern an entfernten Standorten überall auf der Welt die Anmeldung und den Zugriff auf das Unternehmensnetzwerk zu ermöglichen. Das VPN baut einen sicheren, verschlüsselten "Tunnel" für Datenpakete auf, die über das Internet zwischen dem Computer eines Mitarbeiters und dem Firmennetzwerk übertragen werden. Digitale Wegelagerer in einer feindlichen Umgebung, die die Datenpakete durchlaufen müssen, können die ausgetauschten Daten weder lesen noch manipulieren. Im Anwendungsfall dieses Buches wird das Virtual Private Network jedoch nicht als Netzwerk betrieben. Wir werden lediglich eine Rumpf-Version verwenden, die auf den Tunnel und den VPN-Server reduziert ist. Der VPN-Server fungiert als unser Strohmann. Er entschlüsselt unsere Internet-Anfragen, leitet sie an den vorgesehenen Server weiter, empfängt die Antworten, verschlüsselt sie für uns und sendet die verschlüsselten Antworten an unseren Computer zurück.

Natürlich kennt der VPN-Server unsere IP-Adresse. Sie wird mit und auch in unseren verschlüsselten Anfragen übertragen. Der VPN-Server ersetzt jedoch unsere IP-Adresse in den Anfragen durch seine Adresse in den entschlüsselten Paketen und leitet sie dann im Klartext an die empfangende Ziel-Webseite weiter. Falls wir einen HTTPS-geschützten Austausch angefordert haben, ist die Nutzlast der Datenpakete idealerweise für den VPN-Server unzugänglich, da sie in der zwischen unserem System und der empfangenden Ziel-Webseite ausgehandelten Weise mit TLS[16] verschlüsselt ist.

[13] siehe: https://www.zeit.de/digital/datenschutz/2018-05/bnd-ueberwachung-de-cix-internetknoten-klage

[14] https://netzpolitik.org/2015/how-the-german-foreign-intelligence-agency-bnd-tapped-the-internet-exchange-point-de-cix-in-frankfurt-since-2009

[15] https://www.datacenterdynamics.com/en/news/de-cix-files-constitutional-complaint-over-state-surveillance-german-data-center/

[16] Siehe den Glossareintrag zu TLS.

7. Glossar

Anonym/nonym Anonym ist ein Wort mit griechischen Wurzeln und bedeutet, dass etwas nicht mit Namen oder Unterschrift versehen ist. Hierbei wird die Verneinung durch das vorangestellte "a" ausgedrückt und nonym steht für "mit Namen versehen". Ganz Ähnliches findet man bei den Wortpaaren sozial/asozial, symmetrisch/ asymmetrisch, septisch/aseptisch, periodisch/aperiodisch usw...

In der Informatik ist leider für den Vorgang des Brechens der Anonymität die missratene Wortschöpfung Deanonymisierung üblich. Für das nicht anonym sein gibt es gar kein Wort. Ich verwende deshalb dafür im Buch das Adjektiv nonym, also griechisch "mit Namen versehen".

Das in der Informatik verbreitet genutzte Wort deanonymisieren ist unschön, weil es in griechisch-lateinischem Mix eine doppelte Verneinung enthält. Ich halte das in der Literatur bisher nicht verwendete Wort nonymisieren für besser und verwende es auch konsequent in diesem Buch. Niemand, der bei klarem Verstand ist, würde schließlich das Wort deasymmetrisch für eine symmetrische Figur oder das Wort deaperiodisch für die periodische Umdrehung der Erde verwenden.

Bit Die kleinste Einheit der Information in einem klassischen Computer. Ein Bit ist die Informationsmenge, die sich daraus ergibt, dass eine Speicherzelle den Wert 0 oder 1 haben kann.

Byte Die Informationsmenge von 8 Bit. Ein Byte wird in einfachen Textdateien üblicherweise für die Codierung eines Zeichens verwendet. Es gibt $2^8 = 256$ verschiedene Möglichkeiten, ein Byte zu besetzen. 64 dieser Möglichkeiten werden in der Radix64-Codierung verwendet und entsprechen druckbaren Zeichen.

BS Die Abkürzung wird in dieser Version des Buches nicht mehr verwendet. Sie dürften ihr aber bei dem in einigen Übungen nahegelegten Forenbesuch häufiger begegnen. BS steht für das englische Wort "Bullshit". Es ist ein etwas deftigeres Wort für den deutschen Ausdruck "Blödsinn". Wenn ich es möglichst eng am Original übersetzen sollte, würde ich die Formulierung "gequirlter Bockmist" wählen. Als Beispiel möchte ich einen Vortrag auf der CCC-Tagung in Hamburg 2013 nennen, in dem mit Referenz auf DE-Mail der Titel "Bullshit

made in Germany"[1] gewählt wurde. DE-Mail ist nicht mit Ende-zu-Ende-Verschlüsselung ausgestattet, war aber von den quasistaatlichen Herausgebern als sicher beworben worden.

Chiffrat Dies ist die Bezeichnung für das Produkt eines Verschlüsselungsvorgangs. In der Regel ist es eine Datei, die nur mit Kenntnis des gültigen Schlüssels in eine lesbare Datei zurückgewandelt werden kann.

Client Dies ist ein Rechner, der auf einen im Internet von einem anderen Rechner bereitgestellten Service zugreift. Manchmal wird der Ausdruck Client auch für das beim Zugriff aktive Programm verwendet. So kann z. B. auch der Browser, mit dem eine Internetseite besucht wird, als Client und ein Mail-Programm wie Thunderbird, Outlook oder Claws als E-Mail-Client bezeichnet werden.

Darknet Das Darknet besteht aus Webseiten, die nur über ein Anonymisierungsnetzwerk zu erreichen sind. Dort sind sowohl die publizierenden Server als auch die abrufenden Clients durch das Anonymisierungsnetzwerk geschützt. Das Tor-Darknet ist das umfangreichste und bekannteste Darknet. Seine Internetadressen enden auf ".onion".

Diffie-Hellman-Schlüsselvereinbarung auch DH-Schlüsselvereinbarung, engl. DHKE (Diffie Hellman Key Exchange). Dies ist ein Verfahren, mit dem zwei Parteien über einen offenen, unverschlüsselten Kanal einen gemeinsamen geheimen Schlüssel vereinbaren können, ohne dass ein Angreifer, der den Nachrichtenaustausch belauscht, Kenntnis des Schlüssels erlangen kann. Die Entdeckung dieses Verfahrens durch Ralph Merkle, Whitfield Diffie und Martin Hellman im Jahr 1976 öffnete den Weg in die modernen Public-Key-Kryptografieverfahren. Das ElGamal-Verschlüsselungsverfahren ist eine zeitlich leicht umgeordnete Form der Diffie-Hellman-Schlüsselvereinbarung.

DSA steht für Digital Signature Algorithm und bezeichnet die durch US-Behörden standardisierte Version des ElGamal-Verfahrens für die digitale Signatur. Das Verfahren ist recht komplex. Alle Internetnutzer verwenden es, aber wenige Menschen verstehen es. Als vor gut zehn Jahren bei dem Besetzungsverfahren für eine Professur in IT-Sicherheit an meiner Hochschule das Thema digitale Signatur für die Probevorlesung der Bewerber gewählt wurde, behandelte keiner der Bewerber das DSA-Verfahren (oder ElGamal-Signaturen). Alle beschränkten sich zu meiner Enttäuschung auf die intellektuell weniger fordernden RSA-Signaturen. Das Verfahren ist aber ein mathematisches Juwel und ich habe größten Respekt vor Taher Elgamal, der es 1985 entdeckte. In einem meiner Youtube-Videos[2] versuche ich mich an einer (hoffentlich) verständlichen Erklärung von ECDSA, einer Variante für DSA-artige Signaturen mit Elliptischer-Kurven-Algebra.

Ende-zu-Ende-Verschlüsselung auch E2E-Verschlüsselung. Hierbei haben der Nachrichtensender (das eine Ende) und der Nachrichtenempfänger (das andere Ende) die alleinige und manuelle Kontrolle über die Auswahl des Schlüssels, manchmal sogar über die Auswahl des Verschlüsselungsverfahrens. Niemand sonst kennt den Schlüssel. Jede bei der Nachrichtenübermittlung beteiligte Stelle außerhalb der Gehäuse der Endcomputer hat nur Zugriff auf das Chiffrat, sieht weder Schlüssel noch Klartext. Derartig verschlüsselte Nachrichten sind sicher vor staatlichen und kriminellen Zugriffen, sofern das Verschlüsselungsverfahren sicher ist und die privaten Endcomputer sicher sind. Sie müssen noch unter voller und alleiniger Kontrolle der jeweiligen Nutzer stehen.

Will ein Staat trotz E2E-Verschlüsselung mitlesen, so muss er die Nutzercomputer hacken, sozusagen in das Computergehäuse eindringen. In ihrem furchtbaren Dialekt der deutschen Sprache nennen die Behörden das "Quellen-TKÜ".

Viele Mailanbieter und Chat-Dienste werben damit, E2E-Verschlüsselung anzubieten. Solche Behauptungen sind Marketinglügen. Die Server von E-Mail- und Chat-Diensten sind Stellen

[1] siehe: https://www.ccc.de/de/updates/2013/bullshit-made-in-germany

[2] https://www.youtube.com/watch?v=5mNim4ZkWLg

zwischen den Enden. Sie sind für sicheren Transport und Bereithaltung von Daten zuständig und nur dafür. Sie und ggf. deren Brückenköpfe in unseren Computern haben keinen Zugriff auf Nachrichteninhalte und deren Verschlüsselung zu haben. Wenn diese externen Stellen an der Verschlüsselung in irgendeiner Weise beteiligt sind, handelt es sich nicht um E2E-Verschlüsselung oder zumindest um eine sehr kreative Interpretation dieses Begriffes.
Unabhängig von allen technischen Details wäre es ohnehin eine dumme Idee, einen Dienst mit der Verschlüsselung unserer Daten zu betrauen, der von staatlichen Stellen zum Interesse an unseren Daten angehalten werden kann und sogar selbst ein wirtschaftliches Interesse an unseren Daten hat.

Firmware Firmware ist Software, die nicht Teil des im Idealfall vom Nutzer beherrschten Betriebssystems ist, sondern auf im Computer verbauten Chips vorliegt. Komplexe Chips können in der Firmware so etwas wie ein eigenes Chip-Betriebssystem haben. Dies ist in aller Regel proprietäre Software, die gesondert gegen Manipulation von außen geschützt ist. Sie kann bei Bedarf kaum ohne Mitwirkung des Chipherstellers verändert werden.

Fork eines Software-Projektes. Bei quelloffener Software kann jeder Interessent den Quellcode herunterladen, ansehen und seine Kopie auch nach Gutdünken modifizieren. Damit kann auch jeder Entwickler oder jede Entwicklergruppe die ggf. modifizierte Software kompilieren, linken und dadurch eine installierbare Programmdatei erstellen. Wenn der Entwickler dies in der Absicht tut, die Software unter eigener Regie weiterzuentwickeln, zu pflegen und den Code wiederum der Allgemeinheit im Netz zur Verfügung zu stellen, so erstellt er einen Fork bzw. dann hat er das Projekt geforkt (unglücklich formulierte Teileindeutschung von "gegabelt").
Ein prominentes Beispiel finden wir im Bereich der Office-Software. Das Office-Paket LibreOffice ist ein Fork des ursprünglichen Projektes OpenOffice.

Hashwert Dies ist der Ausgabewert einer kryptografischen Hashfunktion. Er ist vergleichbar mit der digitalen Version eines Fingerabdrucks einer Datei oder eines Textschnipsels. Aus dem Hashwert kann keine Information über das Urbild gewonnen werden. Es ist für Angreifer nicht möglich, für einen gegebenen Hashwert ein Urbild zu produzieren, das diesen Hashwert liefert (Preimage Resistance). Es ist nicht möglich, zu einem Urbild ein zweites Urbild zu finden, das den gleichen Hashwert liefert (Second Preimage Resistance). Es ist nicht möglich, zwei verschiedene Urbilder zu finden, die den gleichen Hashwert liefern (Collision Resistance). Gängige aktuelle Hashfunktionen wie SHA2, SHA3, Skein etc. erfüllen diese Kriterien. Eine Hashfunktion, die eines dieser Kriterien nicht oder nicht mehr erfüllt, gilt als gebrochen.

HTTPS steht für "Hypertext Transfer Protocol Secure". Im Gegensatz zu einer HTTP-Verbindung kommuniziert ein Server bei HTTPS authentisiert und verschlüsselt mit dem Client (dem Internet-Browser). Durch die öffentlich verankerte Public Key Infrastructure (PKI) wird die Zugehörigkeit des vom Server dem Client präsentierten öffentlichen Schlüssels zur vom Server gehosteten Domain nachgewiesen. Mithilfe dieses öffentlichen Schlüssels wird eine verschlüsselte Verbindung zwischen dem Browser auf dem Nutzercomputer und dem Server aufgebaut. Bei ordnungsgemäßer Funktion kann der Internetnutzer sicher sein, tatsächlich unmittelbar mit dem gewünschten Partner (Bank, Google, Behörde etc.) verschlüsselt zu kommunizieren. Dritte Parteien, die diese Kommunikation auf der Strecke mitschneiden, können die ausgetauschten Daten nicht mitlesen. Sie können auch nicht feststellen, welche Unterseiten der Internetnutzer innerhalb der besuchten Domain angesehen hat. Wenn Ihr Leben oder Ihre Freiheit von der Vertraulichkeit der Internetkommunikation abhängt, sollten Sie sich nicht auf die Sicherheit von HTTPS verlassen. Sie erhalten bei HTTPS regierungskonform dosierte IT-Sicherheit, die regelmäßig von staatlichen Diensten und Kriminellen unterlaufen wird.

Jabber Freies Chat-Format, das Chat-Kommunikation möglich macht, die im Gegensatz zu kommerziellen Chat-Formaten nicht an spezielle Server gebunden ist. Das Format wurde in XMPP umbenannt, aber der Name Jabber wird umgangssprachlich noch gerne verwendet.

KDF Steht für Key Derivation Function. Dies ist eine Funktion, die auf ein vom Nutzer eingegebenes Passwort angewandt wird, um den Ausgabewert dieser Funktion anschließend direkt als Schlüssel eines symmetrischen Verschlüsselungsverfahrens zu verwenden oder bei einem Login mit einem abgespeicherten Zugangswert zu vergleichen und bei Übereinstimmung Zugang zu gewähren. Zweck der KDF ist es, die Passwortverarbeitung durch Salting zu individualisieren und durch Stretching zu verlangsamen. Dadurch werden Attacken erschwert.

Konsole/Terminal Ein Gerät oder ein Programm, mit dem der Computer über Texteingaben an der Tastatur und Textausgaben an einem Bildschirm gesteuert und administriert werden kann. Vor der Einführung grafischer Benutzeroberflächen war dies das Standardverfahren für die Eingabe von Kommandos am Computer. Bei Verwendung grafischer Benutzeroberflächen in modernen Betriebssystemen öffnen wir dafür ein Programm in einem Fenster, das ein althergebrachtes gegenständliches Terminal simuliert.
Um auf einem Linux-Desktop ein Terminal zu öffnen, finden Sie im Anwendungsmenü meist einen Eintrag namens *Terminal Emulator*. Bei neueren Windows-Versionen nennt Microsoft die primäre Konsolenanwendung etwas hochtrabend *PowerShell*. Es gibt unter Windows auch eine einfache, ursprünglichere Konsole namens *Command Prompt*.

MITM-Attacke MITM steht für "Man in the Middle", deutsch der Mann in der Mitte. Bei diesem Angriff gibt sich eine feindliche Instanz als der intendierte Kommunikationspartner eines Internetnutzers aus. Bei erfolgreichem Angriff nimmt der Initiator den verschlüsselte Kontakt arglos mit dem Man in the Middle auf, der sich dann seinerseits gegenüber dem intendierten Kommunikationspartner als der Initiator der Kommunikation ausgibt und mit dem Empfänger einen zweiten verschlüsselten Kontakt aufbaut. Die rechtmäßigen Kommunikationspartner glauben, vertraulich zu kommunizieren. In Wirklichkeit laufen aber alle Informationen beim Man in the Middle im Klartext zusammen. Der Man in the Middle kann die Nachrichten lesen und bei Bedarf auch manipuliert weiterleiten.

Negligible Adversary Advantage Im Idealfall ist aus einem Chiffrat nichts, kein einziges Bit an Information, extrahierbar. Das Chiffrat soll für den Angreifer nicht von einer (ungefähr) gleich großen Zufallszahlendatei unterscheidbar sein. In der Fachsprache heißt dieses Kriterium "Negligible Adversary Advantage". Traditionelle Chiffratformate hybrider Verschlüsselung, z. B. OpenPGP, erfüllen dieses Kriterium nicht und geben häufig sogar eine Empfänger-ID preis.

Nonce Nonce ist in der IT-Sicherheit ein Begriff für "number only used once". Eine meist lange Zahl, die nur einmal verwendet wird. In der Regel wird schlicht eine Zufallszahl genommen, weil eine Buchführung über bereits verwendete Zahlen kaum praktikabel wäre. Bei ausreichender Länge der Zufallszahl ist die Wahrscheinlichkeit einer Wiederholung praktisch vernachlässigbar.

Nonym/anonym –> Siehe anonym.

Onion Service Dienst, den ein Rechner als Server, anonymisiert über Tor, anderen Tor-Nutzern zur Verfügung stellt. Die Verbindung zum Client wird über einen Rendezvouspunkt im Tor-Netzwerk geknüpft.

Public-Key-Infrastruktur (PKI) Die PKI ist eine hierarchische Baumstruktur von Zertifizierungsstellen, bei der in der Hierarchie höherstehende Einheiten (engl.: certification authorities) durch ihre digitale Signatur die Gültigkeit und Echtheit der Zertifikate (öffentliche Schlüssel mit Zusatzinformation) hierarchisch niedriger stehender Einheiten zertifizieren. Gelegentlich können Kriminelle sich im Zuge eines Hacks einer hierarchisch hochstehenden

Einheit gefälschte Zertifikate beschaffen. Dies richtet regelmäßig größere Flurschäden im allgemeinen Sicherheitsgefüge des Internets an.

Pwned Das Wort könnte Ihnen bei Forenbesuchen begegnen, kommt jedoch in diesem Buch nicht vor. Dies ist im Englischen ein umgangssprachliches Wort für übernommen oder gehackt. Das Wort ist wahrscheinlich auf einen einfachen Tippfehler zurückzuführen, da "p" und "o" auf der Tastatur nebeneinanderliegen (owned -> pwned). Wenn ich Ihr System gehackt habe, erfolgreich war und administrativen Zugriff erhalten habe, kann ich Ihr System verdeckt oder offen kontrollieren. Dann ist Ihr Computer pwned.

Radix64-Codierung Darstellung von Daten im 64er System, wobei 64 druckbare Zeichen verwendet werden. Hiermit können beliebige Dateien gut über textorientierte digitale Kanäle übertragen werden. Es gibt auch die Variante Radix32, die nur 32 druckbare Zeichen verwendet.

Repository Bei Linux-Distributionen erhält man mit der Installation zusätzlich zum nackten Betriebssystem zunächst eine bestimmte Softwareumgebung. Ein Browser, ein Mailer, ein Office-Paket etc. wird gleich mitinstalliert. Zusätzlich gibt es äußerst umfangreiche vorsortierte Kollektionen mit freier Software, von der in einem Paketmanager auf Wunsch Komponenten durch einen Klick zugewählt werden können und deren Verträglichkeit mit anderer installierter Software vom Herausgeber der Distribution geprüft ist. Bisherige Nutzer proprietärer Betriebssysteme sind üblicherweise überrascht und sehr erfreut, wenn Sie diesen Komfort erstmalig erfahren. Solche Kollektionen nennt man Repositories.

Router Ein Computer mit der Spezialaufgabe, eingehende Datenpakete an andere Router weiterzuleiten, die in optimaler Weise die Datenpakete in Richtung der intendierten Bestimmungsstelle oder sogar direkt an die Bestimmungsstelle weitergeben können.

Server Ein Rechner, der für andere Rechner im Internet (für seine Clients) einen Service bereitstellt.

TLS steht für "Transport Layer Security" und bezeichnet ein Verschlüsselungsprotokoll für das Internet. Hierbei liegt eine Verschlüsselung vor, die auf unterer Ebene verankert ist (sozusagen im Maschinenraum des Datenaustauschs) und kaum merklich für den auf oberer Ebene agierenden Nutzer automatisch im Hintergrund abläuft. Es handelt sich ausdrücklich nicht um Ende-zu-Ende-Verschlüsselung, sondern um Verschlüsselung für die Strecke der Daten zwischen bestimmten Stationen auf der Datenverbindung, beispielsweise zwischen kommunizierenden E-Mail-Servern oder zwischen E-Mail- oder Chat-Server des Diensteanbieters und Client auf dem Computer des Nutzers. Die wohl sichtbarste Anwendung von TLS ist die Verschlüsselung bei durch HTTPS geschütztem Zugriff auf eine Webseite.

Tor ist ein Anonymisierungsnetzwerk für Internetnutzer. Es ist ein US-amerikanisches Projekt und wird unter anderen auch vom US-Außenministerium finanziell unterstützt. Dies erweckt bei vielen Computernutzern Misstrauen. Durch die Quelloffenheit der Tor-Software kann aber solches Misstrauen weitgehend ausgeräumt werden. Technisch ist Tor anderen heutigen Anonymisierungswerkzeugen deutlich überlegen.

VM steht für "Virtual Machine", auf Deutsch eine Virtuelle Maschine. Dies ist ein virtueller Rechner, der nicht wirklich physisch vorliegt. Der Rechner wird nur in einem anderen Rechner simuliert. Eine VM setzen wir ein, um beispielsweise Software für ein Betriebssystem (z. B. Windows 10) auf einem anderem Betriebssystem (z. B. Linux) zu entwickeln und diese in dem simulierten Windows-Rechner auch zu testen.

Eine andere Anwendung besteht darin, Internetkommunikation nur von einem gegenüber dem Hauptsystem isolierten simulierten Rechner ausführen zu lassen. Nach der Kommunikation kann man den nur zur Kommunikation verwendeten simulierten Rechner löschen und bei Bedarf einen frischen simulierten Rechner wieder im Ausgangszustand erzeugen. Eventuell in der Kommunikation eingeschleuste Malware ist damit in aller Regel unschädlich gemacht.

VPN steht für Virtual Private Network. Dies ist ein privates Netz innerhalb der Internet-Infrastruktur, über das für Außenstehende unzugängliche Kommunikation stattfinden kann. Eine Standardverschlüsselung ist gegenüber dem Nutzer so verkapselt, dass der Nutzer die Anwesenheit der Verschlüsselung im Normalbetrieb gar nicht wahrnimmt und er scheinbar normalen Internetverkehr mit den anderen Teilnehmern am VPN hat. Aus den Snowden-Enthüllungen ging hervor, dass der US-Spionagedienst NSA verdeckt Zugang zu vielen US-externen Unternehmens-VPNs erlangen konnte.

Workaround Dies ist ein Weg, einen Fehler nicht zu beseitigen, sondern nur zu umgehen, um den Computer oder ein Programm dennoch sinnvoll nutzen zu können.

XMPP -> Siehe Jabber.

Zero day exploit auch kurz Zeroday genannt. Ein Programmfehler, der die Übernahme der Kontrolle über einen fremden Rechner durch einen Angreifer erlaubt und der erst seit null Tagen - also noch nicht - öffentlich bekannt ist. Staatliche Überwachungsbehörden kaufen neu gefundene Zerodays zu hohen Preisen auf, um diese ihrem digitalen Waffenarsenal hinzuzufügen. Sie wollen damit die Fähigkeit erwerben und pflegen, die Rechner von als feindlich oder gefährlich betrachteten Personen oder Organisationen verdeckt zu übernehmen, um dadurch Spionage oder Sabotage zu betreiben.

Index