Gegenmassnahmen

- Passwörter
- Datenhygiene
- Kommunikations-Verschlüsselung
- Wifi-Sd_Cards
- Backups

Passwörter

Wovor schützen gute Passwörter ? Und wovor nicht ?

Prinzipiell sind gute Passwörter natürlich unvermeidbar. Was ein gutes Passwort ist, behandeln wir weiter unten. Es soll aber schon darauf hingewiesen sein: Passwörter hindern Behörden fast nie davor, in ungesicherte Social-Media Accounts wie Instagram, Twitter, Reddit, Tiktok und so weiter, hinein zu kommen. Dafür reicht ihnen ein richterlicher Beschluss, denn dort liegen eure Daten unverschlüsselt und deshalb brauchen sie dafür euer Passwort nicht.

Generell gilt

- Passwörter nicht wiederverwenden
- Starke Passwörter verwenden
- Einen Passwort Manager benutzen
- 2-Faktor-Authentifizierung

Passwortmanager

Tip: Welchen Passwortmanager?

Lies mehr dazu in den Empfehlungen zu Passwortmanagern.

Ein Passwortmanager speichert alle Passwörter in einer, mit einem Hauptpasswort, verschlüsselten Datenbank. Dadurch liegen eure Passwörter nicht einfach in Klartext auf eurem System und ihr müsst sie euch nicht alle selbst merken.

Da ihr euch Passwörter nicht mehr selbst merken müsst, ist es kein Problem und auch empfohlen, dass ihr für jeden Account ein eigenes, starkes Passwort generiert, was mit dem Passwortmanager selbst sehr einfach zu machen ist.

Der Passwortmanager speichert dann auch die Zuordnung zu Webseiten & Apps, für die ihr das jeweilige Passwort generiert habt. Das erschwert so auch Phishing, weil das Passwort auf einer falschen URL nicht als Vorschlag angezeigt wird.

Wie oben schon erwähnt ist der Passwortmanager selbst durch ein starkes Hauptpasswort, und/oder andere Faktoren geschützt (s. unten 2-Faktor-Authentifizierung). Dies ist damit das einzige Passwort, das ihr euch wirklich merken müsst und kann dementsprechend auch etwas komplexer sein, denn es gilt: lieber ein starkes Passwort merken, als viele unsichere (und wahrscheinlich sehr ähnliche) Passwörter.

Starke Passwörter

Okay, aber zumindest ein starkes Passwort für den Passwortmanager braucht ihr ja trotzdem...

Tip

Wie du ein starkes Passwort mithilfe von Diceware einfach erstellen kannst erklären wir dir übrigens hier.

Wann ist ein Passwort denn stark? Eine Wichtige Grundvoraussetzung ist, dass das Passwort zufällig generiert ist. Alles was du dir ausdenkst, egal wie clever dein System sein mag, ist als unsicher zu betrachten. Deine Passwörter sollten also zufällig generiert sein. Eine Möglichkeit dazu ist ein Passwortmanager, eine weitere ist Diceware zeigen wir weiter unten.

Um zu klären wie ein sicheres Passwort aussehen muss, wenn es zufällig generiert wurde, schauen wir uns an wie lange es dauert ein Passwort zu cracken.

Zeit zum cracken eines Passworts

Tatsächlich kommt es sehr auf die genauen Umstände an. Die Berechnungen hier nehmen ein konkretes Szenario an. Das hier gezeigte Szenario geht von relativ guten Konditionen für die Angreifer aus. Das heißt, in der Praxis dauert es eher noch länger.

Technische Details

Wir gehen von einem MD5 gehashten Passwort aus und davon, dass die Angreifer die Hardware zur Verfügung haben die für das Training von ChatGPT verwendet wurde: 10000 NVIDIA A100 GPUs.

Kaufpreis: ca. 9000€ pro Stück für die günstigere Variante mit 40GB Speicher. Insgesamt also 90 Mio. Euro. Auch zur Miete ist diese Masse an Hardware auf Dauer nicht günstiger. Weitere details zum Szenario gibt es bei hive-systems welche die Berechnungen durchgeführt haben.

Zudem ist bei den Zeiten zu bedenken, dass diese für *ein* Passwort von *einer* Person sind. Die komplette Hardware ist damit beschäftigt, es kann währenddessen kein anderes Passwort gecrackt werden.

Wichtige Voraussetzung: Das Passwort muss zufällig generiert worden sein! Das heißt hier geht es um reines Character-Bruteforcing, also ohne auf die Zielperson optimierte Wortlisten.

a table shows the amount of time to password-cracking, according to above described scenario

Zeit zum cracken einer Passphrase

Ein zufälliges, ausreichend langes Passwort aus Buchstaben, Zahlen und Sonderzeichen ist jedoch für Menschen schwer zu merken. Deshalb empfehlen wir für die Passwörter die ihr euch merken müsst, beispielsweise das für für den Passwortmanager, Passphrasen zu verwenden. Diese bestehen aus Wörtern statt aus einzelnen Buchstaben. Damit können Menschen deutlich besser umgehen, sie sind aber nicht weniger sicher Passwörter. Siehe auch: xkcd 936

Technische Details

In der Informationstheorie muss zur Bewertung der Sicherheit immer angenommen werden, dass der Angreifer weiß nach welchem Verfahren wir das Passwort gebildet haben. Daher verwendet der Angreifer hier eine Wordlist-Attack. Ansonsten bleibt alles gleich.

Diceware: Das Erstellen zufälliger Passphrasen kann, wie schon erwähnt, mit Passwortmanagern geschehen, oder ganz analog mit Würfeln und einer möglichst großen Wortliste.

Info

Die Passphrase muss zufällig generiert worden sein. Beispielsweise mit Würfeln und Wortliste (Diceware), oder der jeweiligen Funktion des Passwortmanagers.

a table shows the amount of time to passphrase-cracking, according to above described scenario

2-Faktor Authentifizierung

2FA sorgt dafür, dass das bloße eingeben eines Passworts, nicht als vollständige Autorisierung genügt, da davon ausgegangen wird, das Passwörter eventuell korrumpiert sind. Deshalb wird eine zweite Instanz zur vollständigen Autorisierung angefordert.

In den Empfehlungen zu Passwortmanagern findet sich ein Beispiel Szenario, wie z.B. ein KeePassXC-Datenbank mit einem zweiten Faktor abgesichert werden kann.

Das kann auf verschiedenen Kategorien beruhen:

- Wissen: Der alte Klassiker in Form eines Passworts oder Sicherheitsfragen wie "Wie lautet ihr Geburtsort?"
- Besitz: Ihr benötigt ein spezielles Ding, dass euch entweder eine Nummer anzeigt, oder was per USB in den Rechner gesteckt werden muss. Besitzt der/die Angreifer*in dieses "Ding" nicht, erfolgt auch keine Autorisierung. (Hardware-Token, *2FA-Apps*, *SMS*)
- Sein: Einzigartige biometrische Eigenschaften müssen verifiziert werden. (Biometrie)

Im Folgenden werden verschiedene Technologien aufgelistet, die für 2-Faktor Authentifizierung (2FA), aber auch als einfache 1-Faktor-Authentifizierung, genutzt werden können:

- Hardware-Token mit USB: Sie sehen aus wie normale USB-Sicks. Soll ein damit konfigurierter Service/Festplatte o.ä. entsperrt werden, muss auch dieser Stick in das genutzte Gerät gesteckt werden. Oftmals sind diese Token wiederum mit einem PIN geschützt, sodass es nicht reicht diesen zu klauen. Die PIN-Eingabe ist dabei oftmals auf x versuche limitiert. Da dies alles auf Hardware-Ebene umgesetzt und geschützt wird, ist es eine relativ sichere Möglichkeit der Authentifizierung. Der relevante Standards für Securitytoken dieser Art heißt FIDO2, der alte Standard U2F.
- Hardware-Token mit Screen: Diese USB-Stick großen Geräte haben einen kleinen Bildschirm, auf ein x-stelliger Code angezeigt wird (meist 4-6 stellig). Sie können mit bestimmten services verknüpft werden. Diese Services verlangen dann bei jeder Anmeldung (neben dem Passwort) auch den code, der gerade in diesem Moment auf dem Token angezeigt wird. Die Standard für Token dieser Art sind nicht Open-Source, weshalb wir dazu raten diese nicht zu verwenden.
- TOTP (2FA) Apps: Diese Apps können ebenfalls mit verschiedenen Services verknüpft werden und generieren dann für jeden Service jeweils alternierende Security-Codes.
- Biometrie: Schon lange berühmt in Hollywood. Soll der entsprechende Service entsperrt werden, fordert er eine Biometrische Verifizierung der Nutzenden (Fingerabdruck, Gesichtserkennung, Iris-Scan, Handabdruck, Stimmerkennung etc)
- SMS: Die wohl bekannteste Methode sind 2FA SMS. Zur Verifizierung der Identität der Nutzenden sendet der jeweilige Service eine SMS an die mit dem Account registrierte Telefonnummer. Da das Mobilfunknetz nicht als sicher zu betrachten ist, raten wir hiervon ab.

Biometrie

Biometrie wie Fingerabdrücke oder Gesichtserkennung sind nachweislich fälschbar. Wie einfach das geht hat Starbug vom CCC, bereits für Fingerabdruck-, Gesichts-, Iris- und Venenerkennung gezeigt.

Abgesehen davon können Behörden oder Cops euch zwingen Dinge mit Biometrie zu entsperren. Zur Herausgabe von Passwörtern dürfen sie das nicht.

Der wichtigste Punkt hierbei ist aber wohl, das ihr eure biometrischen Merkmal nie wieder ändern könnt. Ein korrumpiertes Passwort kann zurück gesetzt werden. Ein Fingerabdruck, oder das

Gesicht jedoch nicht.

Fazit

Daher bietet Authentifizierung mit Biometrie keinen guten Schutz gegen Sicherheitsbehörden.

Datenhygiene

Egal ob bei TKÜ, digitaler Forensik oder Hausdurchsuchungen: Es geht immer um Daten, aus denen euch potentiell ein Strick gedreht werden soll. Deshalb gehört es dazu, sich die Frage, welche Daten wirklich notwendig sind, regelmäßig zu stellen.
Natürlich bereitet das ein bisschen mehr Aufwand, Daten tatsächlich zu vernichten und erfordert vor allem Disziplin.

- Brauchen wir für dieses Treffen ein Protokoll?
- Muss das Handy mit auf die Aktion?/Habe ich vor das Handy für irgendwas zu benutzen?
- Muss ich Freund*innen schreiben, welche coole Aktion ich gerade gemacht habe?
  - Selbstdarstellung ist schon vielen zum Verhängnis geworden
- Wenn alle beim Treffen waren, braucht es für manches vielleicht kein Protokoll

Wenn es keine Daten gibt, kommt auch niemand dran. Allerdings kann die Einschätzung einiger weniger, dass die Recherche-Unterlagen jetzt veraltet sind und vernichtet werden können, ein paar Jahre später schwer bereut werden. Der berühmte Aktenordner unter dem Bett wäre aber vielleicht zu riskant gewesen. Wie also Daten sicher aufbewahren? Auf Papier auf jeden Fall nur in den wenigsten Fällen!

Was aber, wenn "euch selbst belastendes Material" entstanden ist? Weg mit dem Mist. Den meisten dürfte jedoch bekannt sein, dass das einfache Löschen von Dateien keineswegs heißt, dass Daten unwiderruflich verschwunden sind. Noch nicht einmal wenn Windows euch warnt, dass mit dem leeren des Papierkorbs aber nun wirklich alles für immer in einem schwarzen Loch verschwindet.

Daten sicher löschen

Um zu veranschaulichen was passiert, wenn Dateien "normal" gelöscht werden gibt es eine Metapher:

Info

Das folgende Szenario gilt aus technischer Perspektive nur bedingt für gängige Arten von Speichern, bspw. für klassische HDD-Festplatten! Bei Flashspeichern wie bspw. SD-Karten, USB Sticks oder SSD's gibt es noch zusätzliche Dinge zu beachten. Mehr dazu unter "Besonderheiten" weiter unten.

Anna & Arthur's WG

Anna & Arthur wohnen in einer WG. Ihre Namen und Adresse stehen im Adressbuch (anders als beim Telefonbuch ist hier alles nach Adresse geordnet).
Die Wohnung ist das Speichermedium/Datenträger (Festplatte, USB Stick, SD-Karte, etc) und Anna & Arthur sind die Daten auf dem Datenträger. Das Adressbuch ist die Adressverwaltung des Datenträgers.
Wollt ihr euch nun Arthur auf eurem Bildschirm anzeigen lassen, gebt ihr dem PC die Adresse von Arthur. Dieser geht für euch zu besagter Adresse, holt Arthur aus seiner Wohnung und präsentiert ihn auf dem Bildschirm.
Soweit der Normalbetrieb, wenn Daten im Speicher liegen und benutzt werden.

Leider ist Arthur aber bei der letzten Aktion der Schlauchschal unter die Nase gerutscht, er wurde erkannt und muss nun schnell weg. -> Daten müssen gelöscht werden.

Klickt ihr nun auf "löschen" wandert diese Datei in den Papierkorb. Im Papierkorb ist gar nichts gelöscht; seht das einfach als einen "Noch zu löschende Dateien"-Ordner.
Also leert ihr auch den Papierkorb. Was ist nun passiert? Ist Arthur verschwunden?

Nein, ihr habt lediglich Arthurs Namen aus dem Adressbuch gelöscht. Arthur selbst sitzt noch immer auf seiner Couch und wartet, dass etwas passiert. -> Die Daten liegen physisch noch immer auf dem Datenträger. Sie sind bloß nicht mehr im Adressverzeichnis des Speichers indexiert.

Schauen die Cops nun ins Adressbuch, werden sie Arthurs Namen nicht mehr finden. Doch wenn sie einfach Straße für Straße, Haustür für Haustür absuchen, stoßen sie irgendwann auf Anna & Arthurs WG, in der Arthur immer noch sitzt.

Das führt uns zum Überschreiben mit zufälligen Bits: Anna & Arthur brauchen random Nachmieter*innen. Denn wenn ihre Genoss*innen einziehen, oder eben alles nur mit Nullen überschrieben wird, könnte das Spuren hinterlassen.

Zusammengefasst: Daten sind erst richtig gelöscht, wenn die Adressen im Speicher, auf dem sie lagen, durch andere zufällige Daten überschrieben wurden. Dieser Vorgang ist jedoch in keinem gängigen Betriebssystem (egal ob PC oder Handy) Standard, denn diese löschen nur die Adresseinträge zu den Dateien. Das erfordert also extra Aktionen.

Besonderheiten

- **Adressierung:** Bei Flashspeichern wie SD-Karten, USB Sticks oder SSDs weiß das Betriebssystem nicht genau, auf welchen exakten Bits die Daten eigentlich liegen. Eine eindeutige Verbindung zwischen physischen Bits und von außen adressierbaren Sektoradressen existiert so nicht. Deshalb können diese Bits auch nicht einfach überschrieben werden, weil gar nicht klar ist, welche denn überschrieben werden sollen.

- **Overprovisioning:** Noch dazu blockieren diese Arten von Speicher bestimmte Adressblöcke vor externem Schreibzugriff, sogenannte "Reserveblöcke". Dieses Overprovisioning hat drei Hauptfunktionen: Fehlerkorrektur, Optimierung von Schreibgeschwindigkeit und schont die Lebensdauer des Speichermediums.

Technical Details - Overprovisioning

- Fehlerkorrektur: Wenn einzelne Speicherzellen fehlerhaft werden (zum bsp. durch Verschleiß), kann der Controller auf diese Reserve zurückgreifen, um zu vermeiden, dass die Daten "kaputt" abgelegt werden.
- Schreibgeschwindigkeit: Da die Reserveblöcke bereits "leer" zur Verfügung stehen, müssen nicht immer erst Zellen gelöscht werden, um sie neu zu beschreiben. Der Controller kann so direkt auf leere Zellen zurückgreifen und sie sofort beschreiben.
- Lebensdauer: Overprovisioning vermeidet durch Rotieren der Daten auf den Speicherzellen, dass einzelne Zellen über eine sehr lange Zeit im immer gleichen Zustand bleiben. Das führt klassischerweise dazu, dass diese Zellen bezüglich ihres "an"- und "aus"-Zustandes asymmetrisch werden. Sie tendieren also eher in die eine, oder die andere Richtung zu kippen. Bei Schreibvorgängen kommt es dann zu Fehlern, weil einem Transistor, der bspw. über Jahre "an" war, nun mit einem extrem kurzen Impuls gesagt wird, dass er nun mal "aus" werden soll. Das passiert aber eventuell nicht, weil er sich schon so lange an "an" gewöhnt hat.

Deshalb reicht es hier nicht aus, mit gängigen Methoden Speicherzellen mit random Bits zu überschreiben. Damit bleiben die Reserveblöcke unangetastet, aus denen aber im Zweifel alte Daten rekonstruiert werden können. Die ATA Spezifikation bietet dafür zwei Befehle: `SECURITY ERASE UNIT` und `ENHANCED SECURITY ERASE UNIT`. Ersteres überschreibt nur mit Nullen, zweiteres mit random Bytes. Werden diese Befehle auf eine SSD angewandt, werden auch besagte Reserveblöcke überschrieben. Sowohl unter Linux als auch unter Windows finden sich dafür Kommandozeilen-Tools, die jedoch etwas *hacky* sein können. Die meisten SSD Hersteller wie Samsung, Kingston, Western Digital und Co. liefern extra dafür eigene Tools mit, derer sich bedient werden kann.

Diese Tools machen im Prinzip nichts anderes als diese Befehle auf SSDs mit ihrer eigenen (proprietären) Firmware anzuwenden.

Verschlüsselte Daten löschen

Eine effizientere Methode ist die Verschlüsselung. Das folgende gilt für sowohl für rotierende Platten (HDDs) als auch für SSDs:

Wird der Datenträger verschlüsselt, wird ein Key generiert, der im Header (~Kopfzeile) des Speichers abgelegt wird. Dabei werdet ihr aufgefordert ein Passwort für die Verschlüsselung

festzulegen. Mit diesem Passwort wiederum wird der im Header liegende Key verschlüsselt - nicht die Daten selbst.

Jede Datenlese- oder Schreiboperation der Daten wird symmetrisch mit dem Key ent-, oder verschlüsselt.

Die Bit-Zustände auf dem physischen Datenträger können auf Grund der mathematischen Eigenschaften moderner Verschlüsselungsalgorithmen nicht von random Bits unterschieden werden. Ein verschlüsselter Datenträger sieht also forensisch genau so aus wie ein zufällig beschriebener.

Um diese Daten nun wieder sicher zu löschen, muss daher nur der Key im Header des Datenträgers gelöscht und überschrieben werden. Das spart nicht nur enorm viel Zeit (dauert nur ein paar Minuten), das schont auch die Lebensdauer des Datenträgers. Ein vollständiges Überschreiben von einer 1TB HDD kann gut und gerne mal mehr als 5 Stunden dauern.

Detailliertere Infos findet ihr beispielsweise hier.

Kurzfassung

- Daten auf unverschlüsseltem Datenträger: gelöschte Daten lassen Spuren zurück, die wiederhergestellt werden können. Deshalb müssen Daten beim Löschen mit zufälligen (random) Bits überschrieben werden (am besten mehrfach).
- Daten auf verschlüsseltem Datenträger: Diese sind höchstens durch den Key in ihrem Header entschlüsselbar. Dieser Key ist mit einem Passwort gesichert. Wird nur dieser Key gelöscht und überschrieben, können Daten nicht mehr hergestellt werden.

Kommunikations-Verschlüsselung

Die Verschlüsselung jeglicher Kommunikation spielt in unseren Anwendungsfällen eine essentielle Rolle. In diesem Artikel wollen wir erklären, was mit Kommunikationsverschlüsselung gemeint ist, welche Arten es davon gibt und welche Vor- bzw. Nachteile sie haben.

Wir unterscheiden hier zwischen Transportverschlüsselung und Ende-zu-Ende-Verschlüsselung (E2EE: End-To-End-Encryption). Vorwegnehmend lässt sich schonmal sagen, dass Transportverschlüsselung nice-to-have ist, für uns aber in keiner Weise ausreicht und wir deshalb immer E2EE wollen.

Transportverschlüsselung

Transportverschlüsselung wird allgemein mit SSL bzw. TLS realisiert. Das kennt ihr zum Beispiel aus eurem Browser, wenn neben der URL ein Vorhängeschloss erscheint und vor der URI `https` steht. Kommt das nicht zum Einsatz, steht dor nur `http` (und meistens erscheint eine Warnung, dass die Verbindung nicht gesichert ist).

Um die Transportverschlüsselung zu erklären, nutzen wir unten stehende Grafik. Anna will Arthur eine Nachricht übermitteln, bspw. per Email. Das Beispiel funktioniert auch mit anderen Diensten ohne E2EE, wie: Telegram, Discord, oder Chats in Spielen. Dabei gäbe es aber nur einen anstatt zwei Servern.

Hier also das Beispiel mit Email.

Anna hat eine Email-Adresse bei dem gelben Server, hier `systemli.org`. Ihre Mail lautet also `anna@systemli.org`

Arthur hat eine Email-Adresse beim roten Server, hier `riseup.net`. Seine Mail lautet also `arthur@riseup.net`

Weil wir ja von Transportverschlüsselung reden, benutzen beide keine E2EE. Das heißt, weder hat Anna einen PGP-Key von Arthur, noch anders herum!

Die Schlüssel und Schlösser symbolisieren sogenannte **Zertifikate** (Vorhängeschlösser). Jeder Server hat sein eigenes Zertifikat, mit dem die Kommunikation mit ihm verschlüsselt (also eingeschlossen) werden kann. Nur der Server im Besitz des Zertifikats, hat auch den zugehörigen

Schlüssel.

Wenn Anna jetzt eine Mail schreiben will, holt sie sich das Zertifikat von Systemli (gelbes Schloss) und verschlüsselt damit ihre Mail. Völlig unabhängig davon, an wen die Mail am Ende gehen wird! Arthurs Empfangsadresse steht dann draußen auf dem Umschlag, wie bei normaler Post auch. Diese Mail (gelber, verschlossener Umschlag mit Schloss) geht dann zum Systemli-Mailserver (gelber Kasten).

Der Systemli-Mailserver schließt nun, die mit seinem eigenen Zertifikat verschlüsselte Mail auf und scant sie z.B. nach Spam. Vor allem schaut er sich die Empfangsadresse auf dem Umschlag an: `arthur@riseup.net`. An der Stelle hinter dem `@` erkennt der Server, an welchen Mailserver er diese Mail nun weiterleiten muss: `riseup.net` (roter Kasten). Also geht er kurz rüber zu Riseup, schnappt sich eine Kopie des ihres Zertifikats und verschlüsselt damit Annas Email wieder und schickt sie so (roter, verschlossener Umschlag mit Schloss) an den Riseup-Mailserver.

Ab hier wiederholt sich dieser Vorgang so lange, bis die Mail Arthur erreicht. Der Riseup-Server packt die Mail aus und wieder ein und schickt sie schließlich an Arthur.

Grafik Ende-zu-Ende Verschlüsselung

Problem

Hier der Verweis auf die Bedrohung Verkehrsdatenüberwachung/TKÜ.

Das Problem hierbei ist offensichtlich. Jede*r Teilnehmer*in in der Kommunikationskette kann die Mail einfach öffnen und lesen. Zusätzlich bleiben bei vielen Anwendungen (wie oben aufgezählt) die Nachrichten auf den (Mail)-Severn als Kopie liegen.

Ende-zu-Ende Verschlüsselung

Wenn ihr die Bedrohung durch Transportverschlüsselung verstanden habt, ergibt sich die Ende-zu-Ende-Verschlüsselung schon fast von selbst.

1. & 2.) Anna besorgt sich das Schloss (public-key) von Arthur. Dieser Punkt ist sehr wichtig, beachtet dazu den Absatz [TOFU]!

3.) Anna verschlüsselt ihre Nachricht mit Arthurs public-key

4.) Die Nachricht bleibt in allen Teilschritten von 4 (a-e) verschlüsselt. Lediglich die Metadaten (bspw. Absende-/Empfangsadresse) darauf sind (an allen möglichen Stellen, also auch beim Transport!) sichtbar und werden von den Servern gelesen, um die Mail weiter zu leiten.

5.) Arthur empfängt seine Nachricht. Weil die Nachricht mit seinem Vorhängeschloss verschlüsselt wurde und er gut auf seinen Schlüssel (private-key) aufgepasst hat, kann nur er die Nachricht mit seinem Schlüssel wieder entschlüsseln.

Grafik Ende-zu-Ende Verschlüsselung

TOFU ist böse

Trust On First Use

Schlüssel muss "out of band" verifiziert werden. Eine unverschlüsselte (also transportverschlüsselte) Mail ermöglicht das Austauschen der öffentlichen Keys.

Grafik machine-in-the-middle attack

Zu den Gefahren von Transportverschlüsselung siehe Verkehrsdatenüberwachung.

Wifi-Sd_Cards

Besonders Fotograf*innen stehen öfter vor dem Problem, dass ihr frisch aufgenommenes Bildmaterial unverschlüsselt auf den SD-Karten ihrer Kameras liegen, bis sie, wieder zurück am Laptop, dieses auf verschlüsselte Festplatten verschieben können.

Hinzu kommt das Problem, dass sich Speichermedien wie SD-Karten, USB Sticks und SSDs nur sehr unzuverlässig bis gar nicht sicher löschen lassen, wenn die Daten darauf unverschlüsselt waren.

Dem können sogenannte Wifi SD-Karten Abhilfe schaffen. Sie werden wie normale SD-Karten auch, einfach in den dafür vorgesehenen Slot der Kamera gesteckt. Sie speichern allerdings keine Bilder, sondern lassen sich über Wifi(-direct) mit einer App auf dem Handy verbinden und schickt jede gemachte Aufnahme sofort an das Handy, wo die Daten dann zumindest schonmal durch das Handypasswort geschützt sind.

Backups

Es sind schon viele Zeilen darüber geschrieben worden, warum Backups so enorm wichtig sind und über die Kuriosität, dass das allseits bekannt ist und sie scheinbar trotzdem niemand macht, sind schon bessere und schlechtere Witze gemacht worden.

Fakt ist
Wir brauchen Backups!

Backups vs. Datenhygiene

Das größte Problem liegt oft darin, dass wir uns gar nicht so bewusst darüber sind, was wir über die Jahre so alles an Daten ansammeln und wie wichtig diese noch für uns sind. Im Artikel zur Datenhygiene beschwören wir das Credo, so wenig Daten wie möglich anzusammeln. Doch das gilt natürlich vor allem für Daten, die nur von temporärem Nutzen sind und die sonst in irgendwelchen Ecken vergessen werden würden, bis sie bei einer Hausdurchsuchung wiedergefunden werden.

So wichtig es ist, mit kompromittierenden Daten so sparsam wie möglich umzugehen, sind wir doch alle Menschen, die in irgendeiner Weise vom Staat und seinen Institutionen abhängig sind:

Amtliche Dokumente, Krankenversicherung, Bankunterlagen, Arbeitsverträge, Zeugnisse und so weiter, sind alles Dinge, die wir im Zweifel immer mal wieder brauchen, um nicht in völliger Armut zu versinken.

Ideell aber noch viel wichtiger sind vielleicht Fotos von unseren Genoss*innen und Weggefährt*innen, Briefe von ihnen, Tagebücher, Geschenke, Erinnerungsstücke. All das sind Dinge, die unendlich schmerzen könnten, wären sie auf einmal nicht mehr da.

Nicht alles von dem oben Aufgezählten kann einfach digitalisiert werden, doch das meiste davon schon.

Nun laden wir alle dazu ein, sich vorzustellen, dass sämtliche Endgeräte, auf denen all eure Passwörter, Zugänge und Ähnliches gespeichert waren, jetzt kaputt oder weg sind. Könntet ihr das verkraften?

Datenhygiene & Backup Hand in Hand

Im oberen Abschnitt haben wir die Datenhygiene dem Backup gegenübergestellt. Hier möchten wir dafür plädieren, beide Konzepte zusammen zu denken, damit das eine vom andern profitiert.

Besonders langjährig genutzte Geräte und Accounts wie iCloud, Google Drive, WhatsApp, Dropbox usw. quellen oft über vor uralten Datenleichen, von deren Existenz niemand mehr weiß.

Wie viele Daten habt ihr so auf euren Geräten, die ihr mal nicht gelöscht habt, weil ihr dachtet: "Ganz vielleicht brauche ich das doch nochmal"?

Und so haben sich über die Jahre Dutzende, wenn nicht Hunderte GB an Daten angesammelt, die viel zu umfangreich sind, als dass ihr sie mal schnell sichten und aufräumen würdet.

Hättet ihr jedoch ein verschlüsseltes Backup all eurer Dateien, könntet ihr eure täglich genutzten Geräte viel entspannter aufräumen. Dann würdet ihr nur das Nötigste mit euch herumtragen.

Success

Es geht also auch darum, Mut zu Datenhygiene zu bekommen, indem gute Backups gemacht werden.

How To Backup?

Wir arbeiten gerade noch an einem Leitfaden für Backups für die Kategorie "Anleitungen", den wir aber auch hier unten verlinken werden.