Hochschule für Telekommunikation Leipzig
University of Applied Sciences

Hochschule für Telekommunikation Leipzig (FH)
Institut für Telekommunikationsinformatik

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science

Thema:	„Inwiefern bietet die Authentifikation ohne Passwort Vor- und Nachteile gegenüber Webanwendungen hinsichtlich Nutzerfreundlichkeit, Datenschutz und Sicherheit?“

Vorgelegt von:	Ertugrul Sener
Geboren am:	17.10.1998
Geboren in:	Berlin

Vorgelegt am:	16. September 2020

Erstprüfer:	Prof. Dr. Erik Buchmann Hochschule für Telekommunikation Leipzig Gustav-Freytag-Straße 43-45 04277 Leipzig
Zweitprüfer:	Juri Lobov T-Systems International GmBH Holzhauser Straße 1-4 13509 Berlin

1 Vorwort

Vor Ihnen liegt die Bachelorarbeit zur Forschungsfrage „Inwiefern bietet die Authentifikation ohne Passwort Vor- und Nachteile gegenüber Webanwendungen hinsichtlich Nutzerfreundlichkeit, Datenschutz und Sicherheit“.

Diese habe ich als Abschlussarbeit im Rahmen meines dualen Studiums für angewandte Informatik an der Hochschule für Telekommunikation Leipzig in Zusammenarbeit mit der Telekom Security im Chapter Software Development angefertigt. Ziel ist es vorhandene Authentifizierungsstrategien kritisch zu bewerten und eine eigene prototypische Lösung zu entwickeln, die bestmöglichst die Anforderungen an Sicherheit, Datenschutz und Nutzerfreundlichkeit erfüllt und über die selektive Auswahl aus den möglichen Authentifizierungsstrategien gleichzeitig die damit einhergehenden Nachteile minimiert.

Die Fragestellung habe ich zusammen mit meinem Hauptprüfer Prof. Dr. Erik Buchmann von der HfTL und meinem betrieblichen Vorgesetzten Juri Lobov entwickelt. Ich bin davon überzeugt, dass die Kombination aus theoretischem Forschungshintergrund und längjähriger praktischer Erfahrung mir bei einer umfangreichen Beantwortung der Forschungsfrage behiflich sein wird.

Daher möchte ich meinen Begleitern für ihre Unterstützung bei der Erarbeitung meiner Arbeit und der Betreuung danken. Ebenfalls möchte ich meinen Kollegen bei der Telekom Security danken, die mir jederzeit Ihre Unterstützung zugesichert haben.

Ich wünsche Ihnen viel Freude beim Lesen.

Ertugrul Sener
16. September 2020

Selbstständigkeitserklärung

Hiermit erkläre ich, Ertugrul Sener, dass die von mir an der *Hochschule für Telekommunikation Leipzig (FH)* eingereichte Abschlussarbeit zum Thema

„Inwiefern bietet die Authentifikation ohne Passwort Vor- und Nachteile gegenüber Webanwendungen hinsichtlich Nutzerfreundlichkeit, Datenschutz und Sicherheit?“

selbstständig verfasst wurde und von mir keine anderen als die angegebenen Quellen und Hilfsmittel verwendet wurden.

Leipzig, den 16. September 2020

Ertugrul Sener

Inhaltsverzeichnis

Abkürzungsverzeichnis

BSI Bundesamt für Sicherheit in der Informationstechnik 13, 19

HTTPS Hypertext Transfer Protocol Secure 15

OTP One time password 22

TOTP Time based one time password 23, 27

U2F Universal second factor 20

URL Uniform Resource Locator 15

W3C World Wide Web Consortium 20

WebAuthn Web Authentication 20

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

2 Einleitung

2.1 Motivation

Die Motivation für die Erarbeitung der Forschungsfrage ergab sich am 29. Juni 2020, als eine Rundmail von dem Data Breach Monitoring-Tool von Firefox die Bildschirme erreichte, welches einen neuen Incident bei Wattpad meldete. Wattpad ist eine Webseite, bei welcher Nutzer Geschichten frei für die Öffentlichkeit schreiben und publizieren können. Nun stellt sich die berechtige Frage, inwiefern diese in Relation kleine Webseite verwertbare Daten für einen Angreifer oder ein bösartiges Tool liefern kann. Dazu genügt es sich die Daten genauer anzusehen. Kompromittiert wurden neben den klassichen Daten wie Passwörtern, IP-Adressen, E-Mail-Adressen und Geburtsdaten auch sehr persönliche Informationen wie geografische Standorte, ein Kurzprofil des Nutzers, das Profilbild, die Social-Media-Profile und sogenannte Session-Tokens jeglicher Dienste des Benutzers, die mit Wattpad verknüpft waren. Entnehmen kann man dies mehreren Artikeln, bei der allerdings die Größe der Datenmengen zwischen 250 und 280 Millionen Einträgen variieren. Eine genaue Zahl liefert der Artikel der riskbasedsecurity mit 268.830.266 (nach der Entfernung von Duplikaten) kompromittierten E-Mail Addressen, die sich laut des Artikels in einer einzigen Datenbank befanden. [7] Diese Zahl ist dennoch nur als Richtwert zu verstehen, da wie bereits erwähnt das Ausmaß des Datenlecks bei Wattpad nicht in Gänze bekannt ist.

2.2 Problemdefinition

Die Identität dieser Nutzer hängt an ihrem Passwort, demnach ist die plötzliche Bedrohung für jeden Einzelnen nun allgegenwärtig. Das Passwort, welches nun im Netz ist, ist schwierig bis unmöglich aus allen (Sub-)Quellen eliminierbar. Die Identität der Personen kann nun von Angreifern genutzt werden, um z.B. finanziellen Schaden anzurichten und sich selbst zu bereichern.

Auch wenn es teilweise nicht so scheint, ist das Problem mit Passwörtern so alt wie das Internet selbst und Wattpad ist nicht das einzige Beispiel von veröffentlichten Datenlecks der letzten Jahre. Angriffsszenarien auf technischer Seite (Code-Injektionen, 0-Day-Exploits, Bruteforce-Angriffe) scheint die informationsafine Menscheit der Neuzeit gut organisiert zu bekommen, doch dann scheitert es in riesigen Unternehmen oft an der Wahl eines Passwortes einer wichtigen Einzelperson, welches dann in Kettenreaktionen zu solchen Data Breaches und größeren Attacken auf Unternehmen führen.

Bei Disskusionen im Internet, wird sich häufig aufgrund der selben (wiederholenden) Gründe gegen eine zwei Faktor Authentifizierung (2FA) und zusätzliche Sicherheiten für die Passwortwahl ausgesprochen. Die 2FA sei zu umständlich und ein Passwort-Manager aufgrund des Masterpasswortes und der Angst vor einem Massenverlust an Daten zu riskant. Das potenzielle Risiko welches durch Passwörter allerdings im Vergleich zu diesen neueren Verfahren aufkommt, wird häufig ignoriert. Individuelle Passwörter für jeden Dienst seien unmöglich merkbar und eine zwei Faktor Authentifizierung sei nicht bedienfreundlich genug um sie für jeden Dienst einzusetzen. Damit meinen Nutzer die verhältnismäßig lange Zeit, die bis zur erfolgreichen Authentifikation vergeht. Je nach Dienst gibt es nämlich keine Sessionverwaltung, sodass jede Authentifizierung auch einen zweiten Faktor (ein externes Gerät) benötigt, welches natürlich die Dauer des Loginvorgangs beeinträchtigt. Die Frage, die sich stellt ist es, ob man mit einer Kombination aus den vorhandenen vielfältigen Authentifizierungsmöglichkeiten eine beqeueme aber gleichzeitig sichere Authentifizierungsvariante schaffen kann, bei denen es dem Anwender möglich sein soll, sich gegenüber einer Webseite zu authentifizieren.

2.3 Stand der Forschung

2.3.1 Unsichere Passwörter

In der fünften Ausgabe der Zeitschrift “Wirtschaftsinformatik & Management” 2018 mit dem Titel “Schwache Passwörter - Nutzer spielen weiterhin Vogel Strauß” schrieb der Autor Geralt Beuchelt: “Der Umgang mit Passwörtern ist so ähnlich wie eine Diät: Eigentlich weiß man genau, was richtig ist - Macht aber oft genug das Gegenteil. Und nicht selten ist der Grund Bequemlichkeit. Warum selber kochen, wann nach einem langen Tag eine Pizza lockt? Und warum lange, umständliche Passwörter verwenden, wenn es einfach zu merkende, die man für alle Accounts verwendet, doch auch tun?”. [3]

Die symbolische Pizza steht für die Mehrfachverwendung von teils schwachen Passwörtern für alle genutzten Dienste inklusive des ’Verwaltungsdienstes’ wie der Mail, welches als meist einziges Identifikationsmerkmal dient, über die weitere Dienste betroffen

sein können. Der Begriff des Passwortes stammt aus dem militärischen Bereich des 16. Jahrhunderts, wobei tatsächlich das einzelne Wort gemeint war, welches einem Zutritt zu Gebäuden verschaffte. Damit verwandt ist das Kennwort, welches nicht das Passieren sondern die Kennung des gemeinsamen Geheimnisses betont. Damit ist gemeint, dass der Passierer mit einem Kennwort auch automatisch als Geheimnisträger identifiziert wird. Als allerdings Computer immer leistungsfähiger wurden, wurde der Begriff der Passphrase etabliert, um die Notwendigkeit längerer Passwörter hervorzuheben. Weitere Schlüsselwörter für das heutzutage bekannte Passwort sind: Schlüsselwort, Kodewort (u.a: Codewort) oder die Parole. Die Länge gillt gemeinhin als die allumfassende Sicherheit von Passwörtern. Dem widerspricht die klare Trennung zwischen Länge und Komplexität von Passwörtern durch das Bundesamt für Sicherheit in der Informationstechnik, die sinngemäß in ihrer Empfehlung zum Thema 'sichere Passwörter' schreiben, dass die Länge von Passwörtern nicht dessen Komplexität und Sicherheit gegen Angriffe widerspiegelt [4].

Um sichere Passwörter zu erzwingen, setzen Webseiten auf Passwort Policys. Diese machen anhand von Regeln, die sie meist selbst aufstellen, das Wählen von Passwörtern komplizierter. Dies resultiert in Passwörtern mit Mindestlängen und einem Mindestzeichensatz und weiteren Regeln wie der Verbot von sich wiederholenden Zeichenketten wie "testtest", längeren Zahlenreihenwiederholungen wie "123123" oder dem Verbot der Nutzung des Usernamen im Passwort. Das gewählte Passwort soll für einen selbst leicht merkbar, für einen Computer oder menschlichen Angreifer allerdings schwer zu erraten sein. So empfielt das Bundesamt für Sicherheit in der Informationstechnik (BSI) Passwörter zu verwenden, die möglichst nicht aus Tastaturmustern bestehen wie 'asdfgh' oder '1234abcd'. Allgemein wird ein 20 bis 25 Zeichen langes Passwort aus zwei Zeichenarten einem acht bis 12 Zeichen langem Passswort aus viert Zeichenarten in Punkto Komplexität gleichgesetzt. [4]

Das Wort 'Policy' ist in diesem Zusammenhang als 'die Regel' zu verstehen und in der Wortkombination sind Passwort Policys die Regeln, die zu einem sicheren Passwort führen. Derart Regeln gibt das BSI vor. So seien der Kreativität bei Passwörtern keine Grenzen gesetzt [4]. Zum Beispiel könne man einen leicht zu merkenden Satz nehmen, diesen mit Bindestrichen verbinden und von jedem Wort den ersten Buchstaben entfernen. Die Frage die sich dabei stellt ist es, ob dieser Satz dann die Tippgeschwindigkeit des Nutzers beeinträchtigt, weil relativ viele Denkprozesse während des Tippens stattfinden müssen. Zunächst ein Mal müsste sich hierbei der Satz in voller länge gemerkt werden, dies ist noch Recht unkompliziert für den allgemeinen Internetnutzer. Danach muss der Satz während des Tippens bereits mit Bindestrichen verbunden werden, auch dies ist noch kein großen Problem. Zum Problem wird es, sich die ersten Buchstaben beim Tippen automatisch wegzudenken, sodass einem kein Fehler unterläuft. Wenn einem doch ein Fehler unterläuft, ist es anders als bei gängigen Passwörtern, sehr schwer zur Fehlerquelle zu springen und den Fehler zu lokalisieren. Alternativ bleibt einem nur das Neutippen des Passwortes, welches eine große Zeitverzögerung für den Nutzer bedeutet und auf lange Sicht den Nutzer dazu bringen wird, ein einfacheres und leicht

tippbares Passwort zu wählen. Grundsätzlich gillt, auch das ist nur im Idealfall so, je länger ein Passwort ist, desto besser. Dies bedeutet wie bereits oben angeschnitten allerdings nicht, dass das Passwort 'aaaaaaaaaaaaaaaaaaaa' ein mathematisch sicheres Passwort ist. Die Länge des Passwortes ist nur einer von vielen Faktoren, die am Ende zur Komplexität und der daraus resultierenden Sicherheit beitragen. Im Idealfall besteht das lange Passwort aus mehreren Zeichenarten. Eine weitere Empfehlung ist es, keine Sonderzeichen an den Anfang oder das Ende des Passwortes anzuhängen, um es für einen Angreifer schwerer erratbar zu machen. Dies lässt sich damit begründen, dass sobald ein Angreifer die restlichen Zeichen des Passwortes erraten konnte oder durch Metadaten anderer Dienste (wie oben in dem Michael-Jackson Beispiel) kennt, das Durchprobieren von allen verfügbaren Sonderzeichen für die erste und letzte Stelle der Zeichenkette keine große Leistung erfordert. Sie machen das Passwort mathematisch zwar sicherer (Mehr Zeichenarten bedeuten mehr Zeichen insgesamt und dadurch mehr Kombinationsmöglichkeiten für Passwörter), bei gegebenen Umständen sind diese einzelnen Sonderzeichen allerdings obsolet und können weggelassen werden. Passwort Policys können allerdings auch teilweise wertvolle Informationen für einen potenziellen Angreifer bieten. Denn was Angreifer durch sehr strikte Passwort Policys unteranderem erkennen können, ist die Mindest- und Maximalzeichenlänge. Dabei wird der Angreifer zum Beispiel aus der Regel 'Das Passwort muss mindestens 8 und maximal 16 Zeichen lang sein.' alle Kombinationen für weniger als 8 Zeichen und mehr als 16 Zeichen bei der Erratung eliminieren können. Weitere Regeln wie 'Das Passwort muss mindestens ein Sonderzeichen beinhalten' können zusätzliche Informationen bieten. Daher sind Passwort-Policys zwar ein sehr wichtiges Werkzeug, um Nutzer zu sicheren Passwörtern zu zwingen. Durch die beeinträchtigte Bequemlichkeit des Nutzers können allerdings immernoch einfach zu erratende Passwörter entstehen. Verhindern lässt sich dies nicht ganz. Die Informationen die Nutzer beim Anmelden bekommen, nutzen Angreifer dann zum Knacken jener Passwörter. Die Faustregel lautet: Je mehr Metadaten, desto besser. (Aus Sicht des Angreifers)

2.3.2 Bequemlichkeitsproblem

Bei einem Angriff ist es nicht zwingend notwendig, dass die Authentifizierungsquelle, also jene Quelle bei der die Daten persistiert sind und die die Authentifizierung bei Eingabe durchführt, diese Daten durch fehlerhafte Programmierung herausgibt. Durch sogenannte Metadaten, das sind Informationen und Merkmale zu Daten, ist es Angreifern häufig möglich das Passwort zu erraten bzw. zurückzusetzen. Wenn also der Benutzername lautet 'MichaelJacksonFanForever' ist die Antwort auf die Sicherheitsfrage zum Lieblingskünstler nicht weit entfernt.

Gleichzeitig neigen Menschen aufgrund von Bequemlichkeit zu leicht merkbaren Passwörtern, die sie mehrfach für verschiedene Dienste verwenden. Dies stützt eine kürzlich

durchgeführte repräsentative Studie der Bitkom Reserach [1] im Auftrag des Digitalverbands Bitkom. So nutze etwa jeder dritte Onlinenutzer (36%) dasselbe Passwort für mehrere Dienste. Auch wenn gleichzeitig 63% der Befragten angaben, bei der Erstellung von Passwörtern auf “einen Mix aus Buchstaben, Zahlen und Sonderzeichen” zu achten, beweist diese Befragung an 1.000 Internetznutzern, dass die Frage nach der Sicherheit von Passwörtern auch im Jahre 2020 immernoch Relevanz hat. Eine ähnliche Studie hat die Bitkom zum Thema ’Nachlässigkeit bei Passwörtern’ am 08.11.2016 [2] gemacht, bei der die prozentuale Verteilung an unsicheren Passwortnutzern die befragt wurden nur einen Prozent höher liegt. Das heißt konkret, dass sich innerhalb von 4 Jahren keine messbare Besserung ergeben hat. Das Bewusstsein über die Internetpräsenz und der Schutz dessen scheinen immernoch keine große Aufmerksamkeit vom modernen Nutzer zu erhalten. Das Problem mit unsicheren Passwörtern ist allerdings so alt wie das Internet.

2.3.3 Übertragungsproblem

Das Problem mit der Unsicherheit von Passphrasen oder auch Passwörtern beginnt allerdings jedes Mal aufs Neue, sobald man ein Passwort eintippt. So ist die Bedrohung nicht mit der Wahl eines mathematisch sicheren Passwortes gebannt. Das wissenbasierte Verfahren (des Passwortes), welches sich einer Zeichenkette bedient die man in ein Feld eintippt, ist per sé dann unsicher sobald einer der Geheimnisträger (Menschen mit Kenntniss über die Parole) kompromittiert bzw. infiziert ist. So gibt es verschiedenste Angriffsvektoren um das Passwort eines Users für einen speziellen Dienst herrauszufinden. Von personalisierten (oder auch allgemeinen) Phishing Mails, zu Shoulder Surfing bis hin zu Trojanern und Keyloggern auf dem System Desjenigen. Diese können vom Angreifer teilweise remote ausgeführt werden, für manche Angriffe benötigt der Angreifer allerdings physischen Zugriff aufs System. Ein weiteres großes Problem ist die Übertragung von Passwörtern über die klassiche User-Browser-Schnittstelle. Dabei wird das Passwort im Browser des Clients gehasht und dann an den Server übertragen. Die sichere Kommunikation anhand des Hypertext Transfer Protocol Secure (HTTPS) findet erst bei der Übertragung zum Server statt, die Eingabe des Passwortes an den Browser allerdings ist ungeschützt. Diese Übertragung von Buchstaben kann mitgelesen werden. Das Hashen löst das Problem das ein Angreifer im selben Netzwerk mitlauschen und das Passwort über einen sogenannnten Man-in-the-middle Angriff entwendet bzw. nutzt. Somit sich der Identität des Passwortbesitzers bedient und sich bei anderen Diensten als Diesen ausgibt. Es bleibt allerdings immernoch das Problem des Wiederholungsangriffs. Der Angreifer muss womöglich also nicht ein Mal das Passwort im Klartext lesen können. Es genügt, den Hash und den Benutzernamen im Request abzufangen um diese dann in einem seperaten Aufruf vom eigenen Rechner an die selbe Uniform Resource Locator (URL) zu senden. Diese Art des Angriffs nennt man einen Replay Attack. Es handelt sich

um das Imittieren von Benutzereingaben durch einen Angreifer, bei der der Angreifer die Passphrasen nicht im Klartext kennt.

Durch den Zugang zum Dienst ist es ihm somit (je nach Implementierung des Dienstes) möglich, sensible Daten des Nutzers einzusehen die nicht für den Angreifer bestimmt sind. Die Frage nach der 'Relevanz' von sensiblen Daten sollte obsolet werden, wenn man an die Möglichkeiten denkt, die der Angreifer mit ihnen nun in der Hand hällt. Mit diesen könnte er den Nutzer zum Beispiel erpressen um an noch mehr Daten oder Geld des Nutzers zu kommen.

2.4 Zielsetzung

Ziel dieser Abschlussarbeit ist es einen Nachweis dafür zu liefern, dass die sichere und bequeme Authentifikation in 2020 keine utopischen Szenarien beschreibt und sich diese Punkte nicht gegenseitig ausschließen. Ziel ist es auch, den Leser in die Sicht des Angreifers auf Systeme einzuweisen, sodass im Idealfall automatische Schutzreaktionen wie das Wählen von sicheren Passwörtern hervorgerufen, wenn nicht sogar eine der beschriebenen FIDO2 Verfahren wie der erste oder sogar der Zweite Faktor, verwendet werden. Der Prototyp soll die verschiedenen Authentifizierungsmöglichkeiten veranschaulischen und präsentieren, um dem Nutzer die Wahl auf eines der Verfahren zu erleichtern. Gleichzeitig ist natürlich ein hauptsächliches Ziel dieser Arbeit auch die Grenzen von 'modernen' Authentifizierungsverfahren aufzuzeigen und zu zeigen, inwiefern eine Kombination dieser vorhandenen Verfahren, Probleme löst. Ein weiteres Ziel soll es sein, in gewisser weise kategorisch darzulegen, welches Verfahren für welchen Nutzertypen geeignet ist. Die Idee dahinter ist, dass es eine klare Trennung in der Nutzung von Accounts zwischen Entwicklern, Unternehmern und dem 'Casual Websurfer' gibt. Was alle drei Arten von Computernutzern allerdings gemeinsam haben ist: Sie besitzen persönliche Daten, an die kein Angreifer bzw. eben kein Dritter gelangen soll. Der Schutz dieser Daten sollte jedem Individuum selbst wichtig sein, um in den nächsten 5 bis 10 Jahren auf Besserung zu hoffen. Nicht nur technisch muss die Menscheit mit dem neuen digitalen Zeitalter umgehen und sich absichern können, sondern auch auf die menschliche Komponente achtgeben. Es sollte dennoch erwähnt sein, dass diese Arbeit nicht darauf abzielt jede einzelne Authentifikationsstrategie und Möglichkeit aufzuzeigen und zu bewerten sondern eher die klassichen und weiterverbreitetsten Verfahren aufzuzeigen, auf die die restlichen Verfahren meist basieren. So ist der Yubikey am Ende nur eine Möglichkeit zur Umsetzung von einer Challenge-Response-basierten Authentikation anhand von privaten Schlüsseln welches bereits im FIDO Standart definiert ist. Auch ist es ein Nicht-Ziel dieser Arbeit das Resultat auf eine einzige perfekte Lösung zu dezimieren und diese zum neuen Standart zu erklären.

3 Grundlagen

Laut des IT-Grundschutz-Kompendiums vom Bundesamt für Sicherheit in der Informationstechnik könne ein Benutzer aus Bequemlichkeit oder pragmatischen Gründen bewusst auf komplizierte und unhandliche Kryptomodule verzichten und Informationen stattdessen im Klartext übertragen. [**A1**] Dies stellt ein hohes Sicherheitsrisiko für Unternehmen, aber auch Privatpersonen dar, da Benutzer nicht gewillt sind ihre Passwörter durch komplizierte Verfahren zu erzeugen und in regelmäßigen Abständen vollkommen randomisiert zu setzen. An neuartige Ansätze des Logins in nativen oder webbasierten Anwendungen stellen sich dadurch völlig neue Herausforderungen. So müssen neue Authentifizierungsmöglichkeiten nicht nur sicher sein, sondern auch komfortabel genutzt und bedient werden können, da sie sonst von den Endnutzern gemiedern oder umgangen werden. Wichtig ist eben auch, dass die breite Masse Zugriff auf die Ressourcen hat, die es zur Nutzung dieser Verfahren braucht. Man denke nur an die ganzen betrieblichen Passwörter, bei denen zum Monatsende nur eine Zahl an der letzten Stelle des Passwortes geändert wird. Laut einer Statistik von 2019, der "Global Data Risk Report From the Varonis Data Lab", gaben 38% aller Nutzer an ein Passwort im Unternehmen zu nutzen, dass sie nicht (oder nur geringfügig) ändern. Außerdem wird laut dieser Statistik alle 364 Tage ein Data-Breach aufgrund von unsicheren Passwörtern in einem mittelständigen Unternehmen stattfinden. Die monatliche Ablaufzeit von Passwörtern in Unternehmen scheint also nicht ganz den Effekt zu erzielen, der ursprünglich damit geplant war, da die Arbeiter die Bequemlichkeit über die Sicherheit stellen.

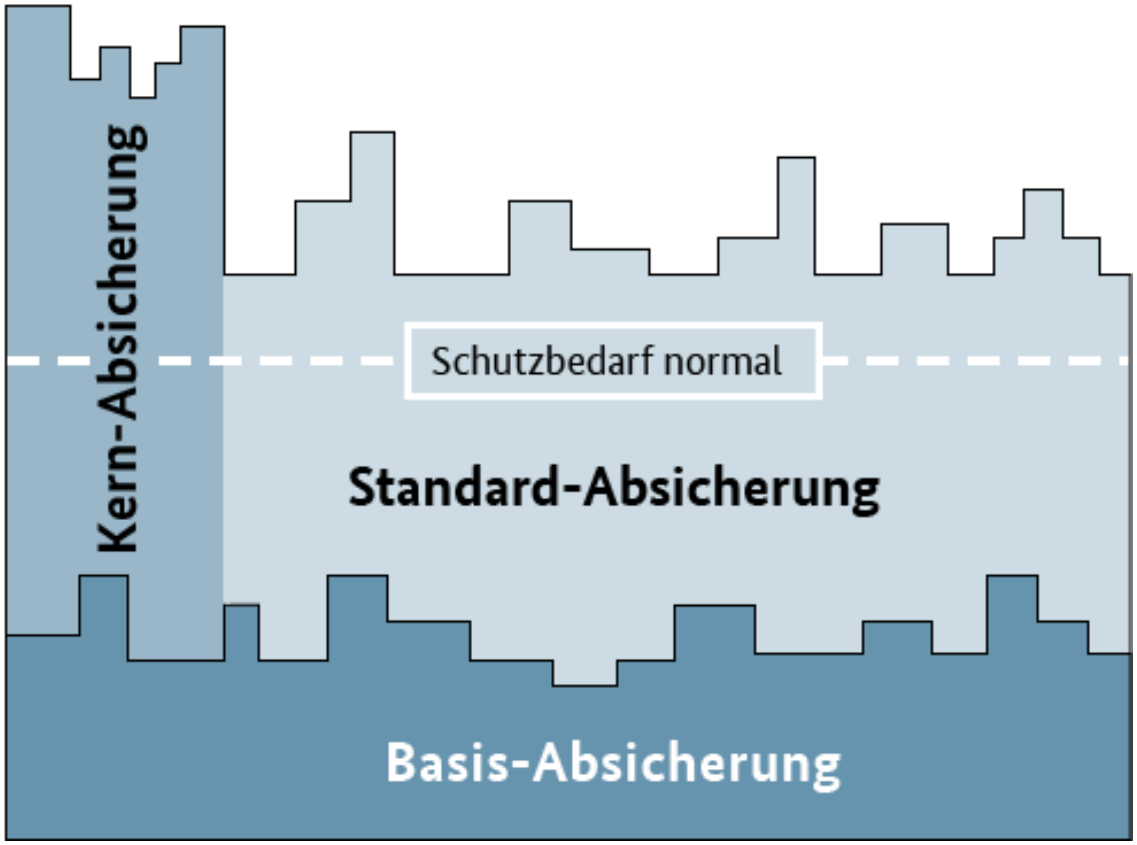
3.1 IT-Grundschutzkriterien

Der IT-Grundschutz definiert den Schutzbedarf eines bestimmten Assets je nachdem welches Risiko bei Verletzung der Grundwerte Vertraulichkeit, Integrität und Verfügbarkeit entstehen [10]. Allgemein existieren folgende Schutzbedarfskategorien:

- **normal** (Schadenauswirkungen begrenzt bis überschaubar)
- **hoch** (Schadenauswirkungen könnten hoch bzw. beträchtlich sein)
- **sehr hoch** (Schadensauswirkungen können ein existenziell bedrohliches Ausmaß annehmen

Bei dem Authentifikationsprototypen wird die Schadensauswirkung für alle nicht personenbezogenen Daten wohl im Bereich normal liegen, da schon im Aufbau darauf geachtet wird, dass nur so viele Daten vom User verwendet werden, wie zwingend notwendig. (Nach dem Need-To-Know Prinzip) Sollte die Webseite oder Teile der Webseite publiziert bzw. im Business - Umfeld genutzt werden, muss eine Neubewertung der Daten nach Vertraulichkeit, Integrität und Verfügbarkeit stattfinden. Vor allem muss darauf geachtet sein, dass Daten über Fingerabdrücke verschlüsselt und Passwörter im gehashten Zustand in Datenbanken persistiert werden. Bei Kompromittierung des Hauptrechners, welches die größte Bedrohung in diesem Szenario darstellen würde, droht ein Data Breach mit dem Angreifer diese Daten weiterverwenden können. Durch Verschlüsselungen durch Schlüssel, die nicht auf dem Hauptsystem (oben u.a als Hauptrechner benannt) liegen. Gleichzeitig sollten die genutzten Verfahren insgesamt mathematisch sicher sein, auf veraltete Verschlüsselungsverfahren ist zu verzichten. Dies gillt auch für Hashes wie MD5, die mittlerweile relativ akkurat durch sehr große vorgerechnete Tabellen erraten werden können.

Im Falle einer Kompromittierung besäße der Schutzbedarf der Daten innerhalb der Datenbank, welche nicht gehasht oder anderweitig verschlüsselt sind, die Kategorie 'sehr hoch'. Eine Kompromittierung kann für das Unternehmen einen Imageschaden sowie weitreichende juristische Klagen zur Folge haben. Damit sind bereits 2 der 7 aufgeführten Schadensszenarien durch den BSI beschrieben. Je nach dem wie kompliziert die Ursprungsdaten sind und welcher Hashingalgorithmus in Kombination mit Salt und Pepper genutzt wurde, können vorallem Nutzerpassphrasen an die Öffentlichkeit gelangen und Angreifer können die Passwörter für andere Dienste nutzen. Die Wahrscheinlichkeit für diesen Schaden ist noch vergleichbar niedrig, weshalb die Schadensauswirkung hoch statt sehr hoch ist.

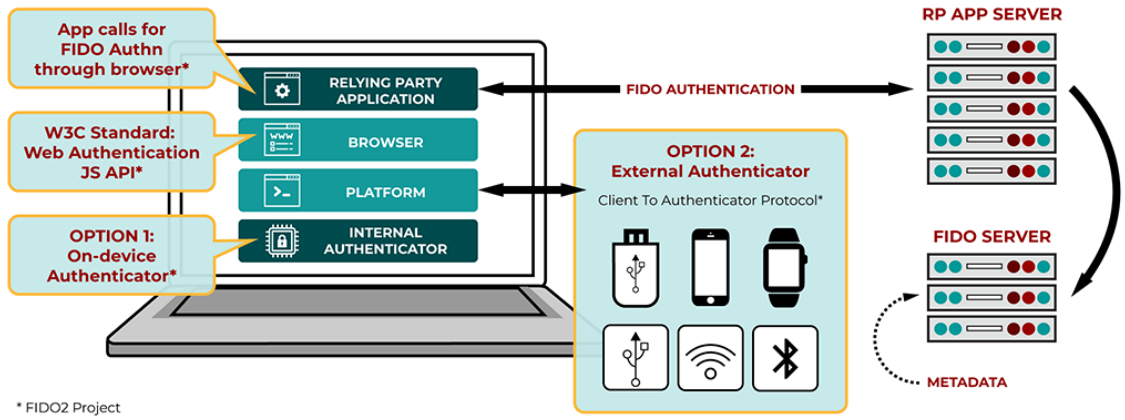


Der IT-Grundschutz definiert drei Arten der Absicherung. Die Basis-Absicherung ist relevant für Institutionen die einen Einstieg in den IT-Grundschutz suchen und relativ schnell alle relevanten Geschäftspozesse mit einfach umzusetzenden Basismaßnahmen sichern wollen. Die Kern-Absicherung konzentriert sich auf besonders wichtige Geschäftsprozesse und vertieft sich in die Sicherung dieser. Von einer Standard-Absicherung spricht man, wenn alle empfohlenen IT-Grundschutz-Vorgehensweisen durchgeführt werden. Sie beschreibt den allumfassenden Schutz der Prozesse und Bereiche der Institution, wie das Schaubild vom BSI verdeutlicht. [9] Bei dem Prototyp wird eine Basis-Absicherung

nach BSI durchgeführt und darauf geachtet alle Datenschutzkriterien zu erfüllen. Dabei wird vor allem die Checkliste des IT-Grundschutzes zu Webservern und Webanwendungen betrachtet. Kriterien die nicht erfüllt werden konnten oder wurden, werden dokumentiert und im Fazit erläutert. Die Wahl der Basis-Absicherung begründet sich damit, das lokale Testdaten im Prototyp verarbeitet werden, für die kein bis nur ein sehr geringer Schutzbedarf besteht. Eine Kern-Absicherung käme nur in Frage, falls ein ganz bestimmter Prozess oder Asset des Prototyps geschützt werden müsste wie zum Beispiel der Zugriff auf die Datenbank durch den Prototypen, welche nicht der Fall ist. Insgesamt ist dieser Teil der Abschlussarbeit auch als Einstieg in den IT-Grundschutz zu verstehen, welcher laut BSI selbst die Basis-Absicherung als Empfehlung und zur Folge meiner Entscheidung hat und sich am Besten für vorhandene Zwecke eignet.

3.2 FIDO2

Diese Probleme sind den Menschen seit einigen Jahren bekannt und wurden vor allem mit Verfahren gelöst, die keine Passwörter (oder allgemein weissensbasierte Token) zur Authentifikation benötigen und diesen maximal als ersten Layer der Sicherheit anerkennen. Das Bekannteste Beispiel für einen Standart zur sicheren und bequemen Authentifikation im Internet findet man unter dem Schlüsselwort FIDO2. FIDO steht dabei für 'Fast Identity Online' (Schnelle Identität im Netz). Sie ist das Ergebnis einer Kooperation des World Wide Web Consortium (W3C) und der FIDO Alliance. FIDO2 basiert auf vorhandenen Protokollen wie Web Authentication (WebAuthn) für die Browser-Server-Kommunikation und CTAP für die Browser-Authenticator-Kommunikation. Auf der offiziellen Webseite der yubico, einem der Hauptentwickler und Publizierer des Vorgängerprotokolls Universal second factor (U2F) wird die FIDO2 - U2F, also die zwei Faktor Authentifizierung spezifiziert durch U2F das immernoch im FIDO2 Protokoll beheimatet ist, wie folgt beschrieben: "an open authentication standard that enables internet users to securely access any number of online services with one single security key [...]. FIDO2 is the latest generation of the U2F protocol" [5]. Während das Vorgänger - Protokoll U2F von Google und Yubico ins Leben gerufen wurde, ist FIDO2 ein offener dezentraler Kommunikationsstandart für die passwortlose Kommunikation welches die Authentifizierung für sowohl Privatnutzer als auch Unternehmen beqeuem und gleichzeitig sicher machen soll.



Um die Sicherheit des Nutzers zu gewährleisten kombiniert FIDO2 die Methoden des UAF und des U2F. Bevor wie bei gängigen Zweifaktoren wie eines PINs oder sechsstelligen Schlüssels der Schlüsselaustausch stattfinden kann, muss der User der Anwendung oder des Dienstes eine lokale Verifikation durchführen. Diese soll sicherstellen, dass es sich bei der Person, die die Authentifikation durchführt und der Person, die den Schlüssel vorher registriert hat, um die selbe Person handelt. Diese Verifikation kann zum Beispiel ein Knopf auf einem USB - Stick sein, auf den der Web-Service wartet oder ein Fingerabdruck-Sensor - Bevor er die Challenge an den Nutzer (bzw. über dessen Browser dann an die Webseite) sendet. So lange also ein Angreifer keinen physischen Zugang zu diesem Gerät erhält, ist das Verfahren sicher. Sollte es doch vorkommen, dass der Angreifer sich Zugriff verschafft und den Knopf drückt, setzt die sozusagen zweite Phase der Authentifikation ein. Die Webseite sendet dem User eine Challenge, welche der User lokal mit seinem Schlüssel auf dem Computer lösen kann. Die Webseite erhält dann das Ergebnis und vergleicht dieses mit dem eigenen Ergebnis. Gibt es ein Match, sendet der Server der Webseite die zugehörige Response zur Challenge an den User zurück und lässt ihn passieren. Wie die Abbildung zeigt gibt es neben der externen Authentifikation durch Smart-Watch, USB- Stick oder Smartphone auch die Option der 'on-device-authentication'. Damit ist die Authentifizierung durch einen PIN oder einen eingebauten Fingerabdruck-Sensor (über biometrische Daten aller Art) gemeint, die allerdings nicht extern angeschlossen ist sondern sich auf dem Gerät selbst befindet. Auf die initiale Schlüsselerstellung und weitere Details zum FIDO Protokoll, die für diese Arbeit relevant sind, wird im nächsten Kapitel: 'Grundlagen' eingeggangen.

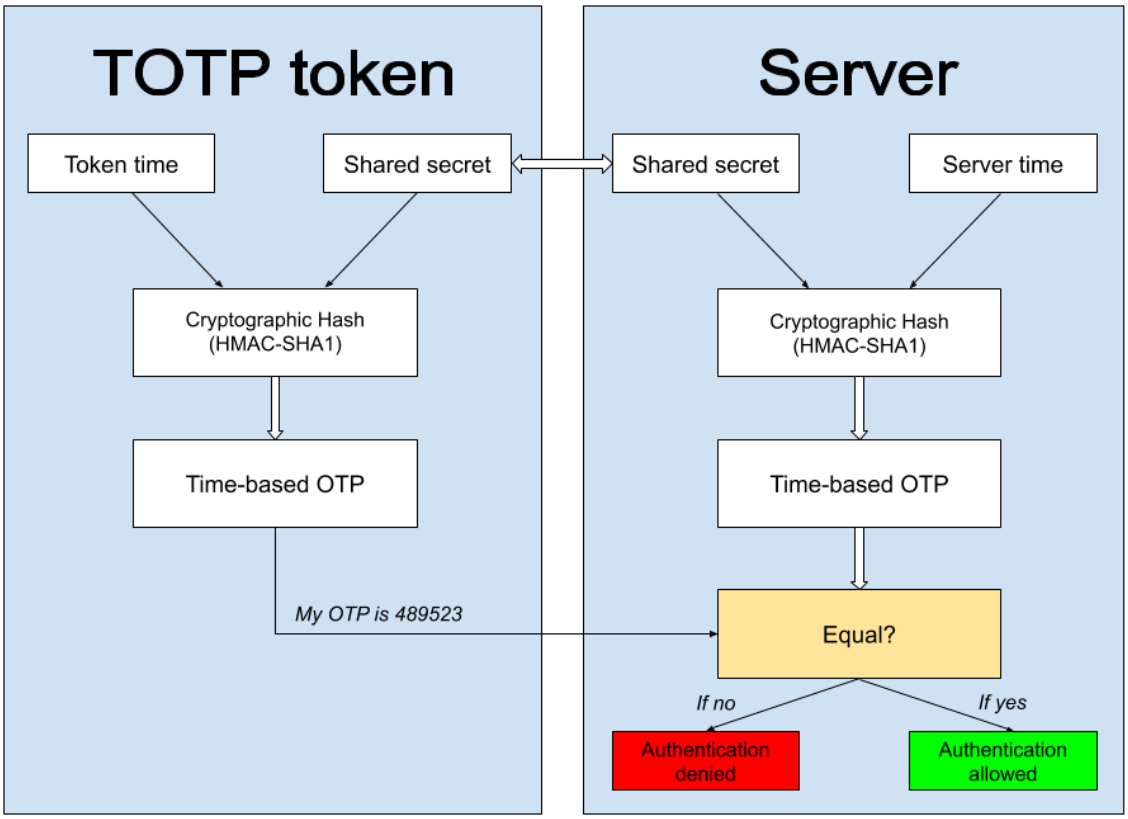
3.3 Authentifizierungsmethoden

3.3.1 Username & Passwort

Trotz der bereits erwähnten Unsicherheiten und Probleme des beliebten Schlüssels aus der Kategorie Wissen, dem Passwort, ist das Passwort laut eines Artikels von Thomas Maus 2008 “nicht aus unserer Arbeitswelt [...] wegzudenken” [12] Der Artikel spricht bereits 2008 über alternative Authentifikationsmethoden wie der Einmalkennwörter und Weiterem. Dies ist nur ein weiterer Beweis dafür, dass schon vor mehr als 10 Jahren die Passwortproblematik erkannt wurde. In dem Artikel geht Herr Maus der Hypothese nach, ob nun Passwörter wirklich per sé unsicherer sind als die Authentifikationsmethoden der beiden anderen Kategorien Besitz und biologische Merkmale. So sei das Kernproblem des Passwortes, dass das Passwort direkt nach der Eingabe ein geteiltes Geheimnis ist, da alle beteiligten Systeme es nun mitschneiden konnten und nun automatisch Geheimnisträger sind. Das Passwort bietet sehr viele Angriffsvektoren, so “Shoulder Sourfing, Phishing, Social-Engineering, Man-in-the-Middle Angriffe, schlechte Passwort Qualitäten, das Teilen des Passwortes mit Familie und Freunden” [12] und vielem mehr. Bei dem Prototyp wird der Ist-Zustand einer Username und Passwort - Authentifikation demonstriert, wie man sie heutzutage auf höchstwahrscheinlich jedem Webdienst in der Form als ersten schwachen Faktor vorfinden wird. Während man über alle Schwächen des Passwortes spricht, muss man dennoch anerkennen, dass das Passwort eine gewisse Flexibilität bietet. Es genügt das Wissen über eine bestimmte Zeichenfolge um sich zu authentifizieren, dieses Wissen muss nicht zwangsmäßig auf ein Blatt geschrieben oder in einem Passwort-Manager beherrbergt werden. Das beste Passwort ist jenes, welches nur in dem Gehirn des Nutzers persistiert ist und mit niemandem geteilt wird. Anders als bei neueren Verfahren, die als sicherer gelten, benötigt es keine Smart-Card, USB-Sticks oder anderweitige technische Hilfsmittel um sich zu authentifizieren.

3.3.2 Einmalkennwörter

Ein Kennwort ist eine ëine Zeichenfolge, die zur Authentifizierung verwendet wird. Damit soll die Identität einer Person [...] auf eine Ressource nachgewiesen werden."[**A4**]. Ein Einmalkennwort im Vergleich ist ein Kennwort, das nur ein einziges Mal für eine Authentifizierung genutzt werden kann. Man unterscheidet drei Arten von One time password (OTP)’s.



Bei der timerbasierten (Time based one time password (TOTP)) Methode wird die aktuelle Systemzeit (Token time) mit dem zu verschlüsselnden Text (Shared secret) anhand eines kryptografischen Verfahrens verschlüsselt. Es entsteht ein kryptografischer Hashwert (Cryptographic Hash), welcher meist den HMAC-SHA1 Hashingalgorithmus verwendet. Dieses Verfahren findet gleichermaßen auf dem Server statt. Der einzige Unterschied zum Server besteht darin, dass die Systemzeit des Servers genutzt wird. Findet innerhalb der festgelegten Zeit (ein Token entfällt laut RFC6238 standartmäßig nach 30 Sekunden, diese Zeit ist modifizierbar) eine erfolgreiche Authentifikation statt, wird der Zugang gewährt. [13]

Ein entscheidender Nachteil dieser Methode entsteht, falls der authentifizierende User innerhalb dieser 30 Sekunden (beispielsweise) durch eine Störung des Netzes oder einen Netzausfall die Verbindung verliert. Der aktuelle TOTP Code wird ungülltig und muss erneut angefragt bzw. erstellt werden. Apps wie der Google Authenticator lösen diesen Problem, in dem sie einen Timer setzen und alle 30 Sekunden einen automatisch generierten neuen Code mit der aktuellen Zeit generieren. Dabei muss natürlich drauf geachtet werden, dass Server und Client synchron sind, dieses Verfahren ist deshalb an eine bestimmte Zeitspanne und ein externes Gerät (sei es Smartphone, TOTP Smart Card oder ein externer Rechner) gebunden.

Laut Margaret Rouse biete das TOTP Verfahren zusätzliche Sicherheit für den Nutzer, da selbst bei Erhalt des Passwortes das TOTP nicht in die Hände des Angreifers gelangt und nach einer gewissen Toleranzzeit verfällt. ([**A5**] Eine Beispielapplikation für die Nutzung eines TOTP - Ansatzes für zusätzliche Sicherheit ist der "Google Authenticator", welches in jedem gängigen App Store zu finden ist.

Ereignisbasierte OTPs besitzen einen Ereigniszähler, der bei jeder versuchten Authenzifizierung einen Zähler auf Server und Clientseite synchronisiert inkrementiert. Sollte der Zähler asychnron werden bzw. der Server einen anderen Wert gespeichert haben als der Client bei der nächsten Authentifizierung sendet, wird der Authentifizierungsvorgang abgebrochen. Man findet diese Funktionalität wortwörtlich beschrieben in Googles Time and event based one time password Patent [**A6**] in folgendem Wortlaut "[...] the characteristics of an event can be the value of a counter that is incremented each time the user pushes a button on the token"[**A6**]. Für diesen Prozess wird der HOTP Algorithmus genutzt, der im RFC4226 näher beschrieben ist. Diese Art der Authentifizierung kann zum Beispiel für die E-Mail Verifikation und damit die Identifikation (wie zuvor erläutert) genutzt werden.

Challenge-response basierte OTP Verfahren bedienen sich an komplizierten mathematischen Verfahren. Das heißt, es erfolgt ein ACK (Acknowledge bzw. Initialanstoß zur Authentifizierung). Der Client, berechnet die Response mithilfe der mathematischen Formel und sendet das Ergebnis an den Server. Sollte es einen Match geben, erhält der Client eine Response vom Server, der seine Echtheit bestätigt. Synchronisationsprobleme kann es bei diesem Verfahren entgegen der ereignis oder timerbasierten OTP-Verfahren nicht geben, da die Berechnung dieses ’Schlüssels’ vollkommen auf der Clientseite funktioniert. Der Server überprüft diese Rechnung nur mit seinem eigenen Wert, stellt aber keine weiteren Rechnungen oder Umformungen mit diesem Wert an. Der Hauptvorteil dieses Verfahrens ist, dass unabhängig von der Zeit und einem speziellen Ereignis eine Anfrage gestellt werden kann. Der Server kann also seine ’Challenge’ abschicken und muss keine ’Response’ innerhalb einer festgegebenen Zeit erhalten, um authentifizieren zu können. Dieses Verfahren gilt als besonders sicher, da es auf Serverseite keinen Algorithmus gibt, der sich vorausberechnen lässt.

3.3.3 Web Authentication

Webauthn ist kurz für die ’Web Authentication’. Zur Verfügung gestellt wurde dieser Standard der Authentifikation 2018 von der FIDO Alliance und dem W3C. [**A7**] Sie ermöglicht eine passwortlose (bzw. benutzerdatenlose, es wird also auch keine User-ID benötigt) Authentifikation durch Tokens (Sicherheitsschlüssel). Für dieses Verfahren wird zunächst ein Buffer aus kryptografischen random Bytes generiert, dass der Verhinderung von ’Bruteforce’ (vom webauthn - Guide auch als ’reply attacks’ beschrieben) -

Angriffen dienen soll. Web Authentication nutzt das vorhandene public-key-Verfahren für Webseiten. Der Standard definiert allerdings nicht welche Art von Schlüssel genutzt wird. Unter den Möglichkeiten zählen der ÜSB security key"[**A7**] oder der "built-in fingerprint sensor"[**A7**]. Webauthn basiert auf vielen bereits vorhandenen Abhängigkeiten der Informatik wie die Standards von HTML5, ECMAScript, COSE (CBOR Object Signing and Encryption COSE, RFC8152) oder dem Nutzen von der Base64url encoding.
Zusammengefasst hat der FIDO2 Standard mit CTAP (dem Protokoll für externe Authentifikationen mit Mobilgeräten) und Webauthn (der Schnittstelle bzw. ÄPI") vorhandene Funktionen definiert, mit der native Authentifizierungsmethoden wie das public-private Key Verfahren auf die Webseite übertragen werden können. Wichtige Beispiele für Web Authentication sind der Yubikey, der USB-Token oder unsere biometrischen Daten (FaceID oder TouchID), die wir täglich in jedem AppStore nutzen, der diese Daten verschlüsselt an eine Webseite bzw. einen öffentlichen Store übermittelt. In unserem Use-Case wollen wir die Authentifizierung anhand von Webauthn als Multi-Faktor nutzen, also als zusätzliche Sicherung neben einem Passwort. Denn wie oben beschrieben, kann das Wissen eines Menschen durch Data Breaches oder menschliches Versagen (Gutglauben) leicht abhanden kommen. Mit einem Yubikey oder einem USB-Token befindet man sich allerdings auf der Ebene des Besitzes, wodurch ein potenzieller Angreifer es schwerer hat an die Daten zu kommen. Biometrische Daten und diese public-private-Key Verfahren gelten nämlich allgemein als sehr sicher. Wodurch es technisch sehr schwer bis mathematisch (in gegebener Menschenzeit) fast unmöglich ist, die Algorithmen hinter ihnen zu knacken.

4 Konzeption

4.1 Prototypenaufbau

Das Ziel des Prototypes ist es, wie eingangs erwähnt, vorhandene Authentifizierungsverfahren abseits der klassischen UserID / Passwort Methode zu begutachten und dessen Schwächen aufzudecken. Der Prototyp beschreibt eine einfache Webseite, die aus dem globalen Internet erreichbar sein wird und zwei Eingabefelder und einen Loginbutton besitzt. Die unterschiedlichen Methoden der Authentifizierung wählt man über ein Dropdownmenü über dem Login - Button. Je nach Authentifizierungsverfahren werden kleine Popup-Boxen sichtbar, die die weiteren Schritte für die Authentifikation erläutern. So muss bei der zwei Faktor Authentifizierung anhand von einem Fingerabdruck weder ein Username noch ein Passwort eingegeben werden. Die Webseite muss lediglich auf die Schnittstellen des Betriebssystems zugreifen, um den Nutzer zu authentisieren. Wie der Nutzer eingangs den Fingerabdruck eingerichtet hat, spielt für die Webseite keine Rolle. Ein Teil des Prototypen soll die gewählten Methoden demonstrieren. Neben einer erfolgreichen Demonstration der Authentifizierung sollen nach jeder Methode auch Kennwerte ausgegeben werden. Einer davon soll zum Beispiel die Zeit von der ersten Eingabe in ein Eingabefeld bis zur Authentifizierung in Sekunden zählen und anzeigen. Weitere Messwerte sind denkbar und werden sich bei der Implementation ergeben.

Neben den vorhandenen Methoden soll die eigene Architektur aufgebaut werden, die sich an vorhandenen Authentifikationsmöglichkeiten bedient. Die UserID und das Passwortfeld sind beim ersten Aufruf der Seite zwar zu sehen, müssen allerdings nicht zwingend für jedes der Verfahren genutzt werden, so kann es zum Beispiel bei einer besitzbasierten Authentifikation bereits reichen, den Besitz (z.B einen USB - Stick, welcher einen privaten Schlüssel beherbergt) im Computer einstecken zu haben und auf den Loginbutton zu drücken. Bei der Architektur muss zwingend eine Datenbank und ein Backend zur Webseite implementiert werden um einerseits die eigeggangenen Daten zu bearbeiten und andererseits in die Datenbank zu persistieren. Das Dropdownmenü zeigt die Authentifikation über biologischen Merkmale (Touch ID oder Face ID) nur sofern das Gerät, auf welchem der Prototyp bedient wir, einen entsprechenden Sensor und die entsprechende Software zur Verarbeitung besitzt.

4.2 Auswahl der Authentifizierungsverfahren

Neben des altbekannten TOTP Verfahrens, wird das Secure Element und die E-Mail Authentifikation betrachtet. Dabei bedienen sich diese Verfahren aller drei Möglichkeiten der Authentifikation, dem Wissen, dem Besitz und der körperlichen Merkmale.

4.3 Kriterien zur Bewertung des Prototypen

4.4 Architektur

5 Prototypischer Lösungsansatz

5.1 Bequemlichkeitsproblem

5.2 Kriterien für erfolgreiche Authentifikation

5.3 Implementierung des Prototypen

6 Auswertung

6.1 Implementierung

6.2 Ablauf und Durchführung

6.3 Fehlerbetrachtung

7 Ausblick und Fazit

A Anhang

A.1 Ergänzende Informationen

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.