ARBEITSKREIS
USABLE SECURITY & PRIVACY

Nutzerzentrierter Schutz sensibler Daten

GERMAN **UPA**
Berufsverband der Deutschen Usability und User Experience Professionals

German UPA

Die German UPA ist der Berufsverband der deutschen Usability Professionals. Der Verband ist ein Netzwerk von und für Usability-Experten, die sich der Wissensvermittlung und Meinungsbildung rund um das Thema Usability und User Experience verpflichtet fühlen.

Innerhalb der German UPA engagieren sich Mitglieder in thematisch unterschiedlichen Arbeitskreisen, in denen sie sich fachlich austauschen und überregional zusammenarbeiten.

Ein Arbeitskreis widmet sich dem Thema Usable Security & Privacy. Die Mitglieder des Arbeitskreises sind Experten für benutzerfreundliche Informationssicherheit und vertreten innerhalb der UPA die Schnittstelle zwischen Usability, IT-Sicherheit und Datenschutz. Neben der Öffentlichkeitsarbeit findet ein reger Wissens- und Erfahrungsaustausch statt. Neue Mitglieder sind jederzeit willkommen. Nähere Informationen finden Sie auf der Webseite der German UPA unter Arbeitskreise.

https://ak-usable-security-privacy.germanupa.de

Inhaltsverzeichnis

Kapitel 1

Geht das überhaupt?

Vereinbarung zweier gar nicht so gegenläufiger Qualitätsmerkmale

Produkte, Systeme, Plattformen und Dienstleistungen – auch die des alltäglichen Gebrauchs – sind zunehmend digital und vernetzt. Aus dieser Tatsache erwachsen viele neue Gegebenheiten sowie Chancen und Risiken. Durch die Digitalisierung sowie die technische Kopplung bisher getrennter Domänen ergeben sich viele Potentiale durch ein höheres Maß an Automatisierung. Zudem lassen sich Anwendungen realisieren, die ein weitreichendes Partizipieren der Nutzer eröffnen. Hiermit geht aber auch zwangsläufig einher, dass Daten vermehrt in digitaler Form erhoben werden und sich über die vernetzten Strukturen verbreiten. Dem Schutz der Daten muss folglich ein hoher Stellenwert beigemessen werden, um den Teilnehmern in derart vernetzten Umgebungen die Kontrolle über ihre Daten weiterhin zu ermöglichen. Digitale Produkte, Systeme, Plattformen und Dienstleistungen sind daher mit adäquaten Sicherheitsmechanismen auszustatten.

Als weiteres Erfolgskriterium digitaler Erzeugnisse steht seit einigen Jahren deren Usability im Fokus. Aufgrund vieler Gegebenheiten hat sich die landläufige Sichtweise etabliert, dass Usability und Security bzw. Privacy gegensätzliche Qualitätsmerkmale seien, die sich nicht miteinander vereinbaren ließen. Mit diesem Spannungsfeld befassen sich die Disziplinen Usable Security und Usable Privacy. Sie erforschen, wie weit sich die Ziele der unterschiedlichen Qualitätsmerkmale ausbalancieren bzw. vereinbaren lassen, um den anstehenden und weiter zunehmenden Herausforderungen der uns ubiquitär umgebenden digitalen Objekte durch benutzbare sicherheits- und privatheitsfördernde Technologien zu begegnen. Bei der zunehmenden Verwendung digitaler Objekte und den darin verbauten Sicherheitsmechanismen werden Nutzer aktuell häufig zum schwächsten Glied der Sicherheitskette gemacht, wodurch die

Effektivität des Schutzes gemindert wird oder sogar ganz entfallen kann, was zum Verlust der Vertraulichkeit sensibler Daten oder zur Kompromittierung ganzer Firmeninfrastrukturen führen kann. Der Bedarf an Usable Security und Privacy ist daher schon heute als sehr hoch einzustufen und wird mit den oben aufgezeigten Entwicklungen zunehmend wichtiger.

Aufbau und Struktur dieser Fachschrift

Das Ziel des Arbeitskreises „Usable Security & Privacy“ ist es, diese Themen sowohl in der Forschung als auch in der Anwendung weiter voran zu treiben. Mit der vorliegenden Fachschrift soll in das Thema eingeführt werden. Kapitel 2 geht dazu zunächst auf verschiedene Nutzertypen ein, die im Rahmen von Usable Security & Privacy zu berücksichtigen sind.

Kapitel 3 befasst sich mit dem Systementwicklungsprozess und zeigt auf, wie dieser durch die interdisziplinären Engineering-Ansätze der betrachteten Domänen zu einem integrierten Vorgehen verzahnt werden kann. Hierbei wird die Relevanz geeigneter Werkzeuge herausgestellt, die in den verschiedenen Systementwicklungsphasen die Arbeiten unterstützen können. Konkrete und aktuell bereitstehende Werkzeuge werden in Kapitel 4 vorgestellt. Zur Veranschaulichung der Zusammenhänge der in den vorangegangenen Kapiteln besprochenen Inhalte spielen wir in Kapitel 5 zwei Fallbeispiele durch. In Kapitel 6 werden weiterführende Informationen und Ressourcen genannt, die zur Vertiefung und Vernetzung mit der Community anregen sollen.

Kapitel 2

Für wen mache ich das?

Nicht alle Nutzer sind gleich

Schutzmechanismen digitaler Produkte, Systeme, Plattformen und Dienstleistungen werden für die Nutzer als Bestandteile von Software, Apps und anderen interaktiven Produkten sichtbar. Sie können nur dann einen effektiven Schutz bieten, wenn die Nutzer verstehen, wie diese Mechanismen korrekt angewendet werden. Bei der Entwicklung dieser Mechanismen müssen daher neben dem Anwendungskontext auch die Eigenschaften und Fähigkeiten der Nutzer berücksichtigt werden. Für den Bereich Sicherheit und Datenschutz sind die vier grundlegenden Nutzertypen hilfreich, die der Verein „Deutschland sicher im Netz" (DsiN) in seinem Sicherheitsindex [34] unterscheidet. Sie helfen dabei, ein besseres Verständnis für die Zielgruppe und ihr Verhalten zu entwickeln.

Diagramm1: Die souveränen Nutzer haben ein deutlich überdurchschnittlich ausgeprägtes Sicherheitsverhalten und verfügen über gutes Wissen bezüglich Schutzmechanismen.

Die gutgläubigen Nutzer zeigen Defizite sowohl bei der Einschätzung von digitalen Risiken als auch bei der adäquaten Anwendung von Schutzmaßnahmen.

Fatalistische Nutzer unterlassen Sicherheitsmaßnahmen, obwohl sie diese kennen und sich bedroht fühlen.

Außenstehende Nutzer zeigen im Vergleich mit den anderen Nutzergruppen deutliche Defizite bei der Kenntnis und der Nutzung von Schutzmaßnahmen.

Bei den letzten drei Nutzertypen ergibt sich ein besonders großes Potential für benutzerfreundliche Sicherheitstechnologien. Mit insgesamt 56,1 % bilden diese Nutzertypen die „absolute Mehrheit“ der Nutzer. Nichtsdestotrotz können auch souveräne Nutzer von Usable Security und Privacy profitieren, wenn sie dadurch im Vergleich effizientere, effektivere und zufriedenstellendere Umsetzungen erhalten.

Personas helfen bei Designentscheidungen

Im Usability Engineering werden oft sogenannte Personas [35] verwendet, die eine gute Hilfestellung bieten, um sich besser in die Rolle eines Nutzers versetzen zu können. Personas sind fiktive Personen, die typische Anwender einer Zielgruppe repräsentieren – seien dies spezialisierte „Power-User“ oder eher unbedarfte Gelegenheitsanwender. Grundlage für die Erstellung solcher Personas können Beobachtungen von potenziellen Nutzern, Interviews oder Marktforschungsergebnisse sein.

In der Regel wird nicht nur eine Persona erstellt, sondern so viele wie nötig sind, um die Zielgruppe des Produkts gut abzudecken. Anhand der Personas kann sich das Designer- und Entwicklerteam die Bedürfnisse der Nutzer besser vergegenwärtigen und es kann unterschiedliche Benutzungsszenarien aus deren Blickwinkel durchspielen. Durch das Nachvollziehen der Denk- und Arbeitsweise der Endanwender fällt es einfacher, die richtigen Designentscheidungen zu treffen – bei der Produktentwicklung insgesamt, aber auch speziell wenn es darum geht, Sicherheitsfunktionen und Schutzmechanismen für eine bestimmte Zielgruppe möglichst benutzerfreundlich zu gestalten.

Ein Beispiel: In Abbildung 1 sind zwei Personas zu sehen. Diese werden für die Entwicklung einer Messenger-App zugrunde gelegt. Die App soll sowohl beruflich wie privat genutzt werden. Aus den Personas kann man ableiten, dass die Themen Security und Privacy bei Martina im Fokus stehen, bei Lars hingegen nicht. Da beide zur Zielgruppe des Produkts gehören, sollten Designentscheidungen dennoch so getroffen werden, dass sie beiden einen möglichst umfassenden Schutz gewährleisten.

Name	Martina Schäfer	Lars Dittrichs
Aussage	„Auch unterwegs brauche ich leichten Zugriff auf alle benötigten Informationen.“	„Schneller Austausch mit Freunden ist schon wichtig, um auf dem Laufenden zu bleiben.“
Beruf	Firmenkundenberaterin bei einem Finanzdienstleister	Ausbildung zum Gesundheits- und Krankenpfleger
Persönliche Daten	48 Jahre alt, verheiratet, 2 Kinder, Hochschulabschluss der Betriebswirtschaft	20 Jahre, ledig, Realschulabschluss
Charaktereigenschaften	zuverlässig und souverän	verantwortungsvoll, aber gutgläubig
Softwareerkenntnisse	interessiert sich für Software und technische Produkte, sowohl beruflich wie privat	geübter Anwender (Smartphone, Apps) kein tiefer gehendes technisches Verständnis
Usabilityziele	schnelle Übertragung der Nachrichten	einfache Bedienung in allen Situationen
Sicherheitsziele	Schutz vertraulicher Kundendaten kein Risiko durch Nutzung auf dem Privathandy	hat nichts zu verbergen (aber manche privaten Nachrichten sollte nicht unbedingt jeder sehen können)
Nutzungssituation	mobile Nutzung vor und nach Kundenterminen	ist mit dem Smartphone rund um die Uhr online

Abbildung1: Beispielpersonas für die Entwicklung einer Messenger-App

Die Rolle des Entwicklungsteams

Bei den bisherigen Bemühungen im Bereich Usable Security & Privacy standen meist die Sicherheitsbedürfnisse und -kenntnisse der Endanwender im Fokus [18]. Eine Schlüsselposition kommt jedoch auch dem Entwickler zu. Denn die Fähigkeiten und Unterstützung der Bedürfnisse des Entwicklungsteams sind maßgeblich für die Existenz und die Usability von Sicherheitsfunktionen, die später in Systemen vorhanden sind bzw. genutzt werden können. Auf der anderen Seite ist der Entwickler selbst Nutzer von Sicherheitsfunktionen in Form von Entwicklungswerkzeugen und wiederverwendbaren Softwarebausteinen. Hierbei ist er auf eine gute Usability angewiesen. Das Marktforschungsunternehmen International Data Corporation schätzte 2014 die Zahl der Personen im Bereich der Softwareentwicklung auf 18,5 Millionen weltweit [3]. Laut einer Prognose des Branchenverbands Bitkom arbeiteten 2016 hochgerechnet 802.000 Menschen in Deutschland im Bereich Software & IT-Services [7].

Angesichts dieser Zahlen wird deutlich, dass es sich bei Softwareentwicklern ebenso wie bei den Endanwendern um eine heterogene Gruppe von Menschen handelt. Das Qualitätsmerkmal Usability kann daher auch bei dieser speziellen Zielgruppe insbesondere durch die Betrachtung von Nutzertypen bei der Entwick-

lung von Sicherheitsmechanismen verbessert werden. In Bezug auf persönliche Merkmale ist bei Softwareentwicklern besonders auffällig, dass sie sich in der Art und Weise unterscheiden, wie sie programmieren bzw. wie sie mit Programmierschnittstellen (engl. Application Programming Interfaces, APIs) umgehen. Das individuelle Verhalten ist vornehmlich durch eine unterschiedliche Motivation geprägt, die auch eine gewisse Erwartungshaltung gegenüber einer API impliziert. Steven Clarke, der Versuchspersonen bei einer Vielzahl von Usability-Studien beobachtete, nahm eine Einteilung von Programmierern in drei verschiedene Nutzertypen vor [10]:

- Der opportunistische Entwickler konzentriert sich auf die schnelle Lösung seiner Aufgabe und geht dabei explorativ nach dem Bottom-up-Prinzip vor. Dieser Programmier-Typ ist am häufigsten unter den Entwicklern vertreten. Er beginnt direkt mit der Lösung seiner konkreten Problemstellung, indem er vor allem Komponenten mit einem hohen Abstraktionsniveau benutzt, wodurch er detaillierte Zusammenhänge ausblendet.

- Der gründliche Entwickler geht nach dem Top-down-Prinzip vor und verfolgt damit das gegensätzliche Handlungsmuster zum opportunistischen Entwickler. Dieser Programmier-Typ kommt nur selten vor. Bevor er eine API einsetzt, versucht er zunächst ein ganzheitliches Verständnis über die Technologie zu erlangen.

- Der pragmatische Entwickler kommt häufiger vor als der gründliche Entwickler, ist aber seltener anzutreffen als der opportunistische Entwickler. Auch in seinem Verhalten ist dieser Programmiertyp zwischen den beiden anderen Nutzertypen einzuordnen. Zuerst ähnelt das Handlungsmuster dem opportunistischen Entwickler. Er beginnt nach dem Bottom-up-Prinzip, um nach einer schnellen Problemlösung zu suchen. Stößt er hierbei allerdings an Grenzen oder Probleme, wechselt er zu einem Top-down-Ansatz, ähnlich dem Vorgehen des gründlichen Entwicklers.

Diese individuellen Eigenschaften sollten bei Designentscheidungen von Entwicklungswerkzeugen für Software und bei dem Entwurf von APIs bedacht werden. Ein Negativbeispiel macht deutlich, wie gravierend die Folgen sein können, wenn Security-Bausteine für Softwareentwickler Usabilitymängel aufweisen: Von zahlreichen Entwicklern, die Apps für Android und iOS entwickeln, wurden bestimmte Softwarekomponenten eingesetzt, um einen vertraulichen Nachrichtenaustausch mittels TLS (Transport Layer Security) [12] sicherzustellen, dem meistverbreiteten Sicherheitsprotokoll im Web. Die Namen dieser Werkzeuge versprachen eine einfache Anwendung (z. B. „SimpleX509TrustManager“ oder „EasySSLSocketFactory“). Tatsächlich führte deren (nicht korrekte, da zu komplizierte) Anwendung jedoch dazu, dass die damit entwickelten Apps jedem Zertifikat vertrauten. Die Folge: Der Schutzmechanismus der eigentlich implementiert werden sollte, ist nicht vorhanden [15, 16, 19].

Betroffen von den Usabilityproblemen sind in solchen Fällen nicht bloß einzelne Anwender, sondern ausnahmslos alle Nutzer einer App. Viele Entwickler verfügen jedoch nicht über die notwendigen Kenntnisse, um solche Implementierungen selbst zu validieren, und vertrauen auf die korrekte Funktionsweise der verwendeten Bausteine. Sie benötigen daher Werkzeuge, mit denen komplexe Sicherheits-, Datenschutz- und Usabilityanforderungen auf verständliche Art und Weise umgesetzt werden können und die Gefahr einer fehlerhaften Verwendung minimiert ist.

Weitere Systemteilnehmer, die Einfluss nehmen

Neben den Endanwendern und Entwicklern gibt es noch andere Akteure, die direkt oder indirekt Einfluss auf ein Produkt haben können [21]. Bei einer betrieblich genutzten Software sind dies z. B. die Lieferanten der Software, Systemintegratoren, Administratoren und Wartungspersonal sowie die Geschäftsführung des Unternehmens. Hinzu kommen oft noch Institutionen, deren Standards oder gesetzliche Anforderungen eingehalten werden müssen.

All diese Systemteilnehmer, die in Bezug auf das Produkt berechtigte Interessen haben, werden auch als Stakeholder bezeichnet.

Wichtig für das Verständnis dieser Stakeholder ist, dass sie sehr unterschiedliche Qualitätssichten auf ein interaktives Produkt haben können. Denn unterschiedliche Stakeholder legen in der Regel Wert auf unterschiedliche Qualitätsmerkmale, welche miteinander konkurrieren oder in Konflikt stehen können, wie durch die nachfolgenden Erläuterungen verdeutlicht wird [vgl. 36]:

- Die Betreiber stellen das Produkt den Benutzern zur Verfügung, organisieren die Nutzung und beeinflussen den Lifecycle. Daher ist die Qualitätssicht dieser Gruppe in der Regel einsatzorientiert. Als Sicherheitsmerkmale sind für diese Gruppe beispielsweise Zugangs- und Zugriffskontrolle sowie Integrität von gespeicherten und übertragenen Daten relevant.

- Die Architekten legen die technische Struktur des Produkts und die Aufgaben der einzelnen Komponenten fest.
  Die Qualitätssicht der Architekten zielt in der Regel auf die Befriedigung von Benutzer- und Betreiberanforderungen ab. Die Architekten sind ebenso wie die Gruppe der Programmierer auf Softwarebausteine angewiesen, die usable und sicher sind.

- Die Entwickler realisieren die Komponenten des Systems in Form von Programmen und Modulen. Ihre Qualitätssicht betrifft in der Regel die Programmstruktur, den Programmierstil und die einzelnen Algorithmen.

Alle Akteure im Überblick

Die Gesamtheit der Nutzer eines digitalen Produkts ergibt sich demnach aus den unterschiedlichen Nutzertypen (vgl. Beispiel in Abbildung 2): Entwickler treffen, wie oben beschrieben, programmatische und daher weitreichende Entscheidungen für die Software. Integratoren haben die Aufgabe, digitale Produkte in bereits bestehende Systemumgebungen einzubinden, während Administratoren für den produktiven Betrieb der Systeme verantwortlich sind, welche letztlich von Endanwendern genutzt werden.

Design- oder Anwendungsentscheidungen der genannten Stakeholder in Bezug auf Sicherheitsmechanismen wirken sich somit unmittelbar auch auf andere Nutzertypen aus.
Aus diesem Grund sollte das Ziel des Usable-Security-Engineering sein, möglichst lückenlos alle relevanten Nutzertypen eines digitalen Produkts zu berücksichtigen.
Die Herausforderung liegt insbesondere darin, ein digitales Produkt für die individuellen Personas der heterogenen Untergruppen effektiv, effizient und zufriedenstellend zu entwickeln.

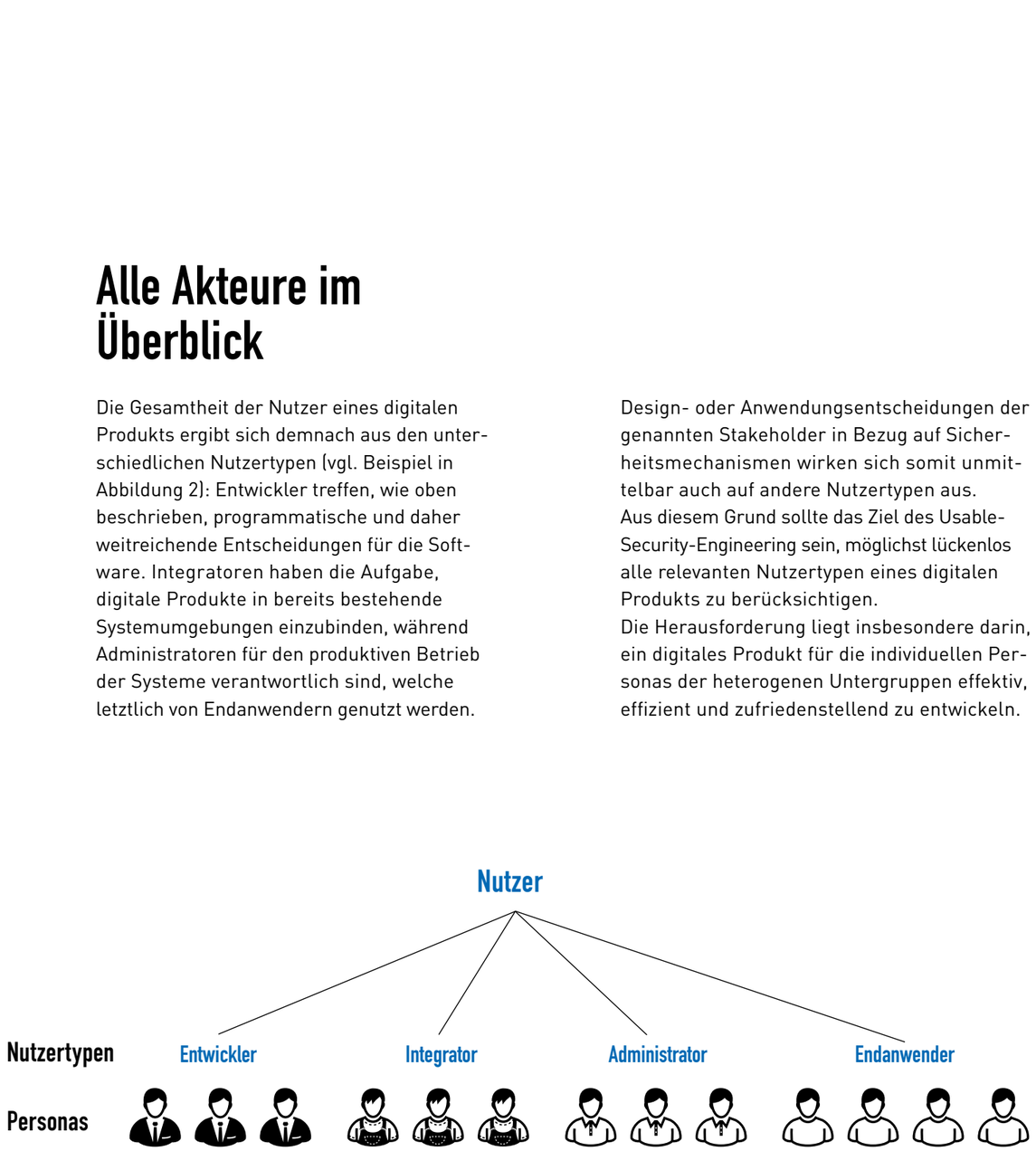


Abbildung 2: Eine Auswahl von Interessengruppen eines digitalen Systems, die es bei einem möglichst lückenlosen Usable-Security-Engineering zu berücksichtigen gilt

Kapitel 3

Wie gehe ich das an?

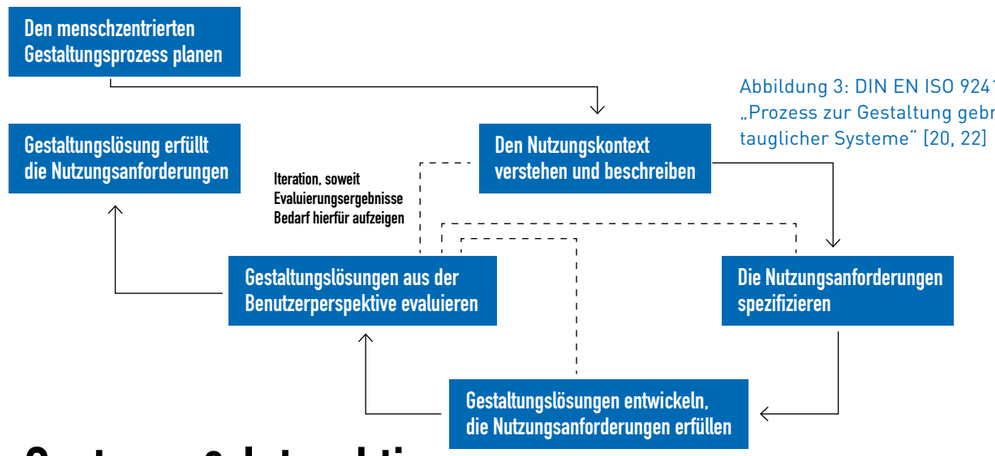


Abbildung 3: DIN EN ISO 9241-210 „Prozess zur Gestaltung gebrauchstauglicher Systeme" [20, 22]

Systeme & Interaktion nutzerzentriert gestalten

Das Einbeziehen von Nutzern in die Gestaltung von Systemen und Interaktion ist mittlerweile weitgehend etablierte Praxis. Während der 1980er Jahre entstanden erste Standardwerke zur Gewährleistung von Usability in Software [24]. In diesem Zuge reifte auch die Erkenntnis, dass Usability-Anforderungen nur oberflächlich standardisierbar sind. Je nach Anwendungsfall und Produkt müssten sie vielmehr mit und durch Nutzer selbst erhoben werden. Diese Orientierung am Nutzer mündete schließlich in der Idee des User Centered Design (UCD) [1, 22]. UCD versteht sich als interaktiver und iterativer Gestaltungsprozess mit Nutzern, in denen Gestalter durch Prototyping und Feedback-Schleifen Anforderungen an das Produkt erheben und das Verständnis darüber verbessern (vgl. Abbildung 3).

In einem in seinen Phasen jeweils iterativen Vorgehen zielt UCD zunächst darauf ab, Personas zu identifizieren und beispielsweise durch Beobachtungen, Interviews oder Fokusgruppen ein Problemverständnis aufzubauen.

Daraus werden erste Interaktionskonzepte sowie Use Cases abgeleitet und mit den Nutzern evaluiert und verfeinert. In einem dritten Schritt werden erste typischerweise Low-Level-Prototypen entworfen, die den Nutzern bereits ein Gefühl für die zukünftige Interaktion vermitteln sollen. Häufig eingesetzte Methoden zur Evaluation sind unter anderem Thinking-Aloud-Sessions, Interviews oder je nach Prototyp auch bereits ergänzend Eye- und Mouse-Tracking.

Ziel- und Hauptargument für die Anwendung eines UCD-Prozesses ist, die Passgenauigkeit des Produktes an die Anforderungen von Nutzern zu erhöhen, indem die Stakeholder selbst in den Gestaltungsprozess eingebunden werden. Dadurch soll auch das Risiko für Fehlentwicklungen minimiert werden. Insbesondere für die beiden Bedarfe nach Sicherheit und Usability vereinen Nutzer allerdings häufig sich widersprechende Interessen in sich. Studien zeigen, dass Nutzer häufig Kompromisse zwischen sicherer und gebrauchstauglicher Nutzung schließen, um ihre Ziele zu erreichen [32]. Die Erhebung solcher Bedarfe geschieht im UCD aber bisher meist beiläufig als zufälliges Nebenprodukt – wenn überhaupt. In Bezug auf Anforderungen an Usable Security & Privacy existieren daher für viele Anwendungen noch keine Best Pratices, geschweige denn gesicherte Forschungserfahrungen.

Für die Gestaltung sicherer Systeme gibt es eine Vielzahl an Vorgehensmodellen und Best Practices. Einerseits existieren organisatorische Sicherheitsvorgaben wie der IT-Grundschutz (ISO 27001) für typische IT-Systeme mit „normalem" (mittlerem) Schutzbedarf [9] . Daneben gibt es verpflichtende anwendungsspezifische Regelwerke für bestimmte Dienstleistungen wie beispielsweise die Abwicklung von Kreditkartentransaktionen (PCI DSS [25]). Außerdem existieren spezifische Rahmenwerke zur sicheren Software- und Produktentwicklung wie das Systems Security Engineering Capability Maturity Model (SSECMM [8]) oder die Common Criteria for Information Technology Security Evaluation [29] als internationaler Standard zur Prüfung und Bewertung der Sicherheitseigenschaften von IT-Produkten.

Weiterhin sind, um die Entwicklung sicherer Produkte zu unterstützen, neben Handlungsempfehlungen und Best Practices wie dem Code Review Guide [30] oder dem Development Guide des OWASP (Open Web Application Security Project) [31], auch einige Produktentwicklungsmodelle und -methoden entwickelt worden. Ein Beispiel dafür ist der Security Development Lifecycle (SDL) von Microsoft [23] (vgl. Abbildung 4). Als Erweiterung des klassischen Wasserfallmodells integriert dieses Modell über alle Entwicklungsphasen eine Reihe von Instrumenten und Verfahren, um sichere Software „by Design" zu erstellen.

In all diesen Rahmenwerken zur Informationssicherheit wird die Usability der Lösung allerdings kaum oder gar nicht berücksichtigt.

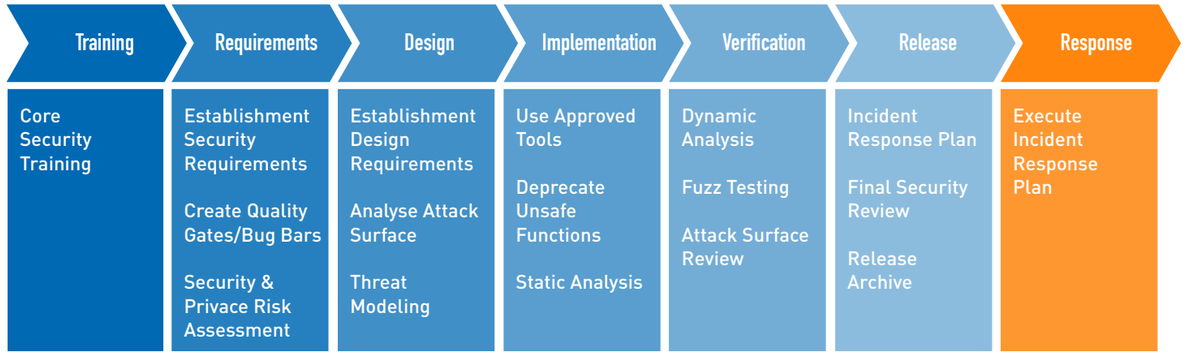


Abbildung 4: Security-Development-Lifecycle(SDL)-Prozess von Microsoft [23]

Herausforderungen von Usable Security & Privacy

Sowohl für UCD wie für die Informationssicherheit (IS) ist es essentiell wichtig, Usability und Sicherheit bzw. Privatheit als integralen Teil in Organisationsprozessen und speziell in Produktentwicklungsprozessen zu verankern. Zielkonflikte zwischen Usability und Security bzw. Privacy entstehen u. a., weil UCD- und IS-Prozesse trotz einer Vielzahl von Standards, Richtlinien, Best Practices, Rahmenwerken und Prozessmodellen in der Praxis weitgehend getrennt voneinander ausgeführt werden. Weder bestehen explizite Schnittstellen für die Einbindung von Nutzern in den IS-Prozess, noch wird im UCD die Informationssicherheit behandelt.

Wir argumentieren hier allerdings nicht für ein allgemeingültiges integriertes Vorgehensmodell. Ob es Bedarf für ein neues Usable-Security-Modell gibt oder eine Anreicherung bereits bestehender Modelle mit Erkenntnissen und Tools aus der Usable Security geeigneter ist, hängt von der jeweiligen Organisationsstruktur ab, in der ein System gestaltet wird. Stattdessen geben wir lieber konkrete Vorschläge, wie für UCD-Prozesse Usable Security & Privacy effektiv umgesetzt werden kann.

Anforderungsebene: Interessenkonflikte bleiben unsichtbar

Produkte müssen nicht nur die Zielkonflikte einzelner Nutzer bezüglich Sicherheit oder Gebrauchstauglichkeit berücksichtigen und lösen. Gerade bei vernetzten Produkten und datenbasierten Dienstleistungen bestehen auch Interessenkonflikte zwischen den Stakeholdern. So können z. B. bestimmte Anwender den Wunsch haben, ihre Daten auch Drittanbietern für erweiterte Dienstleistungen zur Verfügung zu stellen, während die Betreiber dies aus Sicherheitsgründen nicht zulassen möchten. Andererseits können Anwender Schutzbedarfe für ihre Daten haben, während Betreiber ein Interesse an der Auswertung oder der Weitergabe personalisierter Daten verfolgen.

Die Erhebung und Aushandlung dieser Interessen im Entwicklungsprozess ist wichtig, um Security und Privacy als kritische Anforderungen für die Nutzerakzeptanz im Gestaltungsprozess zu berücksichtigen. Dazu müssen nicht notwendigerweise neue Methoden erschaffen werden, sondern es können auch schon bestehende Werkzeuge angepasst werden, um für solche Interessen sensibel zu sein.

Umsetzungsebene: Effizientes Programmieren für Usable Security & Privacy

Aus IS-Perspektive wird der Nutzer hauptsächlich als Schwachstelle gesehen und hat die Verpflichtung, die ihm geltenden Sicherheitsmaßnahmen strikt umzusetzen. Werden diese jedoch vom Nutzer aufgrund von schlechter Usability nicht korrekt ausgeführt oder sogar umgangen, wird der Fehler in der Regel beim Nutzer gesucht, nicht in der Sicherheitsmaßnahme selbst. Um das Programmieren von Interfaces unter Berücksichtigung von Usable Security & Privacy effektiv zu fördern, benötigt es aus unserer Sicht daher Werkzeuge, die nachvollziehbare Prinzipien, Richtlinien und Patterns zur Verfügung stellen, die generisch angewandt werden können.

Sicherheit im User Centered Design verankern

Im Folgenden zeigen wir exemplarisch, wie ein UCD-Prozess erweitert werden kann, um Schutzinteressen der Stakeholder explizit mit in die Entwicklung einzubinden und zugleich ein effizientes Programmieren für Usable Security & Privacy zu unterstützen. Eine Übersicht hierfür liefert Abbildung 5. Dabei wird in frühen Phasen der Produktentwicklung insbesondere auf eine Sensibilisierung und Erweiterung der bestehenden Methoden Wert gelegt. In späteren Gestaltungsphasen kann weiter auf konkrete Tools und Werkzeuge zurückgegriffen werden, um in den Lösungen, die in Betracht gezogen werden, gebrauchstaugliche Sicherheit zu gewährleisten. Die konkreten Methoden zur Umsetzung werden im anschließenden Kapitel 4 beschrieben.

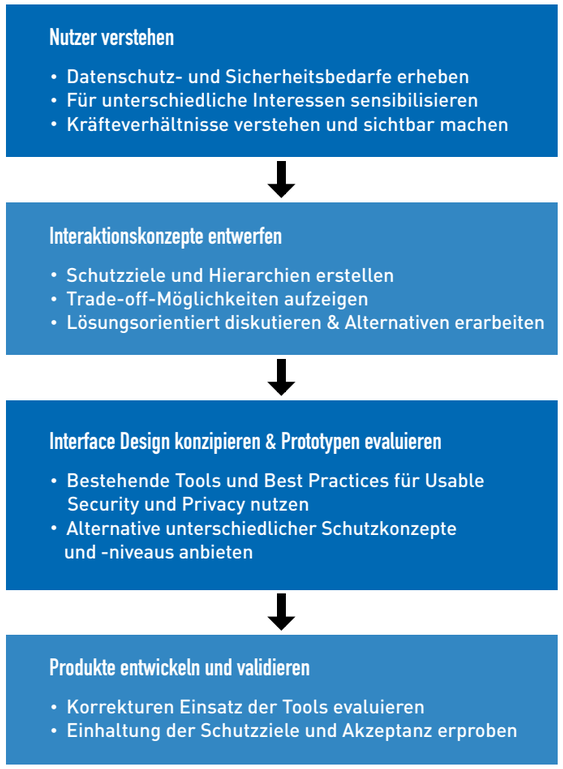


Abbildung 5: Exemplarische Erweiterung eines User-Centered-Design-Prozesses für die Entwicklung von gebrauchstauglichen Sicherheitskomponenten in Systemen

Nutzer verstehen

Um die Handlungsmotivationen der Stakeholder nachzuvollziehen, werden die jeweiligen Schutzziele und Use Cases erhoben und gegenüber gestellt. Dadurch sollen Konflikte für alle Stakeholder sichtbar gemacht werden und ein möglichst genaues Verständnis von Schutzbedarfen und Nutzungsinteressen erstellt werden. Hierzu kann es aus Datenschutzperspektive auch gehören, Daten selbst für Stakeholder zunächst einmal nachvollziehbar und damit diskutierbar zu machen. In diesen Prozess sollten (siehe Kapitel 2) nicht nur die unmittelbaren Endanwender des Systems, sondern explizit alle an einem Produkt mittelbar und unmittelbar partizipierenden Nutzer beteiligt werden.

Interaktionskonzepte entwerfen

Für die identifizierten Use Cases müssen den Stakeholdern die jeweiligen Schutzinteressen und die Mehrwertpotentiale von Produktentscheidungen dargelegt werden. Es ist für die spätere Abwägung wichtig, diese von Nutzern erfassten und bewerteten Vor- und Nachteile in der Konzeption von Interaktionskonzepten zu berücksichtigen und diese ebenso später im Produkt weiter sichtbar zu halten.

Interface Design konzipieren & Prototypen validieren

In dieser Phase sollten die entworfenen Alternativen mit geringem Aufwand umgesetzt werden. Hier ist es wichtig, Mehrwerte und Bedrohungen offen zu kommunizieren, so dass bei den Stakeholdern ein Verständnis über mögliche Alternativen aufgebaut werden kann.

Eine effektive Umsetzung von Usability in diesem Zusammenhang sollte bereits in diesem Stadium auch durch den Rückgriff auf Best-Practice-Sammlungen und einheitliche Gestaltungswerkzeuge unterstützt werden.

Produkt entwickeln und evaluieren

In der Phase der Entwicklung kommt der Umsetzung von Gestaltungspatterns auf Basis erhobener Anforderungen eine noch höhere Bedeutung zu. Diese Patterns tragen dazu bei, die Sicherheit der Systeme zu fördern, indem Schutzmechanismen gebrauchstauglich und anwenderfreundlich umgesetzt werden. Insbesondere kann aber die Effektivität und Effizienz der Umsetzung erhöht werden, in dem auf einen breiten Pool von Patterns und Best Practices zurückgegriffen werden kann.

Interdisziplinäre Awareness und Kooperation

Abseits einer Prozessintegration ist es wünschenswert, Usability und Security auch organisatorisch zu verzahnen. So wie die meisten IS-Modelle auf Trainings von Endanwendern, Administratoren und Entwicklern setzen, sollten IS-Verantwortliche, Administratoren und Entwickler über die Wichtigkeit von Usability für die Informationssicherheit aufgeklärt und zu den grundsätzlichen Usability-Konzepten geschult werden. Umgekehrt sollten Usability-Ingenieure und Interface-Designer fundamentale IS-Konzepte verstehen, um auch die Sicherheit von Software nicht zu vernachlässigen. Prinzipiell sollten alle sicherheitsrelevanten Teile von Produkten, die Auswirkung auf die Benutzungsschnittstelle haben oder Prozessschritte beeinflussen, an denen Menschen beteiligt sind, auch im UCD-Prozess behandelt werden.

Fazit

Sowohl im UCD als auch im Bereich der Informationssicherheit gibt es etablierte Prozesse und standardisierte Vorgehensmodelle. Über eine Erweiterung des traditionellen UCD-Prozesses können Herausforderungen der Sicherheit und Privatsphäre mit Gebrauchstauglichkeit in Einklang gebracht werden. Dies gelingt durch eine Anpassung bzw. Erweiterung bestehender Vorgehensmodelle und Methoden und eine Einbindung spezifischer Best Practices und Werkzeuge in den bestehenden Entwicklungsprozess. Für die Beteiligten in einem solchen Prozess heißt das, insbesondere in früheren Phasen interdisziplinär und organisationsübergreifend auf Schutzziele der verschiedenen Stakeholder hinzuweisen und auf den verschiedenen Ebenen (Nutzer, Hersteller und Betreiber) für Interessen und Anforderungen sensibilisiert zu sein.

In der gesamten Entwicklung sollte auf die Erhebung und den Einsatz von gebrauchstauglichen Schutzmechanismen geachtet werden. Daneben sollten in der gesamten Organisation Prinzipien, Richtlinien und Patterns von Usable Security bekannt sein und es sollten explizit Verantwortlichkeiten geklärt und ihre Umsetzung etabliert sein. So können von Beginn an Sicherheitsfunktionalitäten entwickelt werden, die auf den Benutzer zugeschnitten sind und die sich aufgrund von empirischen Untersuchungen auch tatsächlich als gebrauchstauglich erwiesen haben (und nicht bereits vor dem Beginn eines Entwicklungsprozesses zum Scheitern verurteilt sind).

Kapitel 4

Wie setze ich das um?

Werkzeuge als transdisziplinäre Grundpfeiler

In den vorherigen Kapiteln wurde bereits der hohe Bedarf an gebrauchstauglichen Sicherheitskomponenten in der Softwarebranche herausgestellt. Für Softwarearchitekten und Entwickler bedeutet das, dass sie das Qualitätsmerkmal Usable Security & Privacy vermehrt berücksichtigen und umsetzen müssen. Bestenfalls werden ihnen dabei leichtgewichtige Methoden und Werkzeuge aus Forschung und Praxis an die Hand gegeben, die bei der Entwicklung von digitalen Produkten mit gesondertem Fokus auf Usable Security & Privacy unterstützen. Nach Möglichkeit sollten diese Werkzeuge einfach in den verwendeten Gestaltungsprozess integriert und sowohl beim Design und der Implementierung, als auch bei der Evaluierung der Produkte eingesetzt werden können.

Besonders wirkungsvoll ist die Unterstützung durch Werkzeuge in den frühen Entwicklungsphasen, also „by Design“, da hiermit konzeptionelle Entwurfsfehler vermieden werden können, die ansonsten erst spät erkannt werden und in der Folge aufwändig zu beseitigen sind (oder nur noch dokumentiert werden können). In späteren Phasen der Entwicklung ist die Anwendung von Werkzeugen jedoch ebenfalls sinnvoll, um Prototypen auf Usable-Security-Kriterien hin zu überprüfen.

Zu derartigen Entwurfswerkzeugen in der Softwareentwicklung zählen u. a. Prinzipien (engl. principles), Richtlinien (engl. guidelines), Musterlösungen (engl. patterns). In diesem Kapitel werden diese Werkzeugarten vorgestellt und deren Einsatzgebiete und Anwendungsformen näher erläutert.

Begriffsbestimmungen

Die Begrifflichkeiten für Softwareentwicklungswerkzeuge werden in Literatur und Praxis nicht konsequent verwendet. Wie in Abbildung 6 dargestellt, bietet sich eine Unterscheidung der Werkzeuge über ihren Abstraktionsgrad an. Prinzipien sind nach unserem Verständnis sehr abstrakte Werkzeuge und fassen allgemeine Entwurfsgrundsätze in kurzen Leitsätzen zusammen. Richtlinien sind bereits konkreter und beschreiben die Umsetzung eines oder mehrerer Prinzipien in einem bestimmten Kontext. Patterns wiederum bieten konkrete Lösungsvorschläge für gängige Problemstellungen an, die in einem spezifischeren Anwendungskontext stehen. Gegebenenfalls beinhalten Patterns Implementierungshinweise, wobei sie jedoch so abstrakt bleiben sollten, dass sie nicht nur eine Lösung in einer bestimmten Programmiersprache oder für eine konkrete Anwendung bieten.

Im Rahmen des öffentlich geförderten Projekts USecureD [37] wurden viele dieser Werkzeuge für den Bereich Usable Security in einer systematisierten und konsolidierten Sammlung auf einer kostenfreien und deutschsprachigen Plattform veröffentlicht.

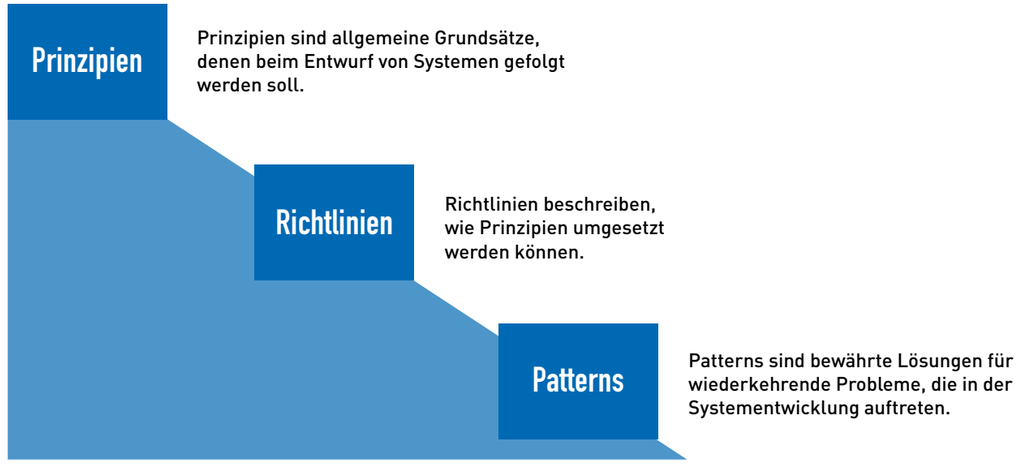


Abbildung 6: Abstraktionsgrad und Definitionen der einzelnen Werkzeugkategorien

Besonders effektiv wird eine Sammlung von Entwurfswerkzeugen, wenn werkzeugübergreifende Verknüpfungen hergestellt werden. So ist es beispielsweise möglich, zu analysieren, welche Richtlinien und Prinzipien thematisch miteinander in Verbindung stehen. Damit wird Anwendern, die ihre Anwendung in Hinblick auf ein konkretes Prinzip optimieren wollen, eine Empfehlung für spezifischere Werkzeuge zur Umsetzung gegeben. Bestenfalls steht damit eine durchgängige Werkzeugkette zur Verfügung, die für jedes beschriebene Prinzip passende Richtlinien und Patterns empfiehlt.

Prinzipien für Usable Security

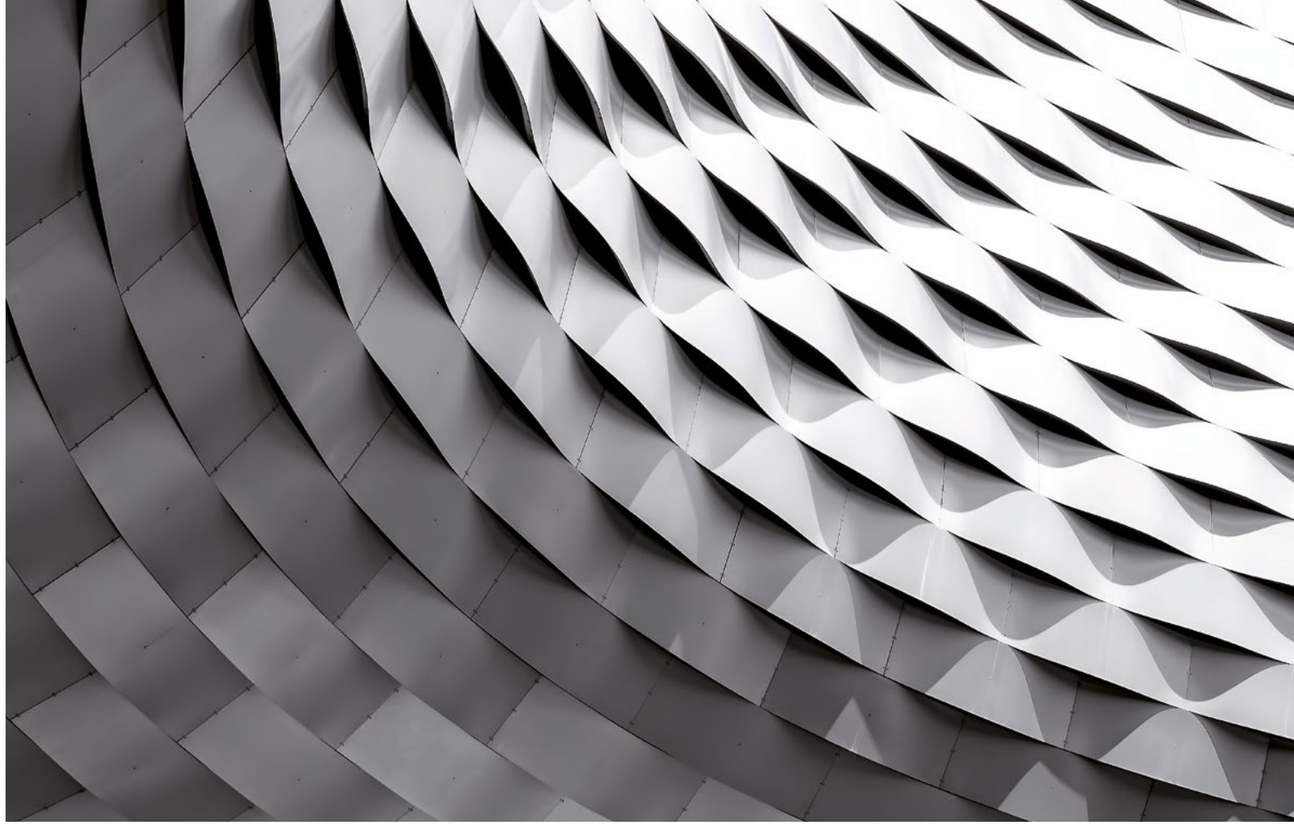
Der erste Werkzeugtyp, welchen wir vorstellen wollen, sind Usable-Security-Prinzipien. Sie basieren auf Erfahrungswerten oder wissenschaftlichen Erkenntnissen und eignen sich aufgrund der kurzen Darstellung u. a. als Einstiegsmedium in die Thematik Usable Security. Hauptsächlich werden sie jedoch während der Konzeptionsphase von Softwareprojekten eingesetzt, um Softwarearchitekten und -entwickler bereits zu Beginn der Systementwicklung für das Qualitätsmerkmal Usable Security zu sensibilisieren.

Eine deutschsprachige Sammlung für Usable-Security-Prinzipien steht in Form einer Internetplattform zur Verfügung und umfasst zum Zeitpunkt der Veröffentlichung dieser Fachschrift 23 Usable-Security-Prinzipien verschiedener Autoren (**https://das.th-koeln.de/usecured/principles**). Zur einheitlichen und strukturierten Dokumentation wurde eine Beschreibungsvorlage entwickelt, welche die typischen und wiederkehrenden Merkmale von Prinzipien abbildet. Die relevantesten sind die Intention (Was soll durch das Prinzip erreicht werden?) und die Motivation (Aus welchem Grund ist das Prinzip wichtig?). Weitere Merkmale sind die Quellen, aus denen das Prinzip hervorgeht, Synonyme, unter denen das Prinzip ebenfalls bekannt ist, und Beispiele, in denen die Anwendung des Prinzips deutlich wird. Schließlich ist eine Verknüpfung mit anderen Werkzeugen hergestellt worden, indem Richtlinien für Usable Security identifiziert wurden, die sich thematisch auf dieses Prinzip beziehen.

Richtlinien für Usable Security

Entwicklungsrichtlinien setzen in den meisten Fällen ein oder mehrere Prinzipien in einem spezifischeren Kontext um. Sie sind wichtig, damit bereits während der Planung und Implementierung von Systemen möglichst viele Ursachen für spätere Schwachstellen eliminiert werden können [6]. Somit ist es möglich, dass Richtlinien zum einen zur Gewährleistung eines hohen Qualitätsstandards beitragen und zum anderen die Komplexität der Entwicklungsprojekte, in denen sie angewendet werden, verringern.

Eine Sammlung möglichst praxistauglicher Richtlinien, wie z. B. Usability Guidelines, die für Sicherheitskomponenten anwendbar sind oder Security Guidelines, die sich durch ein hohes Maß an Usability empfehlen, ist ebenfalls auf der bereits erwähnten Plattform veröffentlicht worden (**https://das.th-koeln.de/usecured/guidelines**). Auch hier folgt die Dokumentation einer einheitlichen Beschreibungsvorlage. Neben Quellen, Synonymen, Beispielen und einer Verknüpfung mit verwandten Richtlinien liegt hier der Schwerpunkt auf dem jeweiligen Kontext, aus dem sich einzelne Richtlinien ergeben und für den sie zutreffend sind.



Patterns für Usable Security

Der Entwicklungsprozess von Sicherheitsfunktionalitäten lässt Softwarearchitekten und Softwareentwickler auf wiederkehrende Probleme treffen. Nicht alle diese Probleme müssen dabei von jedem Entwicklungsteam neu gelöst werden. Für viele Problemstellungen sind bewährte, gut dokumentierte und wiederverwendbare Musterlösungen – sogenannte Patterns – vorhanden.

Christopher Alexander [2] nutzte den Begriff im Bereich der Architektur, jedoch wurde er später auch für die Dokumentation von Problemlösungen bei Softwareentwicklungsprozessen eingeführt [5, 17]. Seitdem sind Patterns fester Bestandteil der Softwarebranche und dienen dem Wissenstransfer von bewährten Problemlösungen bei Design- und Implementierungsfragen. Dies spiegelt sich in einer Vielzahl an Literatur und Datenbanken wider, in denen Patterns für unterschiedlichste Softwareentwicklungsbereiche dokumentiert sind. Dabei werden zahlreiche Themenbereiche abgedeckt, von grundsätzlichen, strukturellen Problemstellungen der Softwarearchitektur bis hin zu speziellen Themen wie Security [27], User Interface Design [33] und Usability [26]. So ist es nicht verwunderlich, dass auch im Bereich Usable Security & Privacy mit der Zeit Patterns entwickelt wurden. Diese behandeln typischerweise die folgenden Themenbereiche:

- Authentifizierung, Autorisierung und Schlüsselmanagement
- Signatur und Verschlüsselung von Kommunikationskanälen
- sicheres Löschen von Daten und Anlegen von Sicherungskopien
- gebrauchstaugliche Bedienelemente für Sicherheitsfunktionen
- Gestaltung von Hinweisen, Warnungen und Sicherheitsstatus

Auch hier folgt die Dokumentation im USecureD-Projekt einer einheitlichen Beschreibungsvorlage. Aufgrund des im Vergleich zu Prinzipien und Richtlinien niedrigen Abstraktionsniveaus von Patterns umfasst die eingesetzte Beschreibungsvorlage deutlich mehr Merkmale. Neben den in der Wissenschaft und Praxis gängigen Attributen Kontext, Problem und Lösung wurden auch Hinweise zur Implementierung und zu Konsequenzen berücksichtigt, die bei der Anwendung eines Patterns auftreten können. Das Attribut Prinzipien verdeutlicht, dass mit einem Pattern ein bestimmtes Prinzip umgesetzt wird.

Werden thematische Verknüpfungen der Patterns untereinander analysiert, dann entsteht eine sogenannte Pattern Language. Den Anwendern einer Pattern Language ermöglichen diese thematischen Verknüpfungen ein schnelles Identifizieren von weiteren, zum Aufgabengebiet passenden Musterlösungen. Um diese Beziehungen schnell erfassen zu können und zugleich aufgabenorientiert innerhalb der Sammlung navigieren zu können wurde eine interaktive Grafik zur Visualisierung erstellt (**https://das.th-koeln.de/usecured/patterns**).
Die Plattform bietet zusätzlich die Möglichkeit des maschinellen Zugriffs auf die Werkzeuge im kompakten Datenformat JSON [13]. Über eine Programmierschnittstelle (API) können Unternehmen die Werkzeuge in eigene Prozesse integrieren und für Qualitätserhebungen oder Assessments verwenden (**https://das.th-koeln.de/usecured/assets/api/swagger.json**).

Die genannten Werkzeuge fokussieren bisher fast ausschließlich auf Usable Security. Usable Privacy ist noch weitestgehend unerforscht und daher sind für diesen Bereich auch kaum konkrete Prinzipien, Richtlinien oder Patterns verfügbar.

“If it’ s not usable, it’s not secure.”

Jared Spool

Kapitel 5

Kann ich das auch mal in Aktion sehen?

Fallstudien zur beispielhaften Veranschaulichung

In diesem Kapitel stellen wir zwei konkrete Beispiele vor, bei denen Usable Security und Privacy eine wesentliche Rolle gespielt haben. Einerseits haben wir bei der Implementierung intelligenter Messsysteme einen UCD-Prozess begleitet, der Sicherheits- und Datenschutzinteressen unterschiedlicher Stakeholder analysiert und ausgehandelt hat, so dass gebrauchstaugliche und sichere Lösungen entstehen konnten. Als zweites Beispiel demonstrieren wir die konkrete Anwendung und Einbettung unterstützender Tools in den Entwicklungsprozess von Software zur Berücksichtigung von Usable Security & Privacy.

Fallstudie Smart Grid

In diesem Abschnitt verdeutlichen wir die Umsetzung des in Kapitel 3 eingeführten UCD-Prozesses mit besonderer Berücksichtigung der sicherheits- und datenschutzrelevanten Funktionen. Als Anwendungsbeispiel dient dabei die Einführung intelligenter Messsysteme und die konfligierenden Interessen der Stakeholder, in diesem Fall Netzbetreiber, Energieversorgungsunternehmen (EVU), Bundesamt für Sicherheit in der Informationstechnik (BSI) und Endkunde.

Problemstellung

Intelligente Messsysteme sind die Herzstücke für die Realisierung des sogenannten Smart Grid. Sie bestehen aus digitalen Stromzählern (Smart Meter) und Smart Meter Gateways, die die sichere Kommunikation der Smart-Meter-Daten gewährleisten sollen. Durch die digitale Erfassung von Verbrauchswerten auf Abnehmerebene nahezu in Echtzeit verspricht sich der Netzbetreiber auf der einen Seite eine effizientere Lenkung des Stroms und der Energieversorger auf der anderen Seite ein genaueres Lastmanagement, so dass weniger überschüssige Energie vorgehalten werden muss. Das BSI setzt enge technische Vorgaben, um die Sicherheit der Stromversorgung als kritische Infrastruktur sicher zu stellen. Der Verbraucher kann nicht zuletzt durch niedrigere Stromkosten als auch flexible Tarife und ansprechendes Energie-Feedback auf Basis von Webtechnologien profitieren. Auf der anderen Seite kann die Übertragung von Stromverbrauchsdaten Verhaltensmuster sowie generelle Informationen wie z. B. die Anzahl der Personen im Haushalt offenlegen.

Für das dargelegte Szenario hat ein Systementwickler daher folgende Aufgabenstellung zu bewältigen: Wie kann ich unterschiedlichen und konkurrierenden Qualitätssichten der Stakeholder eines digitalen Produkts gerecht werden und dabei eine für möglichst alle Nutzer gebrauchstaugliche und zugleich sichere Architekturlösung entwerfen? Das nachgelagerte Interface-Design selbst ist nicht mehr Teil des Fallbeispiels.

Methodik

Nach der Identifikation der Stakeholder begannen wir gemäß dem in Kapitel 3 beschriebenen Prozess mit der Analyse der jeweiligen Interessen und Anforderungen an das Konzept der Letztverbraucherschnittstelle. Wir nutzten dazu Interviews und Fokusgruppen (insbesondere bei der Gruppe der Konsumenten) sowie einen Prototypen zur Visualisierung von Stromverbrauchsdaten. Diese leiteten wir ab zu alternativen Lösungen, die die Bedarfe der Stakeholder in unterschiedlichem Maße adressieren würden, und gaben diese als Empfehlungen weiter.

BSI

Da unser Projekt mit Verspätung startete, konnten wir auf die Vorgaben des BSI nicht mehr einwirken. Das BSI sah die Schnittstelle prinzipiell verpflichtend vor, legte aber keine technische Ausgestaltung fest. De facto war es mit der hardwareseitigen Bereitstellung einer LAN-Schnittstelle getan und die Anbindung von dort an wurde offengelassen. Gerade in Mehrfamilienhäusern liegen Stromzähler aber häufig an entlegenen Orten, so dass die Überbrückung der „letzten Meile" eine zentrale Frage für den weiteren Prozess darstellte.

EVU/Messstellenbetreiber

Diese Stakeholder waren nicht interessiert an potentiellen Mehrwerten für Kunden. Die Ausführung sollte somit so minimal wie möglich sein, um Kosten zu sparen. Gleichzeitig sollte die Sicherheit allerdings an erster Stelle stehen. Eine Anbindung der Schnittstelle an das LAN des Verbrauchers würde allerdings potenzielle Angriffsvektoren ermöglichen. Die beiden einzigen Optionen waren somit, entweder eine lokale Lösung wie z.B. ein Display zu finden oder die Schnittstelle komplett zu schließen.

Konsument

Zunächst fiel auf, wie schwierig es Konsumenten fiel, über Energieverbrauch und Energie als Ressource zu sprechen. Durch den Einsatz von Soft- und Hardware-Prototypen zur Sammlung und Visualisierung der eigenen Stromverbrauchsdaten änderte sich dies insbesondere bei weniger technikaffinen Konsumenten. Im Folgenden erkannten die Konsumenten aber zunehmend Schutzbedarfe in ihren Daten und äußerten den Bedarf einer Kontrolle. Im Hinblick auf ein gewünschtes Frontend zur Einsicht und Steuerung von Verbrauchs- und Vertragsdaten wurden Webseiten als favorisiertes Medium genannt (gegenüber lokalen Displays am Zähler selbst).

Generelle Guidelines

Bei der Umsetzung der Nutzerschnittstelle sollte auf BITV- bzw. WCAG-2.0-Level-III-Konformität geachtet werden. Dies umfasste auch die notwendigen Informationen, um kompetente Sicherheitsentscheidungen treffen zu können. Die Informationen sollten in einer für den Nutzer verständlichen Form und Sprache präsentiert werden. Dabei sollte nicht über das Niveau der niedrigen sekundären Schulbildung hinausgegangen werden.

Zudem sollten die für den Endverbraucher entstehenden zusätzlichen Kosten berücksichtigt werden. Dabei war in der Gesamtbetrachtung zu prüfen, ob das Mehr an Sicherheit in einem vertretbaren Verhältnis zu den Mehrkosten steht.

Außerdem war darauf zu achten, dass sich die Schutzmaßnahmen in die Alltagsroutinen und Handlungsabläufe der Nutzer integrieren lassen, um keine zusätzlichen praktischen Barrieren aufzubauen. So erfüllt z. B. ein lokales, am Zähler montiertes Display zwar hohe Sicherheitsansprüche, jedoch reduziert sich hierdurch die Gebrauchstauglichkeit der intelligenten Messsysteme für die Kunden, da die Zähler schwer zugänglich sein können. Insbesondere Mieter haben ohne Hilfe des Vermieters oft auch gar keinen Zugang zu ihrem Stromzähler.

Ergebnisse

Aus unseren empirischen Untersuchungen konnten wir in Einklang mit technischen und rechtlichen Rahmenbedingungen sowohl Design-Guidelines als auch konkrete Architektur-Gestaltungsvorschläge für eine Smart Meter Gateway-Schnittstelle für den Letztverbraucher ableiten.

Auf Basis der Stakeholder-Gespräche, der Einordnung der Gefahren- und Mehrwertpotentiale sowie der technischen Vorgaben des BSI haben wir fünf mögliche Alternativen für Verbraucherschnittstellen erarbeitet.

Diese haben wir auf Basis der empirischen Analysen im Hinblick auf die vier Merkmale „transparenter Datenschutz", „Usability", „Barrierefreiheit" und „Kosten" bewertet (vgl. Tabelle 1). Jedem Nutzungsszenario steht dabei ein Katalog mit technischen und organisatorischen Maßnahmen gegenüber, mit denen die jeweiligen (Schutz-)Interessen in unterschiedlichem Ausmaß wirkungsvoll umgesetzt werden können.

Name	Datenschutz	Usability	Accessibility	Kosten
Lokales Display	+	+	–/0	–
Lokales Ethernet Interface	+	+	–/0	–/0
Lokales PLC/WiFi	0/+	+	0/+	0/+
Proprietäres Web-Portal	–	0/+*	+	+
Standardisierte Web-API	–/0	+*	0/+	+

Tabelle 1: Fallstudie Smart Grid: Tendenzielle Beurteilung der einzelnen Umsetzungsszenarien im Vergleich. Eine genaue, belastbare Bewertung hängt von der jeweiligen konkreten Umsetzung ab und muss im Einzelfall erfolgen.

Die Einzweckanzeige (Single-purpose Display) dient dazu, die Energiedaten lokal am Zähler zu visualisieren, so dass Verbrauchsdaten die Liegenschaft des Letztverbrauchers nicht verlassen müssen. Beim lokalen Display wird schon konzeptionell eine hohe technische Sicherheit implementiert. Ferner kann die Einzweckanzeige speziell auf die Bedarfe des gebrauchstauglichen und barrierefreien Verbrauchsfeedbacks abgestimmt werden. Als Nachteil ergeben sich deutliche Auswirkungen auf die Entwicklungs- und Produktionskosten. Ein weiterer Nachteil besteht darin, dass die Zähler sich meist mehr oder weniger zugänglich im Keller befinden.

Um Kosten zu senken, könnte auch einfach eine LAN-Schnittstelle für jeden Haushalt bereitgestellt werden, über die eine lokale Webseite des intelligenten Messsystems abgerufen werden kann. Hier besteht allerdings die Herausforderung, die Brücke vom Zähler bis in das LAN des Haushaltes sicher zu gewährleisten. Hier wäre technisches Know-how des Verbrauchers gefragt. Um an dieser Stelle zu unterstützen, könnte das Messsystem selbst mit weiteren Kommunikationsmitteln wie einem WiFi-, 3G- oder Powerline-Communication-Modul ausgestattet sein. Diese erhöhen die Kosten zwar moderater als ein lokales Display, dafür sind allerdings die Zugangs- und Einstiegshürden deutlich höher. Zugleich eröffnen diese Technologien durch ihre Architektur neue Angriffsvektoren, die insbesondere Energieversorger minimieren möchten.

Eine Alternative dazu besteht in einem proprietären Webportal, das z. B. durch den Energieversorger betrieben wird und auf das die Haushalte zugreifen können. Durch sie würden Energiedaten aber „by default" an Dritte weitergegeben; dem Endanwender bliebe keine Wahl. In Bezug auf Kosten und Usability bietet diese Variante allerdings Vorteile. Zur Abmilderung der Datenschutz-Problematik und zur Vermeidung proprietärer Insellösungen gilt es die zugehörigen Datenformate und Übertragungsprotokolle zu standardisieren. Insbesondere würden so Lock-in-Effekte minimiert und es kann von einer hohen Usability ausgegangen werden, da der Kunde den Anbieter frei wählen kann. Die Accessibility könnte gering niedriger ausfallen, da eine einmalige proaktive Auswahl durch den Verbraucher notwendig ist.

Um diese Interessen auszutarieren, hat sich das BSI schließlich für einen regulatorischen Eingriff entschieden. Dieser orientierte sich an einer modifizierten Version des proprietären Web-Portals. Es wurde zur weiteren Erhöhung des Datenschutzniveaus die Rolle des Daten-Treuhänders geschaffen, der die Daten bereitstellt und sicher vor Drittverwerter-Interessen für den Verbraucher verwahrt. Auf diese Weise wurde den zentralen Interessen des Datenschutzes und der Usability auf Verbraucherseite einerseits, sowie der Vermeidung von Zusatzkosten andererseits weitgehend Rechnung getragen.

Fazit

In diesem Beitrag haben wir gezeigt, dass die Frage nach sicheren intelligenten Messsystemen nicht losgelöst von der Frage nach Gebrauchstauglichkeit diskutiert werden sollte. Aus Nutzersicht gilt insbesondere, dass der Mehrwert des direkten und informativen Verbrauchsfeedbacks trotz Sicherheitsmaßnahmen erhalten bleiben muss. Ferner dürfen die praktischen Hürden und zusätzlichen Kosten nicht ausgeblendet werden, wenn ein Mehr an Datenschutz und Datensicherheit nicht nur auf dem Papier erreicht werden soll.

Fallbeispiel Computer-Sicherheitswarnungen

Anhand des Beispiels von Computer-Sicherheitswarnungen soll in diesem Abschnitt veranschaulicht werden, wie ein tool-orientiertes Vorgehen die Gebrauchstauglichkeit von Sicherheitsfunktionen in Softwareprodukten effektiv verbessern kann. Als leichtgewichtige Usable-Security-Werkzeuge eignen sich insbesondere die in Kapitel 4 vorgestellten Prinzipien, Richtlinien und Patterns.

Problemstellung

Der Sinn und Zweck einer Computer-Sicherheitswarnung ist es, den Benutzer während der Interaktion mit einem digitalen Produkt auf potenzielle Risiken aufmerksam zu machen. Damit sind Sicherheitswarnungen ein wichtiges Mittel, um die unterschiedlichen Nutzertypen (vgl. Kapitel 2) vor einem drohenden Schaden zu bewahren. Wie eine Computer-Sicherheitswarnung im Detail aussieht und zu welchem Zeitpunkt diese angezeigt wird, kann z. B. in der Verantwortung eines Softwareentwicklers liegen. Die Tragweite dieser Designentscheidungen ist bei erfolgreichen Softwareapplikationen, die einen hohen Verbreitungsgrad erreichen, enorm hoch, da eine Vielzahl von Endanwendern mit den implementierten Meldungen interagiert. Die Usable-Security-Forschung hat gezeigt, dass Endanwender eine ablehnende Haltung gegenüber Sicherheitswarnungen einnehmen bzw. diese ignorieren, wenn menschliche Eigenschaften und Verhaltensweisen bei der konkreten Ausgestaltung unzureichend berücksichtigt werden [14].

Im geschilderten Szenario hat ein Softwareentwickler daher folgende Aufgabenstellung zu bewältigen: Wie kann ich eine Computer-Sicherheitswarnung effektiv implementieren, um dadurch die Risiken der Benutzer möglichst gering zu halten?

Methodik

Antworten auf diese Frage liefert z. B. eine Richtlinie des CyLab-Instituts der Carnegie-Mellon-Universität, die aus insgesamt sechs Handlungsempfehlungen besteht [4]:

1. Beschreibe das bestehende Risiko vollständig

2. Sei in der Beschreibung prägnant und präzise

3. Biete aussagekräftige Handlungsoptionen an

4. Stelle relevante Informationen zum spezifischen Kontext zur Verfügung

5. Liefere relevante Auswertungsergebnisse

6. Verfolge ein konsistentes Layout

Diese Handlungsempfehlungen können im Rahmen eines benutzerzentrierten Entwicklungsprozesses sowohl beim Entwurf als auch bei der Bewertung und Verbesserung bereits eingesetzter Warnmeldungen als Werkzeug herangezogen werden. Folgt man systematisch den Kriterien der Richtlinie, erhält man im Ergebnis ein Produkt, dessen Qualitätsmerkmal Usable Security deutlich vom aktuellen Stand der Wissenschaft profitiert.

Ergebnisse

Abbildung 7 zeigt als Negativbeispiel eine SSL/TLS-Warnung, wie sie im Jahr 2008 im Firefox-Browser der Organisation Mozilla eingesetzt wurde. Hier sollte die Internetseite mit dem Namen beispiel.de über eine sichere Verbindung aufgerufen werden. Kann die gesicherte Verbindung nicht automatisch aufgebaut werden, wird dies dem Nutzer über eine SSL/TLS-Warnung mitgeteilt. Dieser muss daraufhin eine Entscheidung bezüglich seines weiteren Vorgehens treffen. Die Folge des optischen Designs und der schwachen inhaltlichen Ausgestaltung ist, dass die Mehrzahl der Nutzer in dieser Situation die Meldung ignorieren und schlicht auf den „Continue"-Button klicken würde, um mit der unterbrochenen Handlung fortzufahren [28]. Die Umsetzung ist nicht gebrauchstauglich, denn sie schafft es nicht, dem Nutzer das potenzielle Risiko der Gefahrensituation zu vermitteln.

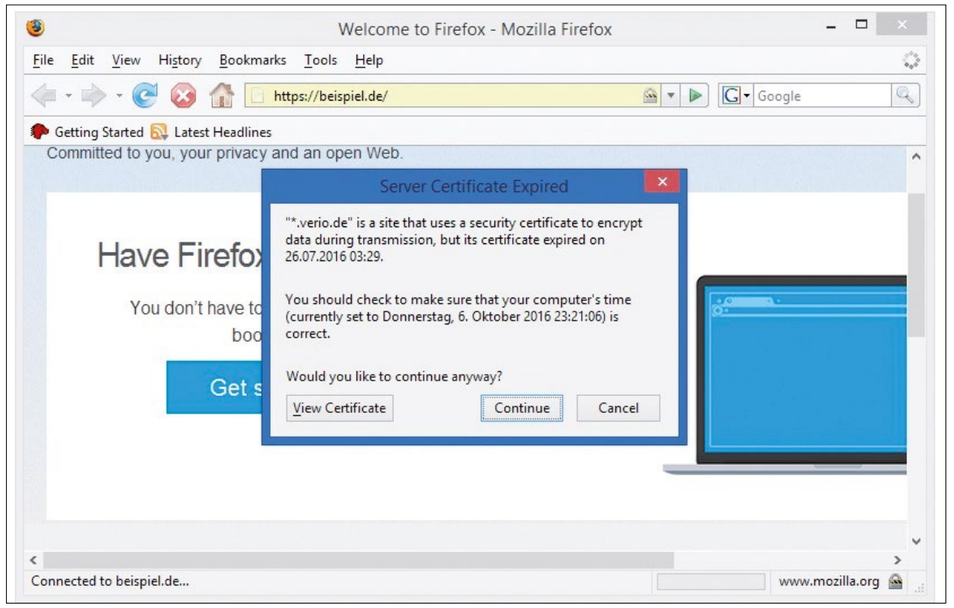


Abbildung 7: SSL-Warnung des Firefox Browsers von Mozilla (Version 2.0.0.17, 2008)

Das Potenzial der genannten Richtlinie kann praxisnah anhand der zeitlichen Weiterentwicklung dieser SSL/TLS-Warnung demonstriert werden. Abbildung 8 zeigt im Vergleich das Design der gleichen Meldung aus dem Jahr 2017.

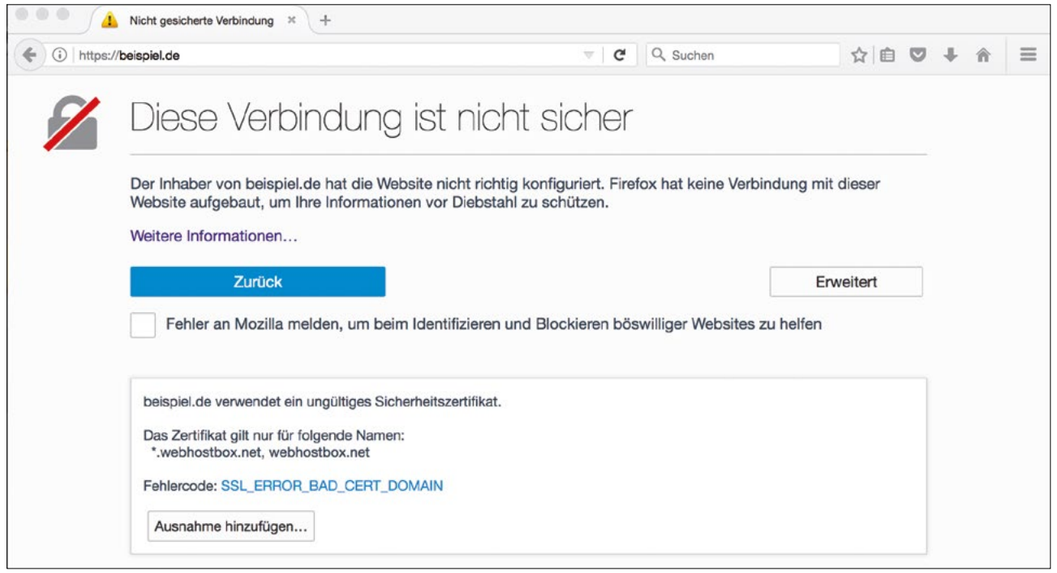


Abbildung 8: TLS-Warnung des Firefox Browsers von Mozilla (Version 51.0.1, 2017)

Die sechs Handlungsempfehlungen der Richtlinie werden exemplarisch an beiden Entwürfen erläutert.

1. Beschreibe das bestehende Risiko vollständig

Die textuelle Ausarbeitung der Meldung aus dem Jahr 2008 beinhaltet keine Beschreibung des bestehenden Risikos. Diese Information benötigt der Nutzer aber, um eine informierte Entscheidung darüber treffen zu können, ob er die Handlung abbrechen oder ohne Berücksichtigung der Warnmeldung fortfahren möchte. Die Meldung aus dem Jahre 2017 warnt hingegen vor der Konsequenz, dass Informationen des Endanwenders gestohlen werden könnten. Eine vollständige Beschreibung des Risikos kann zusätzliche Informationen über die konkreten Gefährdungen enthalten wie z. B. den unbefugten Zugriff auf Konto- und Kreditkartendaten oder Passwörter.

2. Sei in der Beschreibung prägnant und präzise

Die Beschreibungen einer Computer-Warnmeldung sollten wichtige Informationen in knapper Form und möglichst genau und zutreffend beschreiben. Technische Begriffe und Details sollten vermieden werden, um nicht von den wesentlichen Informationen abzulenken. Die Warnmeldung aus 2008 liefert eher technisch ausgerichtete Informationen bezüglich des Zertifikats. In der aktuelleren Warnmeldung von 2017 werden Inhalte dieser Art nur in dem erweiterten Teil angezeigt.

3. Biete aussagekräftige Handlungsoptionen an

Der Endanwender hat in der Warnmeldung von 2008 die Handlungsoptionen fortzufahren (Continue) oder den Vorgang abzubrechen (Cancel). Die Buttons wurden nebeneinander platziert und gleich gestaltet. Es ist nicht auf einen Blick erkennbar, welche der beiden Auswahlmöglichkeiten eine sichere Standardoption darstellt. Die Handlungsoptionen der Meldung von 2017 sind „Zurück“ und „Ausnahme hinzufügen…“. Das Design symbolisiert sehr deutlich die Handlungsoption „Zurück“ als empfohlene Option. Zudem muss der Endanwender zunächst die erweiterten Informationen einblenden, also noch einen zusätzlichen Schritt gehen, bevor er die unsichere Handlung fortsetzen kann. Kurze Bezeichnungen wie „Fortfahren“ „Abbrechen“ oder „Zurück“ haben allerdings alleinstehend wenig Aussagekraft in Bezug auf die damit verbundenen Konsequenzen. Eine bewusste Auseinandersetzung des Endanwenders mit der Warnmeldung kann durch eine kurze eindeutige Beschreibung wie z. B. „Diese Warnung ignorieren (unsicher)“ oder „Zurück zu sicherer Webseite“ gefördert werden [11, 14].

4. Stelle relevante Informationen zum spezifischen Kontext zur Verfügung

Um den Bezug einer Warnmeldung zu einer konkreten Situation zu verdeutlichen, sollten verfügbare Kontextinformationen in die Warnmeldung integriert werden. Das Ziel dieses Vorgehens ist auch bei dieser Empfehlung, dem Endanwender die Entscheidung für eine Handlungsoption zu erleichtern. Die gezeigten Warnmeldungen greifen z. B. den Namen der aufgerufenen Internetseite beispiel.de sowie die im Zertifikat gelisteten Hostnamen und die Gültigkeitsdauer auf. Die Verfügbarkeit solcher Informationen ist dabei abhängig von der jeweiligen Situation.

5. Liefere relevante Auswertungsergebnisse

Die Auswertung von Nutzungsdaten kann sinnvolle Kontextinformationen generieren, welche dem Benutzer zur Bewertung einer Warnmeldung an die Hand gegeben werden können. In Logdateien gesammelte Daten (z. B., was für eine Aktion zu welchem Zeitpunkt und mit welchem Ergebnis durchgeführt wurde) können dabei helfen, ungewöhnliche Ereignisse zu identifizieren und gegebenenfalls davor zu warnen. Die Meldung aus 2017 bietet in diesem Kontext die Option „Ausnahme hinzufügen“ an, um das zukünftige Verhalten des Browsers zu beeinflussen. Besucht der Nutzer diese Seite zu einem späteren Zeitpunkt erneut, wird diese Warnmeldung nicht mehr angezeigt.

6. Verfolge ein konsistentes Layout

Bauer et. al [4] schlagen in ihrer Richtlinie ein konsistentes Layout für Computer-Sicherheitswarnungen vor. Zunächst sollten diese Warnungen keinen Button in der rechten oberen Ecke anbieten (wie das Negativbeispiel in Abbildung 7), um die Meldung nicht schließen zu können, ohne eine Entscheidung getroffen zu haben. Die Dringlichkeit der Situation sollte durch ein Icon kommuniziert werden. In der aktuellen Version wird ein rot durchgestrichenes Schloss eingesetzt, welches die fehlende Sicherheit symbolisiert. Kritische Warnungen sollten andere Inhalte in der Anwendung gänzlich überlagern und die Interaktion mit diesen unterbinden. SSL/TLS-Warnungen in Browserumgebungen sollten dem Anwender daher als fensterfüllende Seite angezeigt werden. Die in Abbildung 8 gezeigte Warnmeldung wird in Form eines modalen Dialogs angezeigt, welcher sich über das Browserfenster legt. Vom Design her gibt es keinen Unterschied zu anderen Meldungen des Betriebssystems (in diesem Fall Windows). Der Benutzer kann also nicht auf den ersten Blick erkennen,

dass es sich hierbei um eine Warnmeldung handelt, die sich explizit auf die Handlungen im Browser bezieht. Textuell sollte eine effektive Computer-Sicherheitswarnung einen kurzen primären Text ähnlich einer Schlagzeile aufweisen. Es kann sinnvoll sein, sekundären Text mit erweiterten Informationen erst auf Wunsch des Nutzers anzuzeigen und typografisch weniger prägnant als den primären Text zu gestalten. Der Nutzer sollte durch eine konkrete Frage zum weiteren Vorgehen angesprochen werden. Als Antwort auf die gestellte Frage folgen direkt darunter mindestens zwei Handlungsoptionen. Sekundäre Optionen, die keine direkte Antwort auf die gestellte Frage liefern, sollten weniger prominent am unteren Bildrand platziert werden. Im direkten Vergleich setzt die aktuelle SSL/TLS-Warnung des Browsers Firefox deutlich mehr Aspekte des vorgeschlagenen Layouts um als die Meldung von 2008.

Fazit

Sowohl die Fallstudie Smart Grid also auch das Fallbeispiel der Computer-Sicherheitswarnungen zeigen, dass die Qualität von Softwarelösungen durch die Anwendung von Usable-Security-Methoden verbessert werden kann. Insbesondere ist es dadurch möglich Produkte zu entwickeln, die sowohl gebrauchstauglich als auch sicher sind.

Kapitel 6

Wie mache ich von hier aus weiter?

Teilnahme an Aktivitäten des Arbeitskreises

Die Aktivitäten des Arbeitskreises gestalten sich entsprechend der Zielsetzung, sowohl im privaten als auch im geschäftlichen Umfeld ein stärkeres Bewusstsein für das Thema Usable Security & Privacy zu schaffen, in vielfältiger Weise, z. B. durch Fachvorträge, wissenschaftliche und praxisorientierte Workshops, Präsentationen bei UX-Stammtischen oder durch sonstige öffentlichkeitswirksame Aktionen oder Veröffentlichungen wie das erste Positionspapier zu aktuellen Problemfeldern und Herausforderungen aus Sicht der UX Professionals. Neben weiteren Veröffentlichungen dieser Art sind auch Poster, White Papers oder praxisorientierte Checklisten geplant. Wir wollen ein Forum für den gemeinsamen Gedankenaustausch und die interdisziplinäre Zusammenarbeit zum Thema anbieten sowie vorhandenes Wissen durch die Bereitstellung empfehlenswerter Literatur, Methoden, Best Practices, Patterns und Guidelines konsolidieren und praxistauglich aufbereiten.

Die Mitglieder des Arbeitskreises organisieren sich dafür über eine Mailingliste sowie durch regelmäßige Telefon- bzw. Videokonferenzen, die je nach zeitlicher Verfügbarkeit alle paar Wochen in Abstimmung durch die Teilnehmer

stattfinden. Nach Möglichkeit werden halbjährliche Vor-Ort-Treffen organisiert, um gemeinsam an aktuellen Themen zu arbeiten. Im Rahmen der Tagungsreihe „Mensch und Computer" finden zudem jährliche Treffen sowie gemeinsame Workshops statt. Auch sonst können sich Interessenten auf vielfältige Weise einbringen.

Mitmachen und Vernetzen

Forscher und Praktiker aus anderen Disziplinen wie z. B. dem Security Engineering, die sich ebenfalls mit der Thematik beschäftigen, sind eingeladen, sich mit den Mitgliedern des Arbeitskreises in Verbindung zu setzen und neue Brücken zu schlagen, um das Forschungsfeld gemeinsam voranzubringen. Wer sich dauerhaft engagieren möchte, kann sich als Mitglied der German UPA direkt beteiligen und sich jederzeit per E-Mail an den Leiter des Arbeitskreises Hartmut Schmitt (**Hartmut.Schmitt@germanupa.de**) wenden.

Weitere Organe, Verbände und Institute

GI – Gesellschaft für Informatik
Fachbereich „Sicherheit – Schutz und Zuverlässigkeit": http://fb-sicherheit.gi.de/struktur/fachgruppen.html
Fachbereich „Mensch-Computer-Interaktion": https://fb-mci.gi.de/

BITKOM – Bundesverband Informationswirtschaft Telekommunikation und neue Medien e.V.: https://www.bitkom.org/Themen/Datenschutz-Sicherheit/Datenschutz-Sicherheit

Helmholtz Center for Information Security (CISPA), Saarbrücken: https://www.cispa.saarland

CRISP – Center for Research in Security and Privacy, Darmstadt: https://www.crisp-da.de

KASTEL – Kompetenzzentrum für angewandte Sicherheits- Technologie KASTEL, Karlsruhe: https://www.kastel.kit.edu

Alexander von Humboldt Institut für Internet und Gesellschaft (HIIG): https://www.hiig.de

nrw.uniTS – IT-Sicherheit Nordrhein-Westfalen:http://www.nrw-units.de/nrwunits/

Bundesamt für Sicherheit in der Informationstechnik (BSI): https://www.bsi.bund.de

Competence Center for Applied Security Technology (CAST) e.V.:https://www.cast-forum.de

Stiftung Datenschutz:
http://www.stiftungdatenschutz.org/

Forum Privatheit:https://www.forum-privatheit.de

Stiftung Neue Verantwortung – Think Tank für die Gesellschaft im technologischen Wandel: https://www.stiftung-nv.de

Work Group „Usable Security And Privacy", Universität Bonn: https://net.cs.uni-bonn.de/work-groups/usecap/

Forschungsgruppe SECUSO, Karlsruher Institut für Technologie: https://secuso.aifb.kit.edu/

Institut für Softwaretechnik und Theoretische Informatik, Quality and Usability Lab an der TU Berlin, Forschungsgruppe „Usable security & privacy":http://www.qu.tu-berlin.de/menue/forschung/gruppen/usp

"What usable security really means: trusting and engaging users."

Iacovos Kirlappos, M. Angela Sasse

Kapitel 7

Quellen-verzeichnis

1. Abras C, Maloney-Krichmar D, Preece J (2004) User-centered design. In: Encyclopedia of Human-Computer Interaction. Sage Publications 37, Thousand Oaks, S. 445–456

2. Alexander C, Ishikawa S, Silverstein M (1977) A Pattern Language: Towns, Buildings, Construction. Oxford University Press USA

3. Avram A (2014) IDC Study: How Many Software Developers Are Out There? http://www.infoq.com/news/2014/01/IDC-software-developers

4. Bauer L, Bravo-Lillo C, Cranor L, Fragkaki E (2013) Warning Design Guidelines. CyLab Carnegie Mellon University

5. Beck K (1996) Smalltalk Best Practice Patterns. Prentice Hall, Upper Saddle River

6. Birolini A (1997) Zuverlässigkeit von Geräten und Systemen. Springer Berlin Heidelberg, Berlin, Heidelberg

7. Bitkom (2019) ITK-Arbeitsmarkt. https://www.bitkom.org/Marktdaten/ITK-Arbeitsmarkt/index.jsp

8. Bitkom (2019) ISO/IEC 21827 (SSE-CMM) Capability Maturity Model (SSE-CMM®) / Model der Ablaufstauglichkeit. http://www.kompass-sicherheitsstandards.de/Evaluierung-von-IT-Sicherheit/ISO-IEC-21827-SSE-CMM

9. Bundesamt für Sicherheit in der Informationstechnik (2016) IT-Grundschutz-Kataloge. Standardwerk zur Informationssicherheit. 15. Ergänzungslieferung. Bundesanzeiger Verlag, Köln

10. Clarke S (2010) How Usable Are Your APIs? In: Oram A, Wilson G (Hrsg) Making software: what really works, and why we believe it, 1st ed. O'Reilly, Beijing, S. 545–565

11. Cranor LF (2008) A Framework for Reasoning About the Human in the Loop. In: Proceedings of the 1st Conference on Usability, Psychology, and Security. USENIX Association, Berkeley, CA, USA, S. 1:1–1:15

12. Dierks T, Rescorla E (2008) RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2.

13. Ecma International (2017) Standard ECMA-404 The JSON Data Interchange Syntax. https://www.ecma-international.org/publications/standards/Ecma-404.htm

14. Egelman S, Cranor LF, Hong J (2008) You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, S 1065–1074

15. Fahl S, Harbach M, Muders T et al (2012) Why Eve and Mallory love android: an analysis of android SSL (in)security. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM Press, Raleigh, NC, USA, S. 50

16. Fahl S, Harbach M, Perl H et al (2013) Rethinking SSL development in an appified world. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM Press, Berlin, S. 49–60

17. Gamma E, Helm R, Johnson R, Vlissides J (1994) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley

18. Garfinkel S, Lipford HR (2014) Research on usable security: history, themes, and challenges. Morgan & Claypool, San Rafael

19. Georgiev M, Iyengar S, Jana S et al (2012) The most dangerous code in the world: validating SSL certificates in non-browser software. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM Press, Raleigh, NC, USA, S. 38

20. German UPA e.V., Arbeitskreis Qualitätsstandards (2016) German UPA Qualitätsstandard für Usability Engineering v1.1.

21. Holz T, Pohlmann N, Bodden E et al (2016) Human-Centered Systems Security - IT-Sicherheit von Menschen für Menschen. https://www.ptj.de/lw_resource/datapool/systemfiles/cbox/2021/live/lw_file/strategiepapier_it-sicherheit.pdf

22. International Organization for Standardization (2011) ISO 9241-210:2010 Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme; Deutsche Fassung EN ISO 9241-210:2010.

23. Microsoft (2012) Microsoft Security Development Lifecycle (SDL) - version 5.2 - Introduction. https://msdn.microsoft.com/de-de/library/windows/desktop/cc307406.aspx

24. Norman DA (1988) The psychology of everyday things. New York : Basic Books

25. Microsoft (2012) Microsoft Security Development Lifecycle (SDL) – version 5.2. https://docs.microsoft.com/en-us/previous-versions/windows/desktop/cc307748(v%3dmsdn.10)

26. Röder H (2012) Katalog Usability Patterns – Version: 1.2. http://www.usabilitypatterns.info/catalog/catalog.html

27. Steel C, Nagappan R, Lai R (2005) Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management. Prentice Hall PTR, Upper Saddle River

28. Sunshine J, Egelman S, Almuhimedi H et al (2009) Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In: Proceedings of the 18th Conference on USENIX Security Symposium. USENIX Association, Berkeley, CA, USA, S. 399–416

29. The Common Criteria (2017) Common Criteria : New CC Portal. https://www.commoncriteriaportal.org/

30. The Open Web Application Security Project (2016) OWASP Code Review Guide Project. https://www.owasp.org/index.php/Category:OWASP_Code_Review_Project#tab=Main

31. The Open Web Application Security Project (2016) OWASP Developer Guide. https://www.owasp.org/index.php/OWASP_Guide_Project

32. Whitten A, Tygar JD (1999) Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. Proceedings of the 8th USENIX Security Symposium 1999

33. Interaction Design Foundation (2019) User Interface (UI) Design Patterns. https://www.interaction-design.org/literature/topics/ui-design-patterns

34. Deutschland sicher im Netz e.V. (2016) DsiN-Sicherheitsindex 2016. https://www.sicher-im-netz.de/node/1510

35. Usability in Germany (UIG) e.V. (2016): Personas. https://www.usability-in-germany.de/definition/personas

36. Wallmüller E (2002): Qualitätsmodelle im Software Engineering: Boden unter den Füssen. In: MQ - Management und Qualität 2002 (9). Galledia Verlag, Berneck

37.USecureD-Konsortium (2017): UsecureD – Usable Security by Design. https://www.usecured.de

Kapitel 8

Autoren

Luigi Lo Iacono ist Leiter der Gruppe für Daten- und Anwendungssicherheit an der Technischen Hochschule Köln.

Seine Forschungs- und Entwicklungsinteressen liegen im Umfeld der Sicherheit verteilter Systeme und der Usability dieser.

Hartmut Schmitt ist Koordinator für Forschungsprojekte beim saarländischen IT-Lieferanten HK Business Solution GmbH. Er ist seit 2006 in Verbundvorhaben auf den Gebieten Mensch-Computer-Interaktion, Usability/User Experience und Software-Engineering tätig, u. a. als Projektkoordinator in mehreren BMBF- und BMWi-geförderten Verbundvorhaben. Hartmut Schmitt ist Mitglied der Gesellschaft für Informatik und der German UPA, bei der er den Arbeitskreis „Usable Security & Privacy" leitet.

Denis Feth studierte Informatik an der Technischen Universität Kaiserslautern. Seit 2011 arbeitet er am Fraunhofer IESE in Kaiserslautern. Als Senior Engineer ist er dort verantwortlich für die Software-Entwicklung in der Abteilung „Security Engineering". Wissenschaftlich beschäftigt er sich hauptsächlich mit den Themen „Datennutzungskontrolle" sowie „Usable Security und Privacy".

Timo Jakobi ist wissenschaftlicher Mitarbeiter an der Universität Siegen und der Hochschule Bonn-Rhein-Sieg. Sein Forschungsschwerpunkt liegt auf der Entwicklung gebrauchstauglicher Unterstützungsmechanismen für das Management von Privatsphäre in IKT-Anwendungen. Hier stellt insbesondere der Trend zum Internet of Things mit der Anbindung und Analyse unterschiedlichster und abstrakter Daten(-quellen) eine neue Herausforderung dar, um Anwendern Transparenz und Kontrolle über die eigenen Daten zu verleihen..

Peter Leo Gorski ist wissenschaftlicher Mitarbeiter in der Gruppe für Daten- und Anwendungssicherheit an der Technischen Hochschule Köln. Dort arbeitet er an wissenschaftlichen Problemstellungen der Bereiche Service Security und Usable Security. Sein Forschungsinteresse liegt insbesondere auf der gebrauchstauglichen Sicherheit im Kontext der Softwareentwicklung.

Markus Dölle studierte Psychologie mit dem Schwerpunkt „Kognitive Ergonomie" an der Humboldt-Universität zu Berlin und arbeitet seit 2010 als freiberuflicher UX Consultant.
Mit den Themen „Security und Privacy" beschäftigt er sich seit geraumer Zeit und engagiert sich nicht nur in Arbeitsprojekten für die Berücksichtigung der Nutzerinteressen in Bezug auf ihre Daten. Er ist im Organisationsteam des Berliner „World Usability Day" und veranstaltet den „UX/Usability Stammtisch Berlin".

Peter Nehren

ist wissenschaftlicher Mitarbeiter in der Gruppe für Daten- und Anwendungssicherheit der TH Köln. Seine Forschungs- und Entwicklungsschwerpunkte liegen auf Werkzeugen für Usable Security.

Edna Kropp

arbeitet als Usability-Beraterin und forscht zum Thema Usability in der Software-Entwicklung an der Freien Universität Berlin. Ein Schwerpunkt ihrer Arbeit ist die Integration von Human-Centered-Design in Software-Entwicklungsprozesse. Sie ist im Organisationsteam des World Usability Day Berlin.

Sarah Hausmann

ist wissenschaftliche Mitarbeiterin am Institut für Ubiquitäre Mobilitätssysteme an der Hochschule Karlsruhe. Ihr Forschungsschwerpunkt liegt im Bereich der Usable Privacy und Privacy-Awareness in ubiquitären Systemen. Dabei werden Benutzern mögliche Datenfreigaben und deren Konsequenzen transparent dargestellt, um eine gezielte Kontrolle über die Freigabe ihrer Daten zu gewährleisten. In ihrer Diplomarbeit entwickelte sie ein generisches Konzept für usable Privacy-Awareness mobiler Applikationen. Teile dieses Konzepts flossen in die während des Forschungsprojekts „GO Karlsruhe!“ entwickelte Mobilapplikation ein.

Anne Hofmeister

ist wissenschaftliche Mitarbeiterin an der Technischen Hochschule Köln im Labor für Kommunikationstechnik und Datensicherheit. Sie schreibt ihre Masterthesis für ihren Abschluss in Medieninformatik im Bereich Usable Security. Der Fokus liegt auf der Identifikation von Qualitäten, die Sicherheitsmechanismen und sicherheitsrelevante Anwendungen neben reiner Usability aufweisen müssen, um die Usability der primären Aufgaben der Nutzer nicht so zu reduzieren bzw. die User Experience in einem Maße zu verschlechtern, dass die Nutzer sich gegen eine Verwendung entscheiden oder versuchen diese zu umgehen, ohne aber die Effektivität der Sicherheitsmaßnahme zu reduzieren.

Arkadiusz Frydyada de Piotrowski

ist freiberuflicher IT-Berater und hat über zehn Jahre Berufserfahrung sowohl im Usability-Engineering als auch in der Informationssicherheit. Dabei hat er in unterschiedlichen Rollen gearbeitet, als Requirements Engineer, Datenbankentwickler, Programmierer, IT-Security-Consultant und Manager und kennt deshalb viele Aspekte der Usable Security. Er ist Gründungsmitglied des AK „Barrierefreiheit“ der German UPA und hat auf der Mensch und Computer 2009 den Best Paper Award erhalten.

Mandy Balthasar

ist als wissenschaftliche Mitarbeiterin an der Universität der Bundeswehr München am Institut für Softwaretechnologie. Als IT-Consultant und zertifizierte Datenschutzbeauftragte wirkt sie in den Centern of Competence „Web Technologies“ und „User Experience“ mit und legt dabei ihren Themenfokus im Rahmen der IT-Sicherheit besonders auf Datenschutz, Mensch-Computer-Interaktion und Software-Engineering. Dabei ist sie in unterschiedlichen Rollen als Entwicklerin, IT-Security-Consultant und Security Managerin unterwegs. Zudem ist sie als Lehrende an verschiedenen Hochschulen und Fachhochschulen sowie für die Springer Nature Campus GmbH im Einsatz.

Der Arbeitskreis Usable Security & Privacy in der German UPA e.V.

Der Arbeitskreis Usable Security & Privacy bietet den Mitgliedern der German UPA ein Forum für den Gedankenaustausch und die interdisziplinäre Zusammenarbeit rund um das Thema benutzerfreundliche Informationssicherheit.

Der Arbeitskreis beschäftigt sich mit Ansätzen und Konzepten, die sicherheitsfördernde Verfahren für Software und interaktive Produkte stärker an den Zielen und Aufgaben der Nutzer ausrichten und die dafür sorgen, dass Funktionsweisen von Sicherheitselementen auch für Nichtexperten verständlich sind.

Durch den Arbeitskreis soll sowohl bei privaten Endanwendern als auch im geschäftlichen Umfeld ein stärkeres Bewusstsein für das Thema Usable Security & Privacy geschaffen werden.

Kontakt und Mitarbeit
E-Mail an: **ak-usable-security-privacy@germanupa.de**

Weitere Informationen unter:
www.germanupa.de/arbeitskreise/arbeitskreis-usable-security-privacy

Impressum

Usable Security & Privacy
Herausgegeben von German UPA e.V.

Ansprechpartner
Hartmut Schmitt (AK-Leitung)

Autoren der ersten Auflagen
Luigi Lo Iacono, Hartmut Schmitt, Denis Feth, Timo Jakobi, Peter Leo Gorski, Peter Nehren, Markus Dölle, Edna Kropp, Sarah Hausmann, Anne Hofmeister, Arkadiusz Frydyada de Piotrowski, Mandy Balthasar

Kontakt
German UPA e.V.
Keplerstraße 2
39104 Magdeburg

Herausgabedatum
3. Aktualisierte Ausgabe September 2019

Die Fachschrift ist als barrierefreie PDF-Version gemäß WCAG 2.0 AA und PDF/UA auf der Internetseite des AK Usable Security & Privacy verfügbar:
www.germanupa.de/arbeitskreise/arbeitskreis-usable-security-privacy

German UPA e.V.
Keplerstraße 2, 39104 Magdeburg
www.germanupa.de

GERMAN **UPA**
Berufsverband der Deutschen Usability
und User Experience Professionals