Proseminar *Entwicklung sicherer Software*
Wintersemester 2008/2009

Usability: Nutzbarkeit und IT-Sicherheit

Björn Kahlert
kahlert@inf.fu-berlin.de
Betreuer: Dipl. Medieninf. Martin Gruhn

23.02.2009

Zusammenfassung

Nutzbarkeit und IT-Sicherheit gelten allgemeinhin als anti-proportional, d.h. je sicherheitsrelevanter die Software desto benutzerunfreundlicher ist sie. Labortests bestätigen, dass selbst auf Benutzerfreundlichkeit ausgerichtete Sicherheitssoftware vom Großteil der Testteilnehmer nicht richtig verwendet werden konnte. Daher erfordert die Entwicklung zugleich nutzbarer als auch sicherer Software interdisziplinäre Forschung, neue Designrichtlinien und Implementierungsverfahren.

Inhaltsverzeichnis

1 Einleitung

IT-Sicherheit war in den frühen Jahren des Computergeschichte ein stiefmütterlich behandeltes Thema. Die damals bis heute existierenden Designrichtlinien und Implementierungsverfahren haben neben der Benutzerfreundlichkeit vor allem die Robustheit der Software zum Ziel.

Allerdings steigt mit der zunehmenden Verbreitung und Bedeutung von informationstechnischen Systemen, deren Missbrauchspotential. Dies führte unweigerlich zur Entwicklung sicherer Software, um beispielsweise den Versand verschlüsselter anstelle von Klartext-Mails zu ermöglichen.

Das neue Kriterium IT-Sicherheit fand jedoch keinen ganzheitlichen Einzug in bestehende Softwareentwicklungsverfahren. Daraus resultiert das fundamentale Problem, dass sich IT-Sicherheit und Nutzbarkeit auszuschließen scheinen.

Aber warum muss sichere Software überhaupt benutzerfreundlich sein? - Weil die Sicherheit mit der Benutzerfreundlichkeit steht und fällt. 90% aller Sicherheitsprobleme gehen auf Konfigurationsfehler, d.h. der falschen Verwendung von Software zurück [8].

In dieser Arbeit werde ich elementare Begriffe der IT-Sicherheit klären. Anschließend gehe ich - wegen ihrer Relevanz zum Thema - auf gängige und alternative Authentifizierungsverfahren ein. Den Schwerpunkt meiner Betrachtungen stellen 2 Studien dar, die in Labortests den Umgang von Probanden mit sicherheitsbezogener Software untersuchen.

Im Folgenden werde ich die Terme IT-Sicherheit und Sicherheit, sowie Nutzbarkeit und Benutzerfreundlichkeit synonym verwenden.

2 Problem

Um den Problemkreis von Nutzbarkeit und Sicherheit besser begreiflich zu machen, definiere ich in diesem Abschnitt die relevanten Termini. Alternative Authentifizierungsmaßnahmen sollen im Anschluss als Beispiele benutzerfreundlicherer Sicherheit dienen und das Problem verdeutlichen. Schließlich stelle ich auf Seite 6 die Kriterien für benutzerfreundliche Sicherheitssoftware vor und gehe auf sicherheitsspezifische Eigenschaften in der Softwarentwicklung ein.

2.1 Definitionen

Um über Sicherheit und Nutzbarkeit sprechen zu können, muss eine gemeinsame Definition relevanter Begriffe vorhanden sein [7].

IT-Sicherheit umfasst die technologischen und betrieblichen Eigenschaften, die die Verfügbarkeit, Integrität und Vertraulichkeit von Computersystemen gewährleisten.

Verfügbarkeit ist die Möglichkeit eine gewünschte Information oder Ressource zu nutzen

Integrität ist die Vertrauenswürdigkeit von Daten oder Ressourcen

Vertraulichkeit ist die Geheimhaltung von Informationen

Nutzbarkeit ist das Maß für die vom Benutzer empfundene Nutzungsqualität während der Interaktion mit einem Produkt oder System. [Sie] ist die Kombination von Faktoren, die das Nutzungserlebnis mit dem Produkt oder System, einschließlich Erlernbarkeit, Effizienz, Einprägsamkeit, Fehlerrate u. -grad und subjektiver Zufriedenheit beeinflussen.

2.2 Beispiel: Authentifizierungsmaßnahmen

Authentifizierung ist eine klassische Aufgabe, die bei sicherheitsrelevanter Software zu lösen ist. Ein sehr bekanntes und gängiges Verfahren, eine behauptete Identität zu verifizieren, sind Passwörter. Jedoch leidet dieses Verfahren unter einem zentralen Problem: Der Mensch kann sich sequentiell nur 5-9 Symbole merken[9]. So simpel die Ursache ist, so zahlreich sind auch die Konsequenzen:

- es wird ein simples Passwort gewählt
- ein Passwort wird für mehrere Dienste verwendet

- Passwörter werden an für Fremde zugängliche Stellen notiert
- ...

Die hohe Anzahl an Webseiten und Diensten, die jeweils einen eigenen Satz Benutzername-Passwort benötigen, verschärfen das Problem [5]. Trotz der 2004 von Jeff Yan et al. in [9] vorgestellten Mnemonic-Methode zur Generierung leicht einprägsamer Passwörter, besteht das Passwortproblem beispielhaft für die schwierige Vereinbarkeit von Nutzbarkeit und Sicherheit fort.

Alternativen

Passwörter sind nicht die einzigen Authentifizierungsverfahren. Mit dem Ziel, Authentifizierung leichter zu machen, wird seit Jahren an zuverlässigen Fingerabdruckerkennungs-Verfahren gearbeitet. [5] stellt ein biometrisches Verfahren vor, das mittels Prüfsummenbildung einen Großteil der Fingerabdruckcharakteristika beschreibt und damit den Fingerabdrucks-Eigner authentifizieren kann. Revolutionär ist die beachtliche Fehlertoleranz, die das Verfahren auch für neue Szenarien wie dem Verbot der Abgabe von Alkohol an Minderjährige qualifizieren soll.

Allerdings machen die Möglichkeit falsch-positiver Authentifizierungen die zusätzliche Verwendung einer PIN notwendig.

Einen ganz neuen Ansatz verfolgen Ann Nosseir et al. in [6] mit ihrem Fragen-basierten Verfahren. Dabei sammelt eine „smart environnment" Informationen zu den sich in ihr befindlichen Individuen. Eine solche Umgebung kann eine mit Wärmesensoren und Lichtschranken ausgestattete Büroetage sein, die die Bewegungen der Mitarbeiter aufzeichnet. Im Falle einer Authentifizierung muss der Befragte mehrere Multiple Choice-Fragen richtig beantworten, wobei eine Frage so lauten könnte: „Wo waren Sie kurz vor 14 Uhr? - A: Am Schreibtisch, B: In der Küche". Das Verfahren geht davon aus, dass - obwohl die gefragten Informationen jeder wissen könnte - die Antwort mit der notwendigen Detailtiefe und Vollständigkeit nur die berechtigte Person wissen kann.

Laut dem Forscherteam eignet sich dieses Verfahren jedoch nur für Szenarien mit geringem Risiko, wie z.B. beim unerwünschten Zutritt von Mitarbeitertoiletten durch Dritte oder dem Schutz abgesperrter Fahrräder.

Zusammenfassung

Abschließend kann man sagen, dass Authentifizierungsmaßnahmen ein klassisches Beispiel dafür sind, dass Nutzbarkeit und Sicherheit schwer vereinbare Eigenschaften sind. Passwörter sind bei bestimmungsgerechter Anwendung ein zuverlässiges Mittel zur Gewährleistung von Sicherheit. Allerdings ist der Gebrauch von Passwörtern bei einer durchschnittlichen Authentifizierungsdauer von 7-20 Sekunden [1] zeitaufwendig. Außerdem führt

die übliche Nichteinhaltung der Empfehlung zum Umgang mit Passwörtern dazu, dass das gesamte Verfahren kompromittiert wird. In einem Vergleich von 14 verbreiteten Authentifizierungsverfahren, gehört die effektive Sicherheit von Passwörtern zu den 3 schlechtesten [1].

2.3 Kriterien für benutzerfreundliche Sicherheitssoftware

Mehr als 90% der Sicherheitsprobleme gehen auf Bedienungsfehler zurück [8]. In Zeiten, in denen immer mehr Anwender Computernetzwerke für private und geschäftliche Aktionen nutzen [2], ist das Risikopotential besonders hoch. Die falsche Bedienung von sicherheitsrelevanter Software beginnt bereits, wenn der Anwender vergisst, auf „Verschlüsseln" zu klicken, und damit kritische Information im Klartext durch das Internet schickt - schlimmstenfalls in dem Glauben, verschlüsselt zu haben.

Um den Grad der Benutzerfreundlichkeit in Sicherheitssoftware zu ermitteln, nennt [8] folgende Kriterien:

1. Dem Benutzer werden verlässlich die zu erfüllenden Sicherheits-Aufgaben klar gemacht.
2. Der Benutzer können herausfinden, wie man diese Aufgaben erfolgreich erfüllt.
3. Der Benutzer macht keine gefährlichen Fehler.
4. Fühlen sich so wohl mit dem Interface, dass sie mit der Software weiterarbeiten würden.

Punkt 2 stellt einen sehr häufig vernachlässigten Punkt dar: Bestimmte Technologien, wie das Public Key-Verfahren, sind dem Anwender völlig unbekannt. Es ist die Aufgabe des Programms, dem Anwender eine Vorstellungsmodel zu vermitteln, die den fehlerfreien Gebrauch der Software ermöglicht.

2.4 Sicherheitsspezifische Eigenschaften

IT-Sicherheit besitzt Eigenschaften, die sich nur schwer durch den Benutzerschnittstellenentwurf adressieren lassen. [8] schlägt eine unvollständige Liste von Sicherheitseigenschaften vor, die trotz vorhandener Schwierigkeiten, von Design-Strategien berücksichtigt werden müssen:

1. Unmotivierte Benutzereigenschaft
   Sie sagt aus, dass der Benutzer auf ein primäres Ziel fokussiert ist und sich folglich nicht mit Sicherheit auseinandersetzen möchte. Der Anwender tendiert dazu von funktionierender Sicherheit auszugehen und sucht nicht nach unauffälligen oder schwer zugänglichen Sicherheits-Konfigurationsmöglichkeiten.

2. Abstraktionseigenschaft
   Programmierer gehen davon aus, dass mögliche Sicherheitskonfigurationen (z.B. Zugriffslisten, Zertifikatsverwaltung) korrekt erfolgen. Üblicherweise sind derartige Konfigurationen sehr technischer Natur und nicht intuitiv zu bedienen.

3. Feedback-Eigenschaft
   Der Status einer Sicherheitseinstellung hängt von mehreren Faktoren ab und ist damit komplex. Außerdem ist es schwierig zu erkennen, was der Anwender für eine Sicherheitskonfiguration möchte. Die gewünschte, korrekte Sicherheitseinstellung zu erkennen, gestaltet sich damit als schwierig und erschwert es, den Anwender vor einer Fehlkonfiguration zu warnen.

4. Scheunentor-Eigenschaft
   Diese Bezeichnung geht auf folgendes Sinnbild zurück: Ist das Pferd erst einmal geflohen, macht es keinen Sinn mehr, das Scheunentor zu schließen.
   Dahinter steht das Phänomen, dass Fehlkonfigurationen irreversibel sein können. So kann beispielsweise eine versehentlich unverschlüsselte Mail nicht mehr zurückgenommen werden. Das weit verbreitete try-and-error-Konzept greift nicht und führt zu großem Schaden.

5. Schwächstes Glied-Eigenschaft
   Anwender lösen Probleme in aller Regel nur punktuell. Die typische Strategie, ein Programm nach relevanten Konfigurationsoptionen abzusuchen und nach der erstbesten Konfigurationsänderung das Problem als erledigt anzusehen, erweist sich als heimtückisch. Sicherheitsaspekte müssen ganzheitlich betrachtet werden; das Auslassen eines Gliedes lässt die Gesamtsicherheit auf das entsprechende Niveau zurückfallen.

Die Fülle und Folgenschwere der genannten Eigenschaften lässt vermuten, dass vorhandene Softwareentwicklungsverfahren die genannten Punkte nicht berücksicht. [3, 8] fordern ganzheitliche interdisziplinäre Ansätze und die Definition neuer Designziele und effektiver Evaluationsmethoden.

Bruschi et al. bemängelt in [2], dass Sicherheit nicht von Anfang an eine Rolle in der Entwicklung einer Software spielt, häufig nachgerüstet wird und damit die Design-Integrität der Software kompromittiert. Es fehlt an praktischer Methodologie in IT-Sicherheit und vorhandene Sicherheitsstandards finden nur langsam ihren Weg in Mainstream-Softwareentwicklungstechniken. Allerdings spricht [2] trotz der Schwierigkeiten von Definition und Evaluation der Sicherheitsanforderungen von einem erfolgsversprechenden neuen Lösungsansatz: Dabei sollen onthologische Ansätze in Kombination mit anderen Technologien sich dafür eignen, Systemsicherheit zu definieren, sie zu beweisen und die zu ihrer Aufrechterhaltung verbundenen Aufgaben

zu ermitteln. Dieses Vorgehen würde eine genaue Zertifizierung und Akkredtierung erlauben.

3 Studie I: PGP 5.0

Whitten und Tygar führten 1999 unter [8] eine Studie durch, die zum Ziel hatte, die von ihnen unter 2.3 und unter 2.4 vorgestellten Kriterien und Eigenschaften benutzerfreundlicher und sicherer Software zu prüfen.

Sie entschieden sich für das Programm PGP in damalig aktuellen Version 5.0. PGP dient primär dazu, Privatpersonen den verschlüsselten und signierten Versand von Mails zu ermöglichen. Als Sicherheitssoftware galt es damals als besonders benutzerfreundlich, was auch so vom hauseigenen Marketing beworben wurde.

3.1 Grundsätzliche Fragestellung

Angenommen ein Benutzer möchte auf sichere Weise Mails versenden und entscheidet sich für PGP:

1. Versteht der Benutzer, was zu tun ist?
   (z.B. Erstellung eines privaten Schlüssels, Hochladen des öffentlichen Schlüssels, etc.)

2. Wie es zu tun ist?

3. Macht er dabei gefährliche Fehler?
   (z.B. bei der Validierung eines Zertifikats)

4. Wird der Anwender so frustriert, dass er PGP nicht mehr nutzen wird.

5. Schafft der Benutzer die Schritte in vertretbarer Zeit?

3.2 Durchführung

Bei der Evaluation von PGP kamen 2 unterschiedliche Verfahren zum Einsatz:

1. Kognitiver Durchgang („Cognitive Walktrough")

2. Labortest

Bei ersterem handelt es sich um eine Nutzbarkeits-Inspektionsmethode, die in dieser Studie von den beiden Authoren durchgeführt wurde. Letzteres Verfahren wurde mit 12 E-Mail-erfahrenen durschnittlich gebildeten Probanden durchgeführt.

3.2.1 Kognitiver Durchgang

Beim kognitiven Durchgang, geht der Evaluierende die Benutzeroberfläche auf die Weise durch, wie er meint, dass es ein Standard-Benutzer tun würde. Das Verfahren ist dabei dem „Code Walktrough" nachempfunden, wobei nun das Ziel die Erkennung von möglichen Quellen für Fehlbedienungen ist. Im Vordergrund stehen die unter 2.3hochprioritativen Benutzerfreundlichkeitskriterien.

Die Analyse hat folgende Punkte ergeben:

- Visuelle Methapern
  Unter anderem verwendet PGP statt 2 Schlüssel symbolisch nur 1 Schlüssel. Das führt beim Anwender zu der Annahme, dass für die Ver- u. Entschlüsselung ein und derselbe Schlüssel notwendig ist.
  Die Signierungsfunktion wird mit einem Füller symbolisiert. Dies kann dazu führen, dass der Anwender davon ausgeht, dass bei diesem Prozess keine Schlüssel, sondern andere Mechanismen zum Einsatz kommen.

- Unterschiedliche Schlüsseltypen
  Die Versionen vor 5.0 verwendeten RSA für die Verschlüsselung und Signierung, während ab Version 5.0 der Diffie-Hellman-Algorithmus verwendet wird. Diesbezügliche Warnmeldung in Hinblick auf die Abwärtskompatibilität werden nur unzureichend erklärt; der Anwender für ein genaueres Verständnis auf das 132-Seiten starke Benutzerhandbuch verwiesen.

- Schlüssel-Server
  Damit ein Benutzer A eine verschlüsselte Nachricht an Benutzer B senden kann, benötigt A den öffentlichen Schlüssel von B. Die Einbeziehung dieses so genannten Schlüssel- bzw Key-Servers wird den Anwendern nicht richtig verdeutlicht. Während der Key-Server nur über ein Untermenü zu erreichen ist, werden lokale und serverseitige Aktionen für den Benutzer nicht unterschieden und erschweren dami sein Verständnis.
  Interaktionen mit dem Server werden nichts dokumentiert und es erfolgt eine Rückmeldung bei erfolgreicher Interaktion.

- Unumkehrbare Aktionen
  Wenn der Anwender seinen privaten Schlüssel löscht, ist es nicht mehr möglich, zugehörige Nachrichten zu entschlüsseln. Diese und weitere technische Folgen werden dem Benutzer bei einem Löschversuch nicht erläutet. Das System stellt lediglich folgende Rückfrage: „Do you really want to delete these items?".

- Konsistenz im Sprachgebrauch
  PGP verwendet die Begriffe „Verschlüsselung" und „Signierung" zunächst einheitlich, führt dann aber den Begriff „Enkodierung" ein.

Laut Whitten und Tygar führt dies beim Anwender zu der Vorstellung, es handele sich um einen 3., unabhängigen Prozess.

- Zu viele Informationen
  Die Benutzeroberfläche stellt je Schlüssel 7 Eigenschaften tabellarisch dar. Die Authoren schlagen vor, die Oberfläche stark zu minimalisieren und sich auf den Zusammenhang von öffentlichen und privaten Schlüssel zu konzentrieren, um der eigentlichen Aufgabe des Programms näher zu kommen.

3.2.2 Labortest

Ergänzend zum kognitiven Durchgang, führten Whitten und Tygar einen Labortest mit 12 Probanden durch. Die Probanden hatten Erfahrungen mit E-Mail und explizit kein Wissen zu Kryptographie.

Die Probanden sollten sich vorstellen, für die Koordination eines Wahlkampfes zuständig zu sein. Dies sollte die hohe Motivation, die sensiblen Information geheim zu halten, sicherstellen. Die immaginären Teammitglieder waren die Durchführenden des Labortests. Während des 90min Tests gab es nur elementare Hilfe, z.B. bei falsch installierter Software.

Die markantesten Ergebnisse sahen wie folgt aus:

Verschlüsselung

- 3 Anwender versandten ihre Mails unverschlüsselt; einer von ihnen merkte diesen Umstand nicht und wähnte sich in Sicherheit
- 1 Anwender verbrachte viel Zeit mit der Suche nach einem Knopf zur Aktivierung der Verschlüsselung
- 7 Benutzer verwendeten den eigenen öffentlichen Schlüssel statt denen der anderen Teammitglieder
- 3 Probanden gelang die erfolreiche Verschlüsselung

Entschlüsselung

- 1 Nutzer hielt die verschlüsselte Nachricht für den Schlüssel
- 2 Benutzern gelang die erfolgreiche Entschlüsselung

Schlüsselverwaltung

- 3 Benutzer glaubte, den privaten Schlüssel ebenfalls auf den Key-Server hochladen zu müssen
- 2 Anwender verstanden bis zum Ende des Tests nicht, dass sie die öffentlichen Schlüssel der anderen benötigen

3.3 Schlussfolgerung

Den PGP-Labortest bestanden nur 4, d.h. ein Drittel der Probanden erfolgreich, während 3 der Testteilnehmer ihre Mails sogar unverschlüsselt verschickten. Um auf die in 2.3 genannten Kriterien zurückzukommen, muss man feststellen, dass das Public Key-Verfahren nicht vollständig verstanden wurde. Es traten Missverständnisse im Umgang mit öffentlichen und privaten Schlüssel auf. Die Mehrheit der Testteilnehmer gab an, frustriert zu sein und PGP in Zukunft nicht nutzen zu werden.

Als Konsequenz geben Whitten und Tygar an, dass es notwendig ist, die Funktionalitäten von PGP auf das wichtigste zu beschränken, ohne die Integrität zu gefährden. Das dafür notwendige Verständnismodell müsse mit metaphorischen Elementen in der Benutzerschnittstelle geschaffen werden.

4 Studie II: Polaris

7 Jahre nach Veröffentlichung der PGP-Studie und in Anlehnung an selbige veröffentlichen 2006 DeWitt and Kuljis in [4] eine Studie zur Software Polaris.

Polaris hat das Ziel auf benutzerfreundliche Weise die Arbeit und Windows sicherer zu machen. Dabei werden „polarisierte" Anwendungen unter Windows in einer Sandbox nach dem „Principle of Least Authority" (POLA) ausgeführt. POLA definiert, dass eine vom Benutzer gestartete Anwendung nicht die Rechte des Aufrufers erbt, sondern nur mit minimalen Rechten ausgestattet wird. Eine Sicherheitslücke im Programm führt damit nicht dazu, dass ein Virus sich die geerbten Rechte zu nutze machen kann.

4.1 Durchführung

Die Durchführung der Studie lehnt sich stark an Whitten und Tygars PGP-Studie an. Als Probanden dienen 10 Anwender aus dem universitären Umfeld, die regelmäßig mit dem Computer arbeiten. Innerhalb des Tests mussten die Probanden 8 Aufgaben lösen, die u.A. die sichere Ausführung einer heruntergeladenen Anwendung in einer Sandbox umfassen.

Dabei ist zu betonen, dass die Probanden deutlich mehr Erfahrung im Umgang mit Computern sich brachten, als die Probanden der PGP-Studie.

4.2 Ergebnisse

Der Umgang mit Polaris viel den Anwendern schwer. Keinem der Probanden gelang die korrekte erste Benutzung einer „polarisierten" Anwendung ohne detaillierte Anleitung. Jedoch offenbart die Studie auch ein unerwartet fahrlässiges Verhalten seitens der Anwender:

- Der Prozess der „Polarisierung", d.h. Ausführung einer Anwendung in einer Sandbox, wurde als so schwer empfunden, dass sich die Anwender mit dem Verfahren nicht weiter auseinander gesetzt haben. Stattdessen führten sie die heruntergeladene Software willentlich auf unsichere Art aus.
- Die Erwartungen an die Benutzerfreundlichkeit war hoch.
- Trotz des Wissens um der Risiken, öffneten Anwender das Onlinebanking in der gleichen Sandbox, die für die heruntergeladene Software verwendet wurde.
- Obwohl die Anwender die Sicherheit und Vertrauenswürdigkeit von Hyperlinks einschätzen konnten...

- ... hatte das keine Konsequenzen für ihr Handeln.
- ... waren die Konsequenzen 2 aufgeklärten Probanden egal.
- ... begründeten sie ihre Fahrlässigkeit, damit, dass sie ohnehin niemand für ihre Daten interessieren würde.

4.3 Schlussfolgerung

Die extremen Ergebnisse zeigen, dass ein durschnittlicher Anwender sich für Sicherheit nur so lange interessiert, solange sie nicht kompliziert ist. Der Anwender präferiert den Weg des geringsten Widerstandes und ist nicht sensibel für die Wichtigkeit seiner Daten.

DeWitt and Kuljis schlagen aus diesem Grund folgende Punkte für die Entwicklung zukünftiger Software vor:

- Entfernung von Fragen zur Sicherheit an den Stellen, wo die Software die Frage besser als der Anwender beantworten kann
- Sicherstellung, dass der sicherste auch gleichzeitig der schnellste Weg ist, ein Problem zu lösen
- Sensibilisierung für den Wert der eigenen Daten
- Vermeidung von Warndialogen, da der Nutzen zweifelhaft ist

5 Zusammenfassung

Die Entwicklung benutzerfreundlicher Sicherheitssoftware ist tatsächlich ein schwieriges Unterfangen. Gängige Softwarentwicklungstechniken berücksichtigen nicht die Eigenschaften sicherer Software in Bezug auf Nutzbarkeit. Das heißt, dass neue Designrichtlinien und Implementierungsverfahren notwendig sind.

Churchill et al. stellen in [3] allerdings heraus, dass reine technische Aspekte bei der Adressierung des Problems nicht ausreichen, sondern interdisziplinäre Forschungsarbeit notwendig ist.

Der korrekte Umgang mit sicherheitsrelevanter Software ist kein rein passiver Prozess für den Anwender. Es ist notwendig, dass Programme dem Anwender Hilfestellung beim Verstehen von Sicherheitsfragen geben und der Awender diese aktiv nutzt. Im Unternehmensumfeld empfehlen sich sogar spezielle Schulungen[8].

Die Ergebnisse der beiden Studien haben eine große Gemeinsamkeit: Sie sprechen sich für die Minimalisierung der Benutzerschnittstelle auf die Kernaufgaben aus. Hingegen ist der Appel der PGP-Studie, den Benutzer genauer über das zu informieren, was er tut - während die Polaris-Studie aus Pragmatismus sich sogar gegen Warndialoge ausspricht. Dieser Widerspruch zwischen Informationsreichtum und -Armut in Benutzerprogrammen wird in weiteren Studien noch genauer zu untersuchen sein.

Literatur

[1] BRAZ, C., AND ROBERT, J.-M. Security and usability: the case of the user authentication methods. In *IHM '06: Proceedings of the 18th International Conferenceof the Association Francophone d'Interaction Homme-Machine* (New York, NY, USA, 2006), ACM, pp. 199–203.

[2] BRUSCHI, D., WIN, B. D., AND MONGA, M. Introduction to software engineering for secure systems: Sess06 – secure by design. In *SESS '06: Proceedings of the 2006 international workshop on Software engineering for secure systems* (New York, NY, USA, 2006), ACM, pp. 1–2.

[3] CHURCHILL, E., NELSON, L., AND SMETTERS, D. K. Useful computer security. *IEEE Internet Computing 12*, 3 (2008), 10–12.

[4] DEWITT, A. J., AND KULJIS, J. Aligning usability and security: a usability study of polaris. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* (New York, NY, USA, 2006), ACM, pp. 1–7.

[5] JAKUBOWSKI, M. H., AND VENKATESAN, R. Randomized radon transforms for biometric authentication via fingerprint hashing. In *DRM '07: Proceedings of the 2007 ACM workshop on Digital Rights Management* (New York, NY, USA, 2007), ACM, pp. 90–94.

[6] NOSSEIR, A., CONNOR, R., REVIE, C., AND TERZIS, S. Question-based authentication using context data. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction* (New York, NY, USA, 2006), ACM, pp. 429–432.

[7] ROZINOV, K. Are usability and security two opposite directions in computer systems?

[8] WHITTEN, A., AND TYGAR, J. D. Why johnny can't encrypt: a usability evaluation of pgp 5.0. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium* (Berkeley, CA, USA, 1999), USENIX Association, pp. 14–14.

[9] YAN, J., BLACKWELL, A., ANDERSON, R., AND GRANT, A. Password memorability and security: Empirical results. *IEEE Security and Privacy 2*, 5 (2004), 25–31.