Veranstaltung „Informationssicherheit“

HS Ludwigshafen, Prof. Dr. H. Röckle

1 Einordnung

Im Curriculum des Bachelorstudiengangs Wirtschaftsinformatik an der Hochschule Ludwigshafen stellt die Veranstaltung „Informationssicherheit“ einerseits mit dem Themenbereich „Web Applikationssicherheit“ den Abschluss des Zyklus Programmierung / Anwendungssysteme dar. Andererseits bietet die Veranstaltung eine allgemeine Einführung zur Informationssicherheit und dem Informationssicherheitsmanagement.

Beide Themenbereiche werden in der Veranstaltung „Informationssicherheitsmanagement (ISM)“ des Masterstudiengangs Wirtschaftsinformatik mit Schwerpunkt Management & Consulting noch vertieft.

1.1 Lernziele

Nach dem erfolgreichen Besuch der Veranstaltung sollen die Studierenden

- die Begriffe Informationssicherheit, Datenschutz und Informationssicherheitsmanagement
  - verstehen sowie
  - die elementaren Vorgehensweisen des Informationssicherheitsmanagements kennen und
  - in einfachen Fällen anwenden können.
- die wichtigsten Sicherheitsschwachstellen der Webtechnologie
  - kennen und
  - den Schutzbedarf eigener Webanwendungen ermitteln können sowie
  - elementare Maßnahmen zur Sicherheit eigener Webanwendungen implementieren können,

1.2 Literatur

Neben einigen wenigen Büchern finden sich die meisten Quellen zur Informationssicherheit im Internet, z.B. auf Webseiten von für Informationssicherheit zuständigen Gremien:

- Heinrich Kersten, Jürgen Reuter, Klaus-Werner Schröder: IT-Sicherheitsmanagement nach ISO 27001 und Grundschutz, Springer Vieweg, 4. Auflage (2013)
- Michal Zalewski: Tangled Web - Der Security-Leitfaden für Webentwickler, dpunkt.verlag (2013)
- Klaus-Rainer Müller: IT-Sicherheit mit System, Vieweg + Teubner (2011)
- Hans-Peter Königs: IT Risiko-Management mit System, Vieweg + Teubner (2009)
- Gilbert Brands: IT-Sicherheitsmanagement, Springer (2005)
- Website des Bundesamts für Sicherheit in der Informationstechnik, https://www.bsi.de
- Website des Open Web Application Security Project, https://www.owasp.org
- OWASP Appsec Tutorial Series, http://www.youtube.com/user/AppsecTutorialSeries
- Website des Web Application Security Consortium, http://www.webappsec.org
- Sicherheitsseiten des Heise-Verlags, http://www.heise.de/security/
- Website des Information Security Forum (ISF), https://www.securityforum.org/

Auch Hersteller von Sicherheitssoftware (Verschlüsselung, Firewalls, Virenschutz, Authentisierung, Zugriffsschutz, Protokollierung, Intrusion Detection, Incident Management, Backup, etc.) stellen häufig interessante Sicherheitsinformationen auf ihre Websites, die aber naturgemäß nicht ganz uneigennützig sind und deshalb hier nicht zitiert werden.

1.3 Inhalt

Informationssicherheit ist ein außerordentlich umfangreiches Thema. Vor diesem Hintergrund ist die Festlegung des geeigneten Inhalts für eine nur 2 SWS umfassende Veranstaltung sehr subjektiv. Im Rahmen der in 1 genannten Einordnung umfasst die Veranstaltung die folgenden Themen:

- Informationssicherheit und Datenschutz
- Informationssicherheitsmanagement
- Applikatorische Sicherheit
- Web Application Security

1.4 Benötigte Software - Eclipse IDE for Java EE Developers

Wie in der vorausgehenden Veranstaltung „Webanwendungen“ wird auch in dieser Veranstaltung die folgende Software verwendet:

- Eclipse in der Version Eclipse IDE for Java EE Developers
- Apache TomCat
- WireShark

Weitere Informationen zu dieser Software finden Sie in dem Skript zur Veranstaltung Webanwendungen.

Weitere Software und deren Beschaffungsquellen werden ggfs. in der Veranstaltung genannt.

2 Grundlagen

2.1 Bedeutung der Informationssicherheit

Die Sicherheit von Informationen ist schon immer ein wichtiger Erfolgsfaktor speziell von Unternehmen, als Beispiel könnte das Rezept von Coca Cola seit 1887 dienen.

Übung 1: Überlegen Sie sich noch weitere Beispiele ohne IT

Mit der Verbreitung der elektronischen Informationsverarbeitung (IT) änderte sich aber Verschiedenes:

- Es werden heutzutage mehr Informationen gespeichert und verarbeitet als je zuvor
- Informationen werden an mehreren Orten gespeichert und verarbeitet (Hauptspeicher, lokale und zentrale Datenträger, Netzwerkverbindungen, Backups, Ausdrucke)
- Die Vervielfältigung von Informationen erfordert keine nennenswerte Zeit mehr
- Auf Daten kann oft zugegriffen werden, ohne dass dies bemerkt wird

Übung 2: Überlegen Sie sich noch weitere Veränderungen

Aus diesen Gründen ist die Bedeutung der Informationssicherheit mit der Verbreitung der elektronischen Informationsverarbeitung noch einmal stark gestiegen.

2.2 Grundwerte und Definiton der Informationssicherheit

Informationssicherheit bezeichnet den Zustand, in dem die so genannten Grundwerte Vertraulichkeit, Integrität und Verfügbarkeit für alle Informationen erfüllt sind. Dabei gilt:

- **Vertraulichkeit** bedeutet, dass nur diejenigen Personen und Systeme auf Informationen zugreifen können, die dazu berechtigt sind.
- **Integrität** bedeutet einerseits die Korrektheit von Informationen, andererseits, dass nur diejenigen Personen und Systeme Informationen ändern können, die dazu berechtigt sind.
- **Verfügbarkeit** bedeutet, dass Informationen den Personen und Systemen, die dazu berechtigt sind, zur gewünschten Zeit zur Verfügung stehen.

Unter „Information“ versteht man in diesem Zusammenhang nicht nur elektronisch gespeicherte oder verarbeitete Informationen, vgl. 2.7.1.

2.3 Wesen der Informationssicherheit

Die oben genannte Definition der Informationssicherheit vermittelt den Eindruck, dass diese entweder gegeben ist oder nicht gegeben ist. Dieser Eindruck ist aber falsch, denn **100% ige Sicherheit gibt es nicht**. Vor diesem Hintergrund könnten wir die o.g. Definition als „absolute Informationssicherheit“ bezeichnen, die aber in Wirklichkeit nur ein Idealziel bzw. ein Gedankenspiel sein kann.

In der Realität ist Informationssicherheit vielmehr eine Eigenschaft, die niemals komplett gegeben ist, d.h. für bestimmte Informationen sind Vertraulichkeit, Integrität und Verfügbarkeit nur mehr oder weniger vorhanden. Außerdem ist in der Regel für alle Arten von Informationen und für jedes informationsverarbeitende System der Stand der Informationssicherheit ein anderer.

2.4 Gefährdung der Informationssicherheit

2.4.1 Beispiele

Die Verletzung der Informationssicherheit bedeutet die Verletzung mindestens eines Grundwerts:

1. Verletzung der Korrektheit: Manipulation von börsenrelevanten Informationen eines Unternehmens
2. Verletzung der Vertraulichkeit: Vergessen eines USB-Sticks mit Kundendaten.
3. Verletzung der Verfügbarkeit: Sabotage oder DNS-Angriff
4. Verletzung der Verfügbarkeit: Ausfall von Hardware, für die kein Backup oder kein Ersatzteil vorhanden ist

2.4.2 Absichtlich und unabsichtlich

Sicherheit kann grundsätzlich absichtlich oder zufällig verletzt werden. Z.B. gehört ein Bankraub zu den absichtlichen Sicherheitsverletzungen, ein Verkehrsunfall zu den unabsichtlichen. Dasselbe gilt auch für die Informationssicherheit. Speziell gehören die Beispiele 1. und 3. zu den absichtlichen und die Beispiele 2. und 4. zu den unabsichtlichen Verletzungen der Sicherheit.

Übung 3: Überlegen Sie sich zu jedem Grundwert eine weitere absichtliche und eine unabsichtliche Verletzung.

2.4.3 Technisch und personell

Sicherheitsgefährdungen können grundsätzlich aus technischen oder aus personellen Schwachstellen oder aus einer Mischung aus beidem entstehen, z.B. besteht die technische Komponente in Beispiel 2 darin, dass USB-Sticks zur Datenspeicherung genutzt werden und die personelle Komponente darin, dass dieser vergessen wird.

Übung 4: Benennen Sie für die o.g. Beispiele die technischen und die personellen Einflüsse.

2.4.4 Allgemein

Wie sicher bzw. wie gefährdet bestimmte Informationen sind, hängt davon ab,

- in welchen Systemen diese vorliegen,
- welche Möglichkeiten für Angriffe oder unabsichtliche sowie technische oder personelle Sicherheitsverletzungen es für diese gibt und
- welche Maßnahmen zur Erhaltung der Sicherheit ergriffen wurden.

Zu bedenken ist dabei:

- die unterschiedlichen Grundwerte können in völlig unterschiedlichem Umfang erfüllt sein,
- je weiter die Informationen verbreitet sind, umso mehr Angriffsmöglichkeiten bzw. Unsicherheiten kann es potenziell geben,

2.4.5 Angreifer

Ein Angreifer ist jeder, der absichtlich versucht, die Sicherheit von Informationen zu verletzen, sei es durch

- unberechtigtes Lesen von Informationen (Grundwert Vertraulichkeit),
- unberechtigtes Verändern von Informationen (Grundwert Integrität) oder
- Verhindern des Zugriffs auf Informationen (Grundwert Verfügbarkeit).

Je größer der Nutzen eines Angreifers ist, z.B. durch Lesen geheimer Kundendateien oder Forschungsergebnisse, umso eher muss von einem Angriff ausgegangen werden. Im Extremfall muss davon ausgegangen werden, dass ein Angreifer große Geldmittel oder kriminelle Energie für einen Angriff einsetzt, z.B. Bestechung, Sabotage, Infiltration von Geschäftspartnern.

2.4.6 Gefährdungsvektor

Als Gefährdungsvektor bezeichnet man das Tupel bestehend aus

- der gefährdeten Information bzw. Informationsart,

- dem Ort bzw. dem System, wo die Information gefährdet ist,
- der Zugriffsart auf die Information (lesend, schreibend, löschend)
- dem oder den potenziellen Angreifern

Da es sich dabei um 4 Dimensionen handelt, ergibt sich in einem Unternehmen eine riesige Zahl potenzieller Gefährdungsvektoren, so dass dieser Begriff nur theoretisch nützlich ist.

2.5 Sicherheitsmaßnahmen

Das Gegenstück zu Gefährdungen der Informationssicherheit sind Sicherheitsmaßnahmen zur Steigerung der Informationssicherheit. In der Regel kann zu jedem Gefährdungsvektor mit einem bestimmten Aufwand eine passende Sicherheitsmaßnahme konzipiert werden. Obwohl manche Maßnahmen auch mehrere Gefährdungsvektoren auf einmal adressieren können, ergibt sich damit auch eine sehr große Zahl möglicher Sicherheitsmaßnahmen.

Man unterscheidet zwischen technischen, organisatorischen und personellen Sicherheitsmaßnahmen, z.B.

- Typische technische Informationssicherheitsmaßnmahmen sind Firewalls, Virenschutz, Verschlüsselung.
- Zu den organisatorischen Maßnahmen gehören die Durchführung von Risikoanalysen oder die Benennung eines Sicherheitsbeauftragten.
- Zu den personellen Maßnahmen gehören Benutzerschulungen oder Richtlinien zur sicheren Benutzung von IT.

Meistens entsteht bei der Einführung einer Sicherheitsmaßnahme nicht nur Einführungsaufwand sondern auch laufende Kosten in Form von Administrationsaufwänden, Lizenzkosten oder Einschränkungen bei der Verwendung des zu schützenden Systems.

2.6 Angemessenheit

2.6.1 Sicherheitsbedarf

In einem Unternehmen sind nicht alle Informationen gleich sicherheitsbedürftig. Für jedes Feld der folgenden Matrix wird es Beispiele geben:

Bedarf	**Vertraulichkeit**	**Integrität**	**Verfügbarkeit**
niedrig	Speiseplan der Kantine Grundbuch	Speiseplan	Speiseplan
mittel	Kontonummer, BLZ	Fahrplandaten	OLAT
hoch	Alle Patientenakten	Steuerdaten für Kraftwerke Grundbuch	Telefonanlage Call Center Internetanschluss Online-… (-handel/-bank)

Zu den Informationen, für die nur ein geringer Bedarf an Vertraulichkeit besteht, gehört zum Beispiel der Speiseplan der Kantinen und zu den Informationen, für die ein großer Bedarf an Korrektheit (als Teil der Integrität) besteht, gehören die Steuerdaten für Kraftwerke, weil Fehler darin zu immensen Schäden oder Störungen führen können. Oft korreliert der eigene Bedarf an Informationssicherheit mit dem Nutzen eines Angreifers, z.B. in Bezug auf geheime Forschungsergebnisse.

Übung 5: Füllen Sie jedes Feld der dargestellten Matrix mit jeweils mindestens einer Informationsart

2.6.2 Beurteilung von Sicherheitsmaßnahmen

Eine Sicherheitsmaßnahme ist angemessen, wenn der Nutzen den Aufwand überwiegt. Aufgrund des oben dargestellten Aufwands für Sicherheitsmaßnahmen bedeutet das, dass niemals alle Gefährdungsvektoren adressiert werden können. Dies bedeutet wiederum, dass 100% Sicherheit nicht nur nicht möglich, sondern auch unwirtschaftlich wäre.

Die zentrale Aufgabe für Informationssicherheitsbeauftragte in einem Unternehmen besteht also darin, die angemessenen Sicherheitsmaßnahmen zu ermitteln und umzusetzen. Leider ist der Nutzen einer Sicherheitsmaßnahme, also die erhöhte Informationssicherheit, in der Regel nicht präzise monetär darstellbar, so dass diese Aufgabe durch große Unschärfe erschwert wird. Diese Problematik wird durch das Informationssicherheitsmanagement adressiert, vgl. Kap. 5.2.

2.7 Abgrenzung weiterer Grundbegriffe

2.7.1 Security und Safety

In der englischen Sprache wird außerdem zwischen Safety und Security unterschieden. Für die Übertragung ins deutsche gibt es verschiedene Interpretationen:

- Orientiert an der Umgebung: Security („Angriffssicherheit“) ist der Schutz des Systems vor der Umgebung, Safety („Betriebssicherheit“, bzw. „Ablauf- und Ausfallsicherheit“) ist der Schutz der Umgebung vor dem System. Diese Definition steht auch in der Wikipedia, hier werden allerdings Sicherheitsbedrohungen von innen (Fehler, Innentäter) nicht berücksichtigt.
- Orientiert an der Absicht: Security ist der Schutz vor gezielten Angriffen, Safety ist die Abwehr weiterer Gefahren und Ereignisse.

Mitunter wird Safety als Oberbegriff betrachtet, der auch Security einschließt, weil auch absichtliche Angriffe den sicheren Betrieb stören. Diese Sicht ist aber eher ungebräuchlich.

Zusammengefasst: Aufgrund der uneinheitlichen Interpretation lohnt es sich eher nicht, die Begriffe Safety und Security im deutschen zu verwenden.

2.7.2 IT-Sicherheit

Der Begriff „IT-Sicherheit“ (englisch IT-Security) betont die technische Komponente der Informationssicherheit in Abgrenzung zu den bereits seit Jahrhunderten bestehenden Anforderungen des Schutzes von Informationen auf Papier o.ä.

Da aber nahezu alle Informationen heutzutage in irgendeiner Form auch elektronisch gespeichert werden, ist diese Abgrenzung im Prinzip obsolet, so dass IT-Sicherheit und Informationssicherheit im Wesentlichen synonym sind.

In diesem Skript wird der Begriff Informationssicherheit verwendet, um zu verdeutlichen, dass die Informationen auch schon geschützt werden müssen, bevor sie Eingang in eine IT-Anlage finden bzw. nachdem sie ausgedruckt wurden.

2.7.3 IuK-Sicherheit

Der Begriff der „Informations- und Kommunikationstechnologie“ betont, dass die Sicherheit der Telekommunikation ausdrücklich mit berücksichtigt werden soll. Da aber die Vorgehensweisen im reinen IT-Bereich und im Kommunikationsbereich im Wesentlichen dieselben sind, ist diese Trennung auch eher theoretischer Natur.

Innerhalb eines Unternehmens steht es natürlich dem Management offen, die Sicherheitsaufgaben für Kommunikationssicherheit separat zuzuweisen.

2.7.4 Datensicherheit

Der Begriff „Datensicherheit“ (data security) betont, dass sich die Sicherheitsanforderungen primär auf die zu verarbeitenden Daten beziehen, was den Begriff für manche Personen evtl. besser verständlich macht. Dabei wird allerdings weniger deutlich, dass die Daten nur elektronische Entsprechungen von Informationen sind, die geschützt werden müssen und zwar innerhalb und außerhalb der IT-Anlagen. Außerdem verbirgt der Begriff, dass zur Sicherheit, speziell zur Verfügbarkeit, der Daten auch die Sicherheit der datenverarbeitenden Anlagen gehört.

2.7.5 Daten sichern

Die Bezeichnung „Daten sichern“ fällt an dieser Stelle etwas aus dem Rahmen, wird aber zur Abgrenzung aufgeführt: Hierunter versteht man nicht den Oberbegriff der Herstellung allgemeiner

Datensicherheit, sondern eine spezielle Sicherheitsmaßnahme, nämlich das Anfertigen einer Sicherheitskopie von Daten.

Übrigens erhöht das Sichern von Daten die Verfügbarkeit, birgt aber gleichzeitig ein Risiko des Verlusts von Vertraulichkeit, da eine weitere Kopie von Daten angefertigt wird, die, falls sie nicht selbst geschützt wird, wiederum einem Angreifer in die Hände fallen könnte.

2.7.6 Daten schützen

Die Bezeichnung „Daten schützen“ bezieht sich wieder auf die allgemeine Aufgabe, Daten bzw. Informationen zu schützen. Dies darf aber nicht verwechselt werden mit dem Begriff des Datenschutzes, den wir in Kap. 4 detaillieren.

3 Kryptografie

3.1 Einordnung

Als Grundlage für die folgenden Kapitel der Informationssicherheit möchten wir einen Überblick über die Kryptografie geben. Die Darstellung folgt sehr knapp und in groben Zügen dem Buch „Kryptografie“ von Klaus Schmeh.[1]

Kryptografie (von kryptos = geheim und graphein = schreiben) ist die Lehre der Verschlüsselung von Daten.

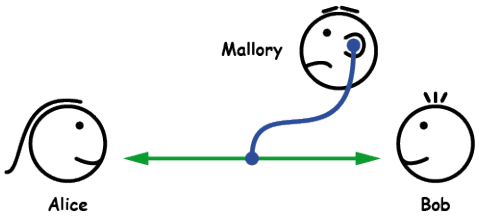
Zur Verschlüsselung gehört üblicherweise auch die Entschlüsselung (Kryptoanalyse), speziell durch unbefugte Personen, also in Form des „Codeknackens“. Kryptografie und Kryptoanalyse bilden zusammen die Kryptologie, es ist aber auch üblich, die Kryptoanalyse ebenfalls zur Kryptografie zur zählen und damit „Kryptografie“ und „Kryptologie“ gleichzusetzen.

Die in diesem Kapitel vorgestellten Ver- und Entschlüsselungsverfahren lassen sich mit Hilfe der Software „CrypTool“[2] nachvollziehen. Dem Leser / der Leserin wird dies aus Gründen des Spaß sowie des Lerneffekts wärmstens empfohlen.

3.2 Grundkonzept

3.2.1 Alice, Bob und Mallory

Üblicherweise wird in der Kryptografie von dem Modell ausgegangen, dass eine Nachricht von einem Sender (A wie Alice) an einen Empfänger (B wie Bob) gesendet wird über ein Medium, das von einem bösartigen Dritten (M wie Mallory) mitgelesen werden kann, z.B. ein abgehörtes Netzwerk.



Im weiteren Sinne sind damit auch Fälle abgebildet, wo Alice eine Datei speichert, die Bob lesen soll, auf die aber evtl. auch Mallory Zugriff hat. Vielleicht speichert Alice die Datei auch für sich selbst, dann sind Alice und Bob identisch.

3.2.2 Schlüssel

Um die Nachricht (von A an B vor M) zu schützen, wird diese mit einem „Schlüssel“ verschlüsselt. Ein Schlüssel ist ein Geheimnis, das Mallory unter gar keinen Umständen erfahren darf. In 3.3 werden wir dies anhand einiger Beispiele darstellen.

---

[1] Klaus Schmeh: Kryptografie, Verfahren, Protokolle, Infrastrukturen, 5. Auflage, dPunkt Verlag, Heidelberg (2013)

[2] http://www.cryptool.de

- Im Fall der symmetrischen Verschlüsselung muss Bob zur Entschlüsselung denselben Schlüssel kennen, den Alice zur Verschlüsselung verwendet hat.
- Im Fall der asymmetrischen Verschlüsselung unterscheiden sich die Schlüssel, die Alice bzw. Bob besitzen, hängen aber natürlich zusammen.

In beiden Fällen stellt sich die (wirklich wichtige und in keinem Fall zu unterschätzende) Aufgabe, wie Alice und Bob an den jeweils richtigen Schlüssel kommen, ohne dass Mallory diesen ebenfalls erfährt.

3.2.3 Was kann Mallory (nicht)?

In der Kryptografie wird von den folgenden Vorstellungen ausgegangen:
- Mallory kann die Verbindung vollständig und unbemerkt abhören. Die Nachricht muss deshalb so gut verschlüsselt sein, dass Mallory trotzdem nichts damit anfangen kann!
- Mallory kann die Nachricht nach Belieben verändern (manipulieren). Das Verschlüsselungsverfahren soll deshalb sicherstellen, dass Bob eine solche Manipulation immerhin bemerkt.
- Mallory kann eigene Nachrichten an Alice und Bob schicken. Das Verschlüsselungsverfahren soll deshalb möglichst sicherstellen, dass Alice und Bob erkennen können, von wem die Nachricht kommt.

Die Frage wie Mallory die Nachrichten abhören könnte, wird im Buch von Schmeh in Kapitel 3 behandelt.

3.3 Symmetrische Verschlüsselung

Ein symmetrisches Verschlüsselungsverfahren benötigt zur Durchführung einen Schlüssel, der sowohl vom Sender zur Verschlüsselung als auch vom Empfänger zur Entschlüsselung verwendet wird.

Im folgenden stellen wir einige symmetrische Verschlüsselungsverfahren dar. Weitere können im Buch von Schmeh gefunden werden (Kap. 4-10).

3.3.1 Abstrakte Formulierung

Wenn wir die Originalnachricht mit N, die verschlüsselte Nachricht mit K und den (gemeinsamen) Schlüssel mit S bezeichnen, dann ist
- das Verschlüsselungsverfahren eine Funktion V: (N,S) → K
- das Entschlüsselungsverfahren eine Funktion E: (K,S) → N

Evtl. ist V = E, in diem Fall würde zur Entschlüsselung dasselbe Verfahren verwendet wie zur Verschlüsselung.

3.3.2 Caesar-Verschlüsselung

3.3.2.1 Der Algorithmus

Vielleicht haben Sie schon in der Grundschule versucht, eine Geheimbotschaft zu schreiben. Vermutlich ist Ihnen dann der folgende Algorithmus eingefallen:
- Für jeden Buchstaben der Nachricht
  - ersetze den Buchstaben durch den im Alphabet folgenden (bzw. Z durch A)

Aus „Wirtschaftsinformatik“ wird dann z.B. „Xjsutdibgutjogpsnbujl“ (stimmt’s?)

Eine Erweiterung dieses Algorithmus ist die Caesar-Verschlüsselung:
- Wähle eine Zahl d zwischen 1 und 25 (z.B. d=3) aus.
  - Für jeden Buchstaben der Nachricht
    - ersetze den Buchstaben durch den im Alphabet um d Positionen späteren Buchstaben (wenn das Alphabet zu Ende ist, zähle wieder ab A)

Übung: Wie heißt die Nachricht, deren Caesar-Verschlüsselung mit d = 3 ergibt:
„`Iulwc iudß iulvfkh Iuövfkh`“?

Das Geheimnis (der Schlüssel) ist in diesem Fall die Zahl d.

Es handelt sich dabei übrigens um eine „monoalphabetische Substitutionschiffre“

3.3.2.2 Erweiterung

Anstatt einfach jeden Buchstaben um dieselbe Zahl von Buchstaben weiterzuschalten, kann auch eine Substitutionstabelle verwendet werden, z.B. A→G, B→U, C→M, etc.

3.3.2.3 Knacken des Algorithmus

Um einen normalen Caesar-Algorithmus zu knacken, müssen nur 25 mögliche Schlüssel durchprobiert werden. Deshalb wäre hier brute force das Mittel der Wahl.

Um die Erweiterung (Substitutionstabelle) zu knacken, wird üblicherweise eine „Häufigkeitsanalyse“ durchgeführt. Wenn der Text lang genug ist, gehorcht jede Sprache mehr oder weniger genau ihrer eigenen Verteilungsstatistik. In deutsch ist z.B. der häufigste Buchstabe meistens das ‚E’. Ist die Verteilung bekannt, kann der Text meistens schnell geknackt werden.

3.3.3 Das Schlüsselverteilungsproblem

Eine sehr einleuchtende Methode für Mallory, an die Nachricht zu kommen, liegt natürlich darin, sich den Schlüssel zu verschaffen (also das d bzw. die Substitutionstabelle) und die Nachricht dann regulär zu entschlüsseln.

Damit Bob die (Caesar-verschlüsselte) Nachricht von Alice entschlüsseln kann, braucht er den (selben) Schlüssel. Wie bekommt er diesen?

- Wenn Alice den Schlüssel unverschlüsselt über dieselbe Leitung schickt, kann Mallory diesen abhören und damit später auch die Nachricht entschlüsseln.
- Alice kann den Schlüssel aber nicht verschlüsselt über die Leitung schicken, weil sie ja noch keinen geimensamen Schlüssel mit Bob hat.
- Alice muss Bob den Schlüssel also auf einem anderen vertrauenswürdigen Weg zukommen lassen.
- Wenn es keinen solchen Weg gibt, kann symmetrische Verschlüsselung in dieser Form nicht angewendet werden.

3.3.4 Vigenère-Verschlüsselung

Um die Häufigkeitsanalyse 3.3.2.3 auszuhebeln, kann man einen längeren Schlüssel verwenden (polyalphabetische Substitutionschiffre), z.B. könnte der Schlüssel 267 bedeuten, dass

- das 1., 4., 7., … Zeichen der Nachricht um 2 Zeichen weitergeschaltet wird,
- das 2., 5., 8., … Zeichen der Nachricht um 6 Zeichen weitergeschaltet wird,
- das 3., 6., 9., … Zeichen der Nachricht um 7 Zeichen weitergeschaltet wird.

Der in der Übung verwendete Geheimtext würde dann verschlüsselt zu „`Hxpvf mtgß mtozenl Hxözenl`“

Um eine solche Botschaft zu knacken, muss man die Schlüssellänge rauskriegen. Anschließend klappt es wieder mit der Häufigkeitsanalyse. Die brute force Methode (Ausprobieren aller möglichen Schlüssel) wird allerdings aufwändiger, je länger der Schlüssel ist.

3.3.5 Vernam-Chiffre / One-Time-Pad

Bei der Vernam-Chiffre wird als Schlüssel eine Zeichenkette verwendet, die genauso lang ist wie die Nachricht selbst. Wenn der Schlüssel außerdem zufällig ist, reden wir vom „One-Time-Pad“. Dieses Verfahren ist nicht zu knacken.

Allerdings ist der sichere Austausch von sehr langen zufälligen Schlüsseln nicht praktikabel. Die Erzeugung solcher Schlüssel ist allerdings möglich und wir besprechen dies kurz in Kap. 3.8.

3.3.6 Permutationschiffren

Zur Verschlüsselung einer Nachricht können auch die Reihenfolge der Buchstaben der Nachricht verändert werden (Permutation). Als Schlüssel dient dann die Information, wie diese Permutationen durchgeführt wurden, z.B.

- Aufspalten der Nachricht in Blöcke mit jeweils 5 Buchstaben
  - Für jeden Block erzeuge einen Kryptoblock in der Reihenfolge
    - (Buchst. 4, Buchst. 1, Buchst. 2, Buchst. 5, Buchst. 3)

In diesem Fall wäre der Schlüssel (4, 1, 2, 5, 3)

Je nach Schlüssellänge können Permutationschiffren sehr sicher sein.

3.3.7 Nachrichten als Bit- oder Zahlenfolge

In modernen Verschlüsselungsverfahren werden mehrere (zum Teil ziemlich komplizierte) Substitutions- und Permutationsschritte nach einer exakten Anleitung nacheinander ausgeführt, wobei der Schlüssel an einer oder mehreren Stellen verwendet wird.

Da die einzelnen Schritte in der Regel mathematisch formuliert sind, ist es sinnvoll, die Nachricht N, die verschlüsselt werden soll, als Zahlenfolge zu betrachten. Zum Beispiel können die zu verschlüsselnden Buchstaben mit ASCII codiert werden und die entsprechenden Codes als Binärcodes hintereinander geschrieben werden. Es ergibt sich eine Folge aus Nullen und Einsen, z.B. 01101011001100001101111001011 0…

Wenn jeweils drei dieser Bits (0 oder 1) zusammengefasst werden, spricht man von Oktalzahlen (8er-System), z.B. 011 010 001 011 000 011 011 110 010 110 … Mit der Ersetzung 0 = 000, 1 = 001, 2 = 010, 3 = 011, 4 = 100, 5 = 101, 6 = 110, 7 = 111 ergibt sich die Oktalzahl 3213033626.

Wenn stattdessen jeweils 4 Bits zusammengefasst werden, spricht man von Hexadezimalzahlen (16er-System), z.B. 0110 1000 1011 0000 1101 1110 0101 1000… Mit 0 = 0000, 1 = 0001, …, 9 = 1001, A = 1010, B = 1011, C = 1100, D = 1101, E = 1110, F = 1111 ergibt sich 68B0DE98. Zur besseren Lesbarkeit werden diese paarweise dargestellt: 68 B0 DE 98.

3.3.8 Blockweise Verarbeitung

Die Substitutionen und Permutationen werden in der Regel nicht auf einzelne Bits angewendet, sondern auf größere Blöcke, z.B. 64 Bit, 256 Bit oder 1024 Bit. Dies entspricht immerhin 8 bzw. 32 bzw. 128 Hexadezimalzeichen pro Block.

3.3.9 Verknüpfungen

Zusätzlich zu Substitutionen und Permutationen können Binärblöcke miteinander verknüpft werden. Dies umfasst z.B. die binäre Addition, Subtraktion oder die XOR-Verknüpfung, also die Binäroperation, für die gilt (0,0) → 0, (0,1) → 1, (1,0) → 1, (1,1) → 0

3.3.10 Effizienz

Die genannten Substitutionen, Permutationen und Binäroperationen sind von Computern sehr schnell auszuführen. Da symmetrische Verschlüsselungsverfahren (fast) ausschließlich aus solchen Schritten bestehen, sind diese in der Regel ebenfalls sehr schnell.

3.3.11 DES-Algorithmus

Der DES-Algorithmus wurde Anfang der siebziger Jahre entwickelt. Er verwendet einen 56-Bit-Schlüssel und wendet Substitutionen, Permutationen und XOR-Verknüpfungen auf 64-Bit-Blöcke an. Er konnte bisher nicht anders geknackt werden als mit brute force. Seit Anfang der 2000er Jahre sind Computer allerdings schnell genug, um tatsächlich alle möglichen 56 Bit Schlüssel in kurzer Zeit durchzuprobieren. Deshalb kann der DES-Algorithmus nicht mehr als sicher gelten.

Näheres zum DES-Algorithmus finden Sie im Buch von Schmeh in Kap. 6.

3.3.12 AES-Algorithmus

Der AES-Algorithmus (nach seinen Erfindern Rijmen und Daeman auch als Rijndael bezeichnet) ist als Sieger aus einem großen Wettbewerb zur Festlegung des Nachfolgers von DES als Standardverfahren zur symmetrischen Verschlüsselung hervorgegangen.

AES arbeitet mit einer Blocklänge von 128 Bit und einer Schlüssellänge von entweder 128 Bit, 192 Bit oder 256 Bit. Er verwendet Substitutionen, Permutationen und Verknüpfungen. Intern ist er ziemlich kompliziert, für uns reicht aber, dass er schnell und sicher ist und mit genügend langen Schlüsseln arbeitet, um nicht brute force geknackt zu werden.

3.3.13 Weitere symmetrische Verfahren

Weitere symmetrische Verschlüsselungsverfahren finden Sie im Buch von Schmeh in Kap. 7.4, z.B. RC2, RC5, Blowfish, IDEA.

3.4 Asymmetrische Verschlüsselung

3.4.1 Grundlagen

Unter asymmetrischer Verschlüsselung versteht man Verfahren, bei denen zur Verschlüsselung und Entschlüsselung unterschiedliche Schlüssel verwendet werden. Üblicherweise besitzten Sender und Empfänger jeweils einen privaten Schlüssel (private key) und einen öffentlichen Schlüssel (public key). Dabei bleibt der private Schlüssel immer nur geheim bei seinem Eigentümer, während der öffentliche Schlüssel auch allen Kommunikationspartnern bereitgestellt werden muss.

Da der private Schlüssel niemals an den Kommunikationspartner übertragen wird, ist damit das Problem des geheimen Schlüsselaustausches gelöst. Umso wichtiger (und schwieriger) wird es allerdings, die Korrektheit des öffentlichen Schlüssels bei allen Kommunikationspartnern sicherzustellen.

Da den asymmetrischen Verfahren außerdem komplizierte mathematische Berechnungen zugrundeliegen, sind diese deutlich langsamer als symmetrische Verfahren.

3.4.2 Verschlüsselung und Entschlüsselung

Wenn Bob eine Nachricht msg in eine Geheimnachricht cryp verwandeln möchte, so dass nur Alice diese entschlüsseln kann, benötigt er Alices öffentlichen Schlüssel pubA und ein Verfahren V. Dann berechnet er

- cryp = V(msg, pubA)

Da asymmetrische Verfahren V in der Regel sehr mathematisch sind, ist die Nachricht msg als Zahl zu betrachten. Wie im symmetrischen Fall müssen wir also den Nachrichtentext codieren und dann in Blöcke zerlegen, wobei jeder Block als eine (Binär-)zahl betrachtet werden kann.

Zur Entschlüsselung benötigt Alice ihren privaten Schlüssel privA und berechnet

- msg = V(cryp, privA)

3.4.3 Mathematik 1: Das RSA-Verfahren

Der Klassiker der asymmetrischen Verschlüsselung ist das RSA-Verfahren (nach den Erfindern Rivest, Shamir und Adleman).

3.4.3.1 Erzeugung eines Schlüsselpaares

Sowohl Alice als auch Bob erzeugen sich jeweils ein Schlüsselpaar, also je einen privaten und öffentlichen Schlüssel. Diese bezeichnen wir mit privA, pubA, privB, pubB. Die Erzeugung funktioniert folgendermaßen (hier für Alice, für Bob geht es genauso):

- Alice wählt sich zwei große Primzahlen p und q und berechnet
  n = p*q und phi(n) = (p-1)*(q-1)
- Alice wählt eine Zahl e, die keinen gemeinsamen Teiler hat mit phi(n)

- (n,e) bilden zusammen den öffentlichen Schlüssel pubA. Dieser muss an Bob übertragen werden und es macht gar nichts, wenn Mallory diesen auch kennt.
- Alice berechnet ein d, für das gilt e*d ist um 1 größer als ein Vielfaches von phi(n), mathematisch heißt das e * d = 1 (mod phi(n))
- (n,d) ist der private Schlüssel privA von Alice, den sie absolut geheim halten muss.

Anhand der Primzahleigenschaft von p und q lässt sich mathematisch beweisen, dass es tatsächlich genau eine solche Zahl d gibt.

3.4.3.2 Verschlüsselung und Entschlüsselung

Zur (blockweisen) Verschlüsselung einer Nachricht msg verwendet Bob den öffentlichen Schlüssel pubA = (n,e) von Alice.

- Bob berechnet msg^e und zieht davon so lange n ab, bis das Ergebnis zwischen 0 und n-1 liegt. Das Ergebnis bezeichnet man als $cryp = msg^e \pmod n$

Zur Entschlüsselung von cryp verwendet Alice ihren privaten Schlüssel privA = (n,d):

- Alice berechnet $cryp^d$ und zieht davon so lange n ab, bis das Ergebnis zwischen 0 und n-1 liegt. Das Ergebnis ist wieder die originale Nachricht $msg = cryp^d \pmod n$

Zusammengefasst wird also zur Verschlüsselung und zur Entschlüsselung jeweils dasselbe Verfahren verwendet, nämlich $V(msg, key) = msg^{key} \pmod n$. Dass dies tatsächlich funktioniert, lässt sich mit einiger nicht ganz einfacher Mathematik beweisen.

3.4.3.3 Effizienz

Da die Bildung der Exponenten zur Berechnung durch Computer deutlich aufwändiger ist als Substitutionen, Permutationen und einfache Verknüpfungen, ist das RSA-Verfahren deutlich langsamer als symmetrische Verfahren.

3.4.4 Sicherheitsüberlegung

3.4.4.1 Sicht von Mallory

Auf derselben öffentlichen Leitung, auf der auch die Nachricht übertragen wird, können nun auch die öffentlichen Schlüssel übertragen werden, im Fall von Alice (n,e). Wenn Mallory die verschlüsselte Nachricht cryp mitliest, fehlt ihm das d, um cryp entschüsseln zu können. Die Sicherheit des Verfahrens hängt also davon ab, wie schwer es ist, von (n,e) auf d schließen zu können.

Das wiederum hängt davon ab, dass man phi(n) = (p-1)*(q-1) rauskriegt und das hängt davon ab, ob man rauskriegt, aus welchen Primzahlen p und q das n = p*q zusammengesetzt ist. Deshalb spricht man vom „Faktorisierungsproblem“: Wie kommt man von n auf p und q?

Offensichtlich wird dieses Problem umso schwieriger, je größer das n ist.

3.4.4.2 Man-in-the-Middle-Angriff

Wenn wir davon ausgehen, dass Mallory nicht nur alles mitlesen kann, sondern auch Nachrichten beliebig manipulieren, dann ist das größte Problem der asymmetrischen Verschlüsselung das folgende:

- Mallory erzeugt sich selbst ein Schlüsselpaar privM, pubM
- Mallory behauptet gegenüber Bob, er sei Alice und schickt ihm seine Schlüssel
- Bob verschlüsselt seine Nachricht mit Mallorys öffentlichem Schlüssel und Mallory entschlüsselt diese mit seinem privaten Schlüssel.
- Ebenso verfährt Mallory gegenüber Alice. Damit kann Mallory in beide Richtungen Nachrichten empfangen, entschlüsseln und wieder verschlüsseln (wenn auch mit dem anderen Schlüssel) weiterleiten. Ggfs. kann er die Nachrichten vor der Neuverschlüsselung auch manipulieren.

3.4.4.3 Korrektheit des öffentlichen Schlüssels

Aus dem genannten Grund muss die Korrektheit der öffentlichen Schlüssel gewährleistet werden. Dies ist ein ganz und gar nicht einfaches Problem, das die Verbreitung der Kryptografie in den letzten 20 Jahren massiv gebremst hat. Lösungsansätze sind:

- Persönlicher Austausch der öffentlichen Schlüssel (wie bei PGP), nur im kleinen Kreis praktikabel

- Aufbau einer Zertifizierungsinfrastruktur (Public Key Infrastruktur), sehr aufwändig und teuer und hat wiederum eigene Sicherheitsrisiken

3.4.5 Digitale Signatur

Unter digitaler Signatur verstehen wir Verfahren, mit denen ein Empfänger überprüfen kann, dass
- eine Nachricht während der Übertragung nicht verändert wurde und
- die Nachricht tatsächlich vom richtigen Sender stammt.

Interessanterweise können dafür dieselben Verfahren wie zur Verschlüsselung verwendet werden, nur die Rolle der beteiligten Schlüssel kehrt sich um.

3.4.5.1 Grundidee

Wir gehen davon aus, dass Alice eine Nachricht an Bob schicken möchte und erinnern uns daran, dass Alice zur Verschlüsselung Bobs öffentlichen Schlüssel pubB verwenden musste und Bob zur Entschlüsselung seinen privaten Schlüssel privB.

Wenn Alice die Nachricht nun (auch) digital signieren möchte,
- verwendet Alice zur Erzeugung der Signatur ihren privaten Schlüssel: sig = V(msg, privA)
- und Bob zur Prüfung der Signatur Alice‘s öffentlichen Schlüssel: msg = V(sig, pubA)

Beobachtungen:
- Da nur Alice ihren privaten Schlüssel privA kennt, kann auch nur Alice diese digitale Signatur anfertigen. Entsprechend ist die Geheimhaltung des privaten Schlüssels sehr wichtig.
- Zur Prüfung der Signatur ist es notwendig, dass Bob tatsächlich den korrekten öffentlichen Schlüssel von Alice kennt. Falls Mallory Bob einen falschen Schlüssel untergeschoben hat, von dem Bob denkt, dass er zu Alice gehöre, dann kann Mallory jederzeit signierte Nachrichten schicken, von denen Bob ebenfalls glaubt, dass sie von Alice kommen.

3.4.5.2 Durchführung

In der Praxis wird nicht die ganze Nachricht msg digital signiert, sondern nur ein Hashwert der Nachricht hash(msg), vgl. Kap. 3.6. Die digitale Signatur wäre also sig = V(hash(msg), privA).

Die Prüfung erfolgt, indem Bob
- einerseits selbst den Hashwert hash(msg) der Nachricht ermittelt und
- andererseits die digitale Signatur auswertet, wobei mit V(sig, pubA) ebenfalls hash(msg) herauskommen muss.

Zusammengefasst: hash(msg) = V(sig, pubA)

Wenn das Ergebnis der Signaturprüfung korrekt ist, bedeutet dies, dass sowohl die Nachricht unverändert ist, als auch dass Alice tatsächlich die Senderin ist. Natürlich gilt das nur, wenn der private Schlüssel tatsächlich geheim und der öffentliche Schlüssel tatsächlich korrekt und die Schlüssellänge lang genug und das Hashverfahren stark genug ist…

3.4.5.3 Challenge-Response

Das Prinzip der digitalen Signatur lässt sich auch für Authentisierungsverfahren nutzen:
- Bob (oder ein Server) kann eine Zahl (oder eine Zeichenkette) msg an Alice schicken,
- Alice kann diese mit Ihrem privaten Schlüssel signieren: sig = V(msg, privA)
- Der Server kann dies mit Alice’s öffentlichem Schlüssel prüfen: msg = V(sig, pubA)

Dabei wird die Nachricht msg, die Bob (oder der Server) schickt, als Challenge (Herausforderung) bezeichnet und Alice’s Antwort als Response (Antwort).

Wenn z.B. Alice’s privater Schlüssel auf einer Smartcard gespeichert ist und diese niemals verlassen kann, dann kann damit ein ziemlich hohes Sicherheitsniveau erreicht werden.

3.5 Hybridverfahren

3.5.1 Grundidee

Da symmetrische Verschlüsselungsverfahren schnell sind und symmetrische Verfahren sicher, können beide Verfahren kombiniert werden, um ein schnelles und sicheres Verfahren zu erzeugen:

- Mit einem asymmetrischen Verfahren wird ein symmetrischer Schlüssel erzeugt, den Mallory nicht kennt, auch wenn er sämtliche Nachrichten mitgehört hat.
- Mit dem symmetrischen Schlüssel wird anschließend der eigentliche Nachrichtenaustausch durchgeführt.

3.5.2 Mathematik 2: Diffie-Hellman

Das klassische asymmetrische Verfahren zur Erzeugung eines symmetrischen Schlüssels ist das Verfahren von Diffie und Hellman:

- Alice und Bob einigen sich auf eine Primzahl p und eine Zahl g < p, die ausgetauscht werden, womit auch Mallory diese sehen kann.
- Außerdem denken sich Alice und Bob jeweils eine Zahl a bzw. b (beide kleiner als p), die nicht ausgetauscht werden, damit kennt Mallory diese nicht.
- Nun berechnet Alice die Potenz g^a und zieht solange p ab, bis das Ergebnis kleiner ist als p. Sie berechnet also $g^a \pmod p$. Bob berechnet entsprechend $g^b \pmod p$.
- Diese Ergebnisse werden wieder ausgetauscht, d.h. Mallory kennt auch diese:

Alice	Mallory	Bob
p,g	p,g	p,g
a, g^a		b, g^b
a, $g^a \pmod p$, $g^b \pmod p$	$g^b \pmod p$, $g^a \pmod p$	b, $g^b \pmod p$, $g^a \pmod p$

- Nun bildet Alice die Potenz $(g^b)^a = g^{ba}$ und Bob bildet die Potenz $(g^a)^b = g^{ab} = g^{ba}$, wobei dasselbe rauskommt. Nun ziehen Alice und Bob davon solange p ab, bis das Ergebnis kleiner ist als p und erhalten das gemeinsame Geheimnis $symm = g^{ab} \pmod p$, das als symmetrischer Schlüssel verwendet werden kann, weil Mallory dieses nicht kennt.

$symm = (g^b)^a \pmod p = g^{ab} \pmod p$ $\qquad$ $symm = (g^a)^b \pmod p = g^{ab} \pmod p$

Die Sicherheit des Diffie-Hellman Verfahrens besteht darin, dass Mallory von $g^a \pmod p$ nicht auf a schließen kann. Dies nennt man das Problem des diskreten Logarithmus.

3.5.3 SSL / TLS

Am verbreitetsten sind Hybridverfahren im SSL bzw. TLS Protokoll zur Verschlüsselung des http-Protokolls (zu https):

- Der Client schickt dem Server eine Challenge.
- Der Server verschlüsselt diese mit seinem privaten Schlüssel, womit der Cient die Identität des Servers prüfen kann. Die Identität des Clients wird in der Regel nicht kryptografisch geprüft, weil der Client keine Schlüssel hat oder der Server den öffentlichen Schlüssel des Clients nicht kennt.
- Der Client denkt sich einen symmetrischen Schlüssel, verschlüsselt diesen mit dem öffentlichen Schlüssel des Servers. Damit kann Mallory diesen nicht lesen, aber der Server kann ihn entschlüsseln. Damit kennen Client und Server denselben symmetrischen Schlüssel mit dem von da an die Kommunikation verschlüsselt werden kann.

Bei diesem sogenannten Handshake verhandeln Client und Server außerdem, welche Krypto-Algorithmen angewendet werden sollen.

Übung: Warum ist das sinnvoll?

3.6 Kryptografische Hashfunktionen

3.6.1 Grundlagen

Eine „Hashfunktion“ (oder „Einwegfunktion“) hf berechnet aus einer Zahl oder einer Nachricht msg einen „Hashwert“ hash = hf(msg), so dass diese Rechnung nicht (leicht) umgekehrt werden kann. Es soll also (fast) unmöglich sein, zu einem vorgegebenen Hashwert eine passende Nachricht zu finden.

Anmerkung: Hashfunktionen gibt es in der Informatik viele, z.B. Prüfsummen, Quersumme o.ä. Diese sind aber leicht umkehrbar. Für eine kryptografische Hashfunktion ist also gerade die Nichtumkehrbarkeit der entscheidende Punkt.

Kryptografische Hashfunktionen werden verwendet

- bei der Speicherung von Passwörtern
- bei der digitalen Signatur, vgl. 3.4.5
- bei Challenge Response und damit auch bei Hybridverfahren, vgl. 3.4.5.3 und 3.5

3.6.2 Sicherheit

Bei einer kryptografischen Hashfunktion wird Nachrichten beliebiger Länge jeweils ein Hashwert mit beschränkter Länge zugeordnet. Da es damit sehr viel mehr Nachrichten als Hashwerte gibt, muss es zwangsläufig verschiedene Nachrichten mit demselben Hashwert geben, sogenannte „Kollisionen“. Die Sicherheit des Verfahrens besteht darin, dass es sehr schwer sein soll, zu einem vorgegebenen Hashwert hash eine Kollision msg zu finden.

3.6.2.1 Länge des Hashwerts

Damit Kollisionen nicht durch einfaches Ausprobieren gefunden werden können, haben die Hashwerte von sicheren Hashverfahren heutzutage eine Länge von mindestens 160 Bit.

3.6.2.2 Wörterbuch-Attacken und Rainbow-Tables

Bei einer Wörterbuchattacke werden zum Durchprobieren nur Wörter aus einem vorgegebenen Wörterbuch gehasht und mit dem vorgegebenen hash verglichen. Dadurch gibt es im Gegensatz zum vollständigen Ausprobieren keine Erfolgsgarantie, dafür ist das Ausprobieren nach einiger Zeit beendet. Deshalb sollen z.B. Passwörter möglichst nicht aus einem Wörterbuch ausgewählt werden.

In einer Rainbow-Table werden möglichst viele Hashwerte und passende Nachrichten gespeichert. Taucht der vorgegebene Hashwert hash in der Rainbow-Table auf, braucht man eine passende Nachricht nur noch abzulesen. Solchen Rainbow-Tables liegen üblicherweise ebenfalls Wörterbücher zugrunde. Als Gegenmittel wird das sogenannte „Salt“ verwendet, d.h. die Nachricht wird vor dem Berechnen des Hashwerts um einige Zeichen – das Salt – verlängert, so dass die Nachricht normalerweise nicht in einem Wörterbuch enthalten ist.

3.6.2.3 Effizienz

Im Gegensatz zu fast allen IT-Anwendungen ist es bei kryptografischen Hashfunktionen von Vorteil, wenn die Berechnung lange dauert, z.B. 0,2 – 0,5 Sekunden. Diese Zeitspanne fällt bei der normalen Anwendung kaum auf, verzögert aber Angriffsversuche durch Ausprobieren enorm, weil z.B. das Ausprobieren von nur 100.000 Passwortkombinationen bereits 20.000 – 50.000 Sekunden dauert, das sind immerhin schon fast 6 – 14 Stunden.

3.6.3 Beispiele

Kryptografische Hashfunktionen sind z.B.

Name	Jahr	Länge des Hashwerts	Sicherheit	Anmerkungen
MD4	Vor 1990	128 Bit	unsicher	
MD5	1991	128 Bit	sehr unsicher	
SHA-1	1991	160 Bit	„angeknackst“	Lange Zeit Mittel der Wahl
RIPEMD	1994	128 Bit	unsicher	

RIPEMD-160	1996	160 Bit	sicher	
SHA-2	2000	224, 256, 384, 512 Bit	sicher	
SHA-3 (Keccak)	2008	224, 256, 384, 512 Bit	sicher	Ging aus einem Wettbewerb hervor
bcrypt, scrypt	1999, 2009			Zur Passwortverschlüsselung absichtlich langsam
scrypt, PBKDF2	2009			Passwort-based-key-derivation-functions

3.7 Kryptografische Zufallsgeneratoren

In vielen Fällen hängt die Sicherheit eines Systems davon ab, dass eine Zahl verwendet wird, die Mallory nicht herausbekommen kann, z.B. die Session-Id einer Internet-Browser-Session.

Eine solche Zahl wird meistens durch einen Zufallsgenerator erzeugt. „Normale“ Zufallsgeneratoren erzeugen allerdings gar keine „echten“ Zufallszahlen, sondern eine nachvollziehbare Folge von sogenannten „Pseudozufallszahlen“.

Für eine hohe Sicherheit brauchen wir stattdessen ein Verfahren, das von Mallory nicht mit praktikablem Aufwand geknackt werden kann. Es gibt grundsätzlich zwei Ansätze:

- Echter Zufall, z.B. durch Auslesen physikalischer Schwankungen von Kondensatoren o.ä. oder durch möglichst zufällige Mausbewegungen des Benutzers
- Pseudozufall, indem mit einem geheimen Schlüssel x_1 gestartet wird und daraus mit einer geeigneten mathematischen Funktion f (der „Fortschaltfunktion“) die weiteren Pseudozufallszahlen berechnet werden:
  $x_2 = f(x_1)$, $x_3 = f(x_2)$, $x_4 = f(x_3)$, $x_5 = f(x_4)$, etc. Da Mallory die gesamte Folge selbst weiterentwickeln kann, wenn er die Funktion f und auch nur einen Wert x_i kennt, verwenden wir nur die erste Hälfte von jedem x_i als Zufallszahl, die andere Hälfte halten wir geheim.

Es gibt auch Fortschaltfunktionen, in denen noch ein geheimer Schlüssel verwendet wird, also $x_{i+1} = f(key, x_i)$.

Im Buch von Schmeh finden Sie in Kap. 15 noch viele Informationen zu kryptografischen Zufallsfunktionen.

3.8 Stromchiffren

3.8.1 Grundlagen

Eine Stromchiffre funktioniert ähnlich wie ein kryptografischer Zufallsgenerator. Durch ständiges fortschalten wird eine Folge – ein Strom – von zufälligen Bits generiert.

Diese Folge wird bitweise mit der geheimen Nachricht verknüpft, was einer Benutzung als One-Time-Pad entspricht, vgl. 3.3.5. Speziell für die Verschlüsselung eines Stroms digitaler Signale wie z.B. bei der Mobil- oder VoIP-Telefonie sind Stromchiffren sehr gut geeignet.

3.8.2 Beispiele

Name	**Jahr**	**Sicherheit**	**Anmerkungen**
RC4	1987	Abhängig von der Implementierung	Sehr einfach und sehr schnell
A5	1991	unsicher	Sehr schnell, wird in GSM-Handys verwendet
Trivium	2004-2008	sicher	Wettbewerbssieger, besonders für Hardware-Implementierung
HC-128	2004-2008	sicher	
Salsa20	2004-2008	sicher	

4 Datenschutz

An dieser Stelle möchten wir auf den Begriff und die Aufgaben des „Datenschutz“ eingehen und insbesondere dessen Einordnung und Abgrenzung im Zusammenhang mit der Informationssicherheit darstellen.

4.1 Grundbegriffe

Datenschutz bezeichnet

- den Schutz **natürlicher Personen** vor dem Missbrauch **personenbezogener Daten**.[3]
- „Natürliche Personen“ sind „echte“, lebende Personen im Gegensatz zu juristischen Personen, zu denen z.B. Kapitalgesellschaften gehören.
- „Personenbezogene Daten“ sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).[4]

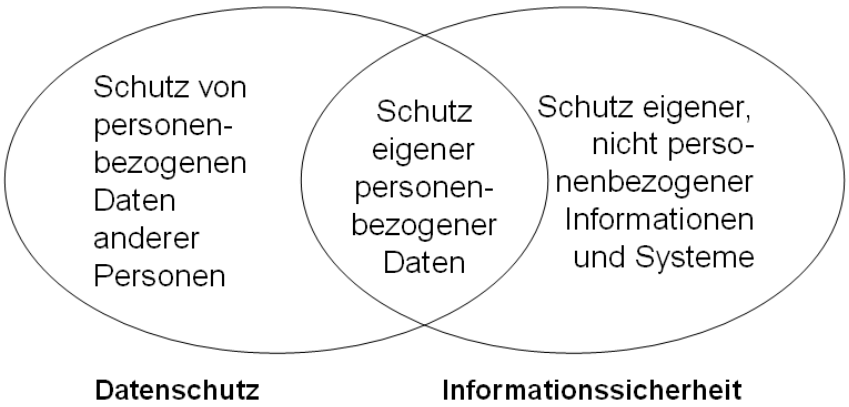
Anmerkungen:

- Datenschutz betrifft personenbezogene Daten hauptsächlich aber nicht ausschließlich bei der Verarbeitung mit technischen Geräten.
- Auf englisch spricht man von „privacy“ (allgemein: Schutz der Privatsphäre), „data privacy“ oder „information privacy“. Im britischen Sprachgebrauch spricht man auch von „data protection“. Im amerikanischen ist dies unüblich.

4.2 Zusammenhang und Abgrenzung

Informationssicherheit und Datenschutz haben viele Berührpunkte, trotzdem handelt es sich um unabhängige Begriffe:

- Die Informationssicherheit betrifft nicht nur personenbezogene Daten, damit geht Informationssicherheit teilweise über den Datenschutz hinaus.
- Informationssicherheit bezieht sich auf die eigenen Daten. Datenschutz betrifft aber auch den Schutz von personenbezogenen Daten anderer Personen. Damit geht auch Datenschutz teilweise über die Informationssicherheit hinaus.



4.3 Bundesdatenschutzgesetz (BDSG)

4.3.1 Warum brauchen wir Datenschutzgesetze?

Datenschutz befasst sich mit dem Schutz von personenbezogenen Daten, speziell derer von anderen Leuten. Betrachten wir die Interessenslagen:

- Niemand hat ein eigenes Interesse an Schutz und Geheimhaltung der Daten über andere.
- Im Gegenteil: Unternehmen haben ein großes Interesse an Informationen über Personen, die zum potenziellen Kundenkreis gehören könnten.

[3] http://de.wikipedia.org/wiki/Datenschutz
[4] Bundesdatenschutzgesetz (BDSG) 2001, §3 (1)

- Darauf aufbauend gibt es eine ganze Branche, die mit Kundendaten handelt, um aus dem genannten Unternehmensinteresse Geschäftsmodelle zu entwickeln.
- Selbst Privatpersonen sammeln oft Daten über andere Personen.

Eine weitere Beobachtung besteht darin, dass es für eine natürliche Person unmöglich ist, selbst zu kontrollieren, wo welche Daten über einen selbst im Umlauf sind.

In diesem Zusammenspiel sind Unternehmen in der Position des Stärkeren, die natürlichen Personen in der Position des Schwächeren und sollen deshalb vom Staat durch Gesetze geschützt werden. Hierzu dient in Deutschland das Bundesdatenschutzgesetz (BDSG) und weitere Gesetze, in der EU entsprechende Richtlinien.

4.3.2 Entstehung

Ursprünglich entstand Datenschutz als Gegenbewegung zu staatlichen Datensammlungen, die mit dem Aufkommen der elektronischen Datenverarbeitung ab den 1960er Jahren möglich wurden. Beispiele dafür sind die Rasterfahndung gegen Terrorismus in den 1970ern und die Volkszählung in Deutschland Anfang der 1980er Jahre.

Damit befasste sich der Datenschutz ursprünglich vor allem mit der Datenverarbeitung in Behörden, erst mit der Zeit rückten mehr und mehr auch Privatunternehmen in den Fokus.

Aufgrund einer Klage gegen die Volkszählung wurde 1983 vom Bundesverfassungsgericht das „Grundrecht auf informationelle Selbstbestimmung“ definiert und in dessen Folge das Bundesdatenschutzgesetz entwickelt.[5]

Heute gilt das Bundesdatenschutzgesetz für alle Personen und Organisationen in Deutschland. Landesdatenschutzgesetze gelten für die Behörden des jeweiligen Bundeslandes, z.B. unterliegt die Hochschule Ludwigshafen dem Landesdatenschutzgesetz Rheinland-Pfalz.

4.3.3 Highlights des deutschen Bundesdatenschutzgesetzes

Im Folgenden zählen wir einige Aspekte des deutschen Bundesdatenschutzgesetzes auf. Diese Aufzählung ist für das Selbststudium gedacht und soll Ihnen eine erste Orientierung ermöglichen, ohne das gesamte BDSG zu lesen.

- Geltungsbereich: „Erhebung, Verarbeitung und Nutzung personenbezogener Daten durch **öffentliche** Stellen …, **nicht-öffentliche Stellen**, soweit sie die Daten unter Einsatz von Datenverarbeitungsanlagen verarbeiten, nutzen oder dafür erheben oder die Daten in oder aus nicht automatisierten Dateien verarbeiten, nutzen oder dafür erheben, es sei denn, die Erhebung, Verarbeitung oder Nutzung der Daten erfolgt ausschließlich für persönliche oder familiäre Tätigkeiten.“ (BDSG, §1, (2))
- Zur Verarbeitung gehört Speichern, Verändern, Sperren, Übermitteln, Löschen (§3, (4))
- „**Besondere Arten personenbezogener Daten** sind Angaben über die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen, Gewerkschaftszugehörigkeit, Gesundheit oder Sexualleben.“ (§3, (9))
- **„Datenvermeidung und Datensparsamkeit**: Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten und die Auswahl und Gestaltung von Datenverarbeitungssystemen sind an dem Ziel auszurichten, so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen. Insbesondere sind personenbezogene Daten zu anonymisieren oder zu pseudonymisieren, soweit dies nach dem Verwendungszweck möglich ist und keinen im Verhältnis zu dem angestrebten Schutzzweck unverhältnismäßigen Aufwand erfordert.“ (§3a)
- **Zulässigkeit**: „Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten sind nur zulässig, soweit dieses Gesetz oder eine andere Rechtsvorschrift dies erlaubt oder anordnet oder der Betroffene eingewilligt hat.“ (§4, (1))
- „Werden personenbezogene Daten … erhoben, so ist <der Betroffene>, sofern er nicht bereits auf andere Weise Kenntnis erlangt hat, von der verantwortlichen Stelle über
  1. die Identität der verantwortlichen Stelle,
  2. die Zweckbestimmungen der Erhebung, Verarbeitung oder Nutzung und

[5] http://de.wikipedia.org/wiki/Volkszählungsurteil

3. die Kategorien von Empfängern nur, soweit der Betroffene nach den Umständen des Einzelfalles nicht mit der Übermittlung an diese rechnen muss,
zu unterrichten." (§4, (3))

- „**Beauftragter für den Datenschutz**: … Stellen, die personenbezogene Daten automatisiert verarbeiten, haben einen Beauftragten für den Datenschutz schriftlich zu bestellen… Die Sätze 1 und 2 gelten nicht für die nichtöffentlichen Stellen, die in der Regel höchstens neun Personen ständig mit der automatisierten Verarbeitung personenbezogener Daten beschäftigen." (§4f)
- „**Datengeheimnis**: Den bei der Datenverarbeitung beschäftigten Personen ist untersagt, personenbezogene Daten unbefugt zu erheben, zu verarbeiten oder zu nutzen (Datengeheimnis). Diese Personen sind, soweit sie bei nicht-öffentlichen Stellen beschäftigt werden, bei der Aufnahme ihrer Tätigkeit auf das Datengeheimnis zu verpflichten. Das Datengeheimnis besteht auch nach Beendigung ihrer Tätigkeit fort." (§5)
- „**Schadensersatz**: Fügt eine verantwortliche Stelle dem Betroffenen durch eine nach diesem Gesetz oder nach anderen Vorschriften über den Datenschutz unzulässige oder unrichtige Erhebung, Verarbeitung oder Nutzung seiner personenbezogenen Daten einen Schaden zu, ist sie oder ihr Träger dem Betroffenen zum Schadensersatz verpflichtet. Die Ersatzpflicht entfällt, soweit die verantwortliche Stelle die nach den Umständen des Falles gebotene Sorgfalt beachtet hat." (§7)
- „**Technische und organisatorische Maßnahmen**: Öffentliche und nicht-öffentliche Stellen, die selbst oder im Auftrag personenbezogene Daten erheben, verarbeiten oder nutzen, haben die technischen und organisatorischen Maßnahmen zu treffen, die erforderlich sind, um die Ausführung der Vorschriften dieses Gesetzes, insbesondere die **in der Anlage** zu diesem Gesetz genannten Anforderungen, zu gewährleisten. Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem **angemessenen Verhältnis zu dem angestrebten Schutzzweck** steht." (§9)
- „**Datenerhebung und -speicherung für eigene Geschäftszwecke**:
  (1) Das Erheben, Speichern, Verändern oder Übermitteln personenbezogener Daten oder ihre Nutzung als Mittel für die Erfüllung eigener Geschäftszwecke ist zulässig,
  1. wenn es für die Begründung, Durchführung oder Beendigung eines rechtsgeschäftlichen oder rechtsgeschäftsähnlichen Schuldverhältnisses mit dem Betroffenen erforderlich ist,
  2. soweit es zur Wahrung berechtigter Interessen der verantwortlichen Stelle erforderlich ist und kein Grund zu der Annahme besteht, dass das schutzwürdige Interesse des Betroffenen an dem Ausschluss der Verarbeitung oder Nutzung überwiegt oder
  3. wenn die Daten allgemein zugänglich sind … es sei denn, dass das schutzwürdige Interesse des Betroffenen an dem Ausschluss der Verarbeitung oder Nutzung gegenüber dem berechtigten Interesse der verantwortlichen Stelle offensichtlich überwiegt…
  (3) Die Verarbeitung oder Nutzung personenbezogener Daten für Zwecke des Adresshandels oder der Werbung ist zulässig, soweit der Betroffene eingewilligt hat… Darüber hinaus ist die Verarbeitung oder Nutzung personenbezogener Daten zulässig, soweit es sich um … Berufs-, Branchen- oder Geschäftsbezeichnung, seinen Namen, Titel, akademischen Grad, seine Anschrift und sein Geburtsjahr beschränken, und die Verarbeitung oder Nutzung erforderlich ist
  1. für Zwecke der Werbung für eigene Angebote der verantwortlichen Stelle, die diese Daten mit Ausnahme der Angaben zur Gruppenzugehörigkeit beim Betroffenen nach Absatz 1 Satz 1 Nummer 1 oder aus allgemein zugänglichen Adress-, Rufnummern-, Branchen- oder vergleichbaren Verzeichnissen erhoben hat,
  2. für Zwecke der Werbung im Hinblick auf die berufliche Tätigkeit des Betroffenen und unter seiner beruflichen Anschrift oder
  3. für Zwecke der Werbung für Spenden, die nach § 10b Absatz 1 und § 34g des Einkommensteuergesetzes steuerbegünstigt sind…
  (4) Widerspricht der Betroffene bei der verantwortlichen Stelle der Verarbeitung oder Nutzung seiner Daten für Zwecke der Werbung oder der Markt- oder Meinungsforschung, ist eine Verarbeitung oder Nutzung für diese Zwecke unzulässig." (§28)
- §28a: Datenübermittlung an Auskunfteien
- §28b: Scoring
- §29: Geschäftsmäßige Datenerhebung und -speicherung zum Zweck der Übermittlung
- § 30: Geschäftsmäßige Datenerhebung und -speicherung zum Zweck der Übermittlung in anonymisierter Form

- § 30a: Geschäftsmäßige Datenerhebung und -speicherung für Zwecke der Markt- oder Meinungsforschung.
- § 32: Datenerhebung, -verarbeitung und -nutzung für Zwecke des Beschäftigungsverhältnisses
- **„Benachrichtigung des Betroffenen**:
  (1) Werden erstmals personenbezogene Daten für eigene Zwecke ohne Kenntnis des Betroffenen gespeichert, ist der Betroffene von der Speicherung, der Art der Daten, der Zweckbestimmung der Erhebung, Verarbeitung oder Nutzung und der Identität der verantwortlichen Stelle zu benachrichtigen. Werden personenbezogene Daten geschäftsmäßig zum Zweck der Übermittlung ohne Kenntnis des Betroffenen gespeichert, ist der Betroffene von der erstmaligen Übermittlung und der Art der übermittelten Daten zu benachrichtigen. Der Betroffene ist in den Fällen der Sätze 1 und 2 auch über die Kategorien von Empfängern zu unterrichten, soweit er nach den Umständen des Einzelfalles nicht mit der Übermittlung an diese rechnen muss.
  (2) Eine Pflicht zur Benachrichtigung besteht nicht, wenn
  1. der Betroffene auf andere Weise Kenntnis von der Speicherung oder der Übermittlung erlangt hat, …
  7. die Daten für eigene Zwecke gespeichert sind und
  a) aus allgemein zugänglichen Quellen entnommen sind und eine Benachrichtigung wegen der Vielzahl der betroffenen Fälle unverhältnismäßig ist, ...
  8. die Daten geschäftsmäßig zum Zweck der Übermittlung gespeichert sind und
  a) aus allgemein zugänglichen Quellen entnommen sind, soweit sie sich auf diejenigen Personen beziehen, die diese Daten veröffentlicht haben, oder
  b) es sich um listenmäßig oder sonst zusammengefasste Daten handelt (§ 29 Absatz 2 Satz 2) und eine Benachrichtigung wegen der Vielzahl der betroffenen Fälle unverhältnismäßig ist. (§33)“
- „**Auskunft an den Betroffenen**:
  (1) Die verantwortliche Stelle hat dem Betroffenen auf Verlangen Auskunft zu erteilen über
  1. die zu seiner Person gespeicherten Daten, auch soweit sie sich auf die Herkunft dieser Daten beziehen,
  2. den Empfänger oder die Kategorien von Empfängern, an die Daten weitergegeben werden, und
  3. den Zweck der Speicherung.
  Der Betroffene soll die Art der personenbezogenen Daten, über die Auskunft erteilt werden soll, näher bezeichnen. Die Auskunft über die Herkunft und die Empfänger kann verweigert werden, soweit das Interesse an der Wahrung des Geschäftsgeheimnisses gegenüber dem Informationsinteresse des Betroffenen überwiegt…
  (8) Die Auskunft ist unentgeltlich…“ (§34)
- „**Berichtigung, Löschung und Sperrung von Daten**:
  (2) … Personenbezogene Daten sind zu löschen, wenn
  1. ihre Speicherung unzulässig ist,…
  3. sie für eigene Zwecke verarbeitet werden, sobald ihre Kenntnis für die Erfüllung des Zwecks der Speicherung nicht mehr erforderlich ist, oder
  4. sie geschäftsmäßig zum Zweck der Übermittlung verarbeitet werden und eine Prüfung … ergibt, dass eine längerwährende Speicherung nicht erforderlich ist…
  (5) Personenbezogene Daten dürfen nicht für eine automatisierte Verarbeitung … erhoben, verarbeitet oder genutzt werden, soweit der Betroffene dieser bei der verantwortlichen Stelle widerspricht und eine Prüfung ergibt, dass das schutzwürdige Interesse des Betroffenen wegen seiner besonderen persönlichen Situation das Interesse der verantwortlichen Stelle an dieser Erhebung, Verarbeitung oder Nutzung überwiegt.“ (§35)
- **Anlage (zu § 9 Satz 1)**: „Werden personenbezogene Daten automatisiert verarbeitet oder genutzt, ist die innerbehördliche oder innerbetriebliche Organisation so zu gestalten, dass sie den besonderen Anforderungen des Datenschutzes gerecht wird. Dabei sind insbesondere Maßnahmen zu treffen, die je nach der Art der zu schützenden personenbezogenen Daten oder Datenkategorien geeignet sind,
  1. Unbefugten den Zutritt zu Datenverarbeitungsanlagen, mit denen personenbezogene Daten verarbeitet oder genutzt werden, zu verwehren (Zutrittskontrolle),
  2. zu verhindern, dass Datenverarbeitungssysteme von Unbefugten genutzt werden können (Zugangskontrolle),
  3. zu gewährleisten, dass die zur Benutzung eines Datenverarbeitungssystems Berechtigten

ausschließlich auf die ihrer Zugriffsberechtigung unterliegenden Daten zugreifen können, und dass personenbezogene Daten bei der Verarbeitung, Nutzung und nach der Speicherung nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können (Zugriffskontrolle),
4. zu gewährleisten, dass personenbezogene Daten bei der elektronischen Übertragung oder während ihres Transports oder ihrer Speicherung auf Datenträger nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können, und dass überprüft und festgestellt werden kann, an welche Stellen eine Übermittlung personenbezogener Daten durch Einrichtungen zur Datenübertragung vorgesehen ist (Weitergabekontrolle),
5. zu gewährleisten, dass nachträglich überprüft und festgestellt werden kann, ob und von wem personenbezogene Daten in Datenverarbeitungssysteme eingegeben, verändert oder entfernt worden sind (Eingabekontrolle),
6. zu gewährleisten, dass personenbezogene Daten, die im Auftrag verarbeitet werden, nur entsprechend den Weisungen des Auftraggebers verarbeitet werden können (Auftragskontrolle),
7. zu gewährleisten, dass personenbezogene Daten gegen zufällige Zerstörung oder Verlust geschützt sind (Verfügbarkeitskontrolle),
8. zu gewährleisten, dass zu unterschiedlichen Zwecken erhobene Daten getrennt verarbeitet werden können.
Eine Maßnahme nach Satz 2 Nummer 2 bis 4 ist insbesondere die Verwendung von dem Stand der Technik entsprechenden Verschlüsselungsverfahren.“

4.3.4 Beobachtungen

Datenvermeidung und Datensparsamkeit

Wesentliche Prinzipien zum Schutz des Bürgers sind die Grundsätze der Datenvermeidung und Datensparsamkeit, denn nur solche Informationen, die erst gar nicht erfasst werden, können auch unter keinen Umständen missbraucht werden. Speziell das Gebot der Datensparsamkeit gibt auch eine wichtige Handhabe für einzelne Personen, sich gegen die Speicherung ihrer Daten zu wehren.

Auskunftspflichten

Eine weitere Handhabe für den einzelnen Bürger, ist sein **Auskunftsrecht**, das er gegenüber jeder datenspeichernden Stelle einmal jährlich kostenfrei ausüben kann.

Ob die angefragte Stelle die Daten tatsächlich liefert ist eine andere Frage, grundsätzlich ist aber jede solche Stelle bei Bußgeldandrohung dazu verpflichtet.

Übung 6: Überlegen Sie sich, wie eine solche Anfrage nach Datenschutz Selbstauskunft an die Telekom oder die Schufa aussehen könnte.

Allgemeines

Das Datenschutzgesetz versucht, eine Balance zwischen den Rechten und den Pflichten von Unternehmen in Bezug auf die Verarbeitung personenbezogener Daten herzustellen. Überlegungen zu den wirtschaftlichen Chancen und Risiken der Unternehmen spielen dabei keine Rolle.

Die Vorgaben sind selten konkret und lassen Interpretations- und Ermessensspielräume. („*…das schutzwürdige Interesse des Betroffenen wegen seiner besonderen persönlichen Situation das Interesse der verantwortlichen Stelle an dieser Erhebung, Verarbeitung oder Nutzung überwiegt*“)

Es kann beobachtet werden, dass in der Informationssicherheit bzw. im Datenschutz grundverschiedene Terminologien verwendet werden. Informationssicherheit ist risikozentriert und auf monetären Nutzen bzw. die Verhinderung monetären Schadens fokussiert, während Datenschutz ein streng juristisches Thema ist, bei dem Geld keine Rolle spielt.

Trotzdem wird im Datenschutzgesetz ein Zusammenhang zwischen Datenschutz und Informationssicherheit hergestellt (vgl. §9 und die Anlage dazu), bei dem ganz im Sinne der Informationssicherheit die Angemessenheit möglicher Schutzmaßnahmen betont wird: „*Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht.*“

Es bleibt allerdings offen, wie die Angemessenheit beurteilt werden kann, immerhin bedeutet dies einen Vergleich zwischen einem Persönlichkeitsrecht (Gefährdung des Rechts auf informationelle Selbstbestimmung) und einer monetären Größe (Aufwand für Einführung und Betrieb der Sicherheitsmaßnahmen).

Internationales

Die genannten Ausführungen gelten zunächst nur für Deutschland. Die Gesetzgebung in Europa ist ähnlich. In USA gelten die meisten dieser Regelungen allerdings nicht oder nicht in derselben Art.

4.4 Vergleich zwischen Datenschutz und Informationssicherheit

Die folgende Aufstellung liefert die wesentlichen Unterschiede zwischen Datenschutz und Informationssicherheit:

	Datenschutz	**Informationssicherheit**
Bezeichnungen	Privacy, Data Protection	Datensicherheit, Information Security
Ausgangspunkt	Datenschutzgesetze	Eigeninteresse, eingeschränkt auch KonTraG, Basel II, SOX
Welche Daten sind betroffen?	Personenbezogene Daten, speziell eigene personenbezogene Daten und personenbezogene Daten Dritter	Alle Daten des betreffenden Unternehmens bzw. der betreffenden Person
Wer wird geschützt?	Schutz natürlicher Personen, speziell Dritter	Selbstschutz einer Organisation oder Privatperson
Was wird geschützt?	Privatsphäre und Grundrecht zur informationellen Selbstbestimmung	Unternehmenswerte in Form von Informationen und IT-Anlagen
Compliance	Generelle gesetzliche Pflicht durch Datenschutzgesetze und -richtlinien	freiwillige Normen: Unternehmensinterne Richtlinien, nationale (z.B. BSI) und internationale (z.B. ISO 2700x) Standards mit Möglichkeit der Zertifizierung Pflicht zur Kontrolle hoher Risiken nach KonTraG
Schutzmaß-nahmen	Schutz (nur) personenbezogener Daten durch zunächst organisatorische Maßnahmen wie Datenschutzbeauftragter, Mitarbeiterschulung. Technische Maßnahmen nach Gesetz aber *in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck*	Einführung von Sicherheitsmanagement und Sicherheitsorganisation. Organisatorische, technische und personelle Maßnahmen für alle Daten und Geräte, die nach Einschätzung der Risikoanalyse schützenswert sind. Dabei können die technischen Maßnahmen, die im Datenschutzgesetz erwähnt werden, mit einbezogen werden.
Verantwortlich	Management und Datenschutzbeauftragter, der laut Gesetz benannt werden muss	Management und Informationssicherheits-beauftragter (freiwillig)

4.5 Datenschutz für Privatpersonen

4.5.1 Allgemeines

Als Privatperson betrifft Sie der Datenschutz in zweierlei Hinsicht:

- Einerseits sind auch Sie dem BDSG verpflichtet und dürfen nicht unbeschränkt personenbezogene Daten anderer natürlicher Personen verarbeiten, außer nach BDSG, §1, Abs. 2 für ausschließlich persönliche oder familiäre Tätigkeiten.

- Andererseits sind Sie Nutznießer des BDSG, da dieses die Speicherung Ihrer eigenen personenbezogenen Daten reguliert. Soweit Sie hierzu durch geeignetes Verhalten beitragen, gibt es dafür auch den Begriff „Selbstdatenschutz“.

Anmerkungen:
- Falls Sie Daten anderer nutzen, um Gewinne zu erzielen, dann ist das gewerblich und nicht persönlich.
- Falls Sie Daten anderer nutzen, um diesen zu schaden, dann fällt das in das Strafrecht und nicht in das Datenschutzrecht.

4.5.2 Persönlichkeitsprofile

Häufig wird Datenschutz von einzelnen Bürgern nicht ernst genommen, da ihnen egal ist, ob z.B.
- der Otto-Versand ihre Kleidergröße kennt und speichert,
- die Postbank weiß, bei welcher Tankstelle Sie tanken.
- Amazon weiß, welche Bücher Sie in den letzten 10 Jahren angesehen bzw. gekauft haben,
- Google weiß, wonach Sie in den letzten 10 Jahren gesucht haben,
- Apple oder Google weiß, wo Sie sich in den letzten drei Jahren aufgehalten haben,
- Facebook alle Ihre Eigenarten, alle Ihre Freunde und alle deren Eigenarten kennt,

Übung 7: Stellen Sie sich Situationen vor, in denen eine solche Information oder eine Kombination solcher Informationen Ihnen unangenehm sein könnte. Stellen Sie sich vor, was eine Kombination all dieser Informationen über sie aussagen könnte. Denken Sie dabei auch an kleinere Personengruppen, z.B. Hartz-4-Empfänger, Drogenabhängige, Homosexuelle, HIV-infizierte, etc.

4.5.3 Beispiele

PRISM-Skandal

Übung 8: Was haben Sie über die Datensammlungen der Geheimdienste erfahren? Inwiefern könnte Sie das beunruhigen? Inwiefern könnten Sie sich vorstellen, dass jemand dadurch beunruhigt wird?

Beispiel Payback-Karte

Eine interessante Frage ist: „Was ist der monetäre Wert personenbezogener Daten?“ Mit Payback-Karten werden z.B. anonyme Einkaufsvorgänge ent-anonymisiert. Viele Kunden geben dafür Ihre Zustimmung gegen vergleichsweise geringe monetäre Vorteile.

Eine weitere interessante Frage ist, ob dieselben Kunden Ihre Zustimmung geben würden, wenn ihnen klar wäre, welche Interessens- und Gewohnheitsprofile erstellt werden können, wenn die Einkaufsdaten aus vielen Einkaufsvorgängen über längere Zeit kumuliert werden.

Beispiel Vorratsdatenspeicherung

Das Internet ist anonym, oder? Solange ich nicht irgendwo meinen Namen eintrage, weiß niemand wer ich bin, oder?

Schauen wir genauer hin:
- Wenn Sie eine Pelzjacke beim Onlineversand bestellen, müssen Sie Ihre Adresse angeben.
- Wenn Sie ein Datenpaket bei einem Internetserver bestellen, müssen Sie Ihre Computeradresse angeben.

Damit das Datenpaket zu Ihnen nach hause auf den Rechner kommen kann, muss es irgendwie adressiert werden. In der Regel sind Sie verbunden mit einem Internetprovider, der Ihre Telefonnummer kennt und Ihnen für eine bestimmte Zeit eine Internetadresse (IP-Adresse) gibt. Wenn Sie einen Webserver kontaktieren, senden Sie dem Internetserver diese IP-Adresse und dieser sendet Ihnen das Datenpaket.

Ihre Sicht:
- Der Webserver sieht nur Ihre IP-Adresse, kennt aber nicht Ihre Telefonnummer und erst recht nicht Ihren Namen und Ihre Adresse. Diesem gegenüber sind Sie anonym.

- Der Internetprovider kennt Ihre IP-Adresse und Ihre Telefonnummer. Auf seinen Vermittlungsservern kann er sehen, welche Webseiten Sie abrufen.
  - Im Normalfall haben Sie einen Vertrag mit dem Internetprovider und er kennt auch Ihren Namen. Damit sind Sie diesem gegenüber nicht anonym.
  - Bei einer Call-by-Call-Verbindung kennt der Internetprovider evtl. nur Ihre Telefonnummer aber nicht Ihren Namen, solange er diesen nicht von Ihrem Telefonprovider erfährt.

Sicht des Internetproviders:
- Der Internetprovider darf Ihre Adresse (oder Telefonnummer) aus Datenschutzgründen nur für Abrechnungszwecke verwenden, aber nicht, um Ihre Surfgewohnheiten zu kontrollieren – außer es gibt gesetzliche Gründe, Sie zu überwachen, z.B. im Rahmen der Strafverfolgung.
- Er muss Ihre personenbezogenen Daten löschen, nachdem die Abrechnung erfolgt ist. Wenn Sie eine Flatrate haben, darf er Ihre Daten gar nicht speichern, außer zur Erbringung seines Service – oder für Statistiken, die ihm helfen sollen, seinen Service zu verbessern.

Sicht der Strafverfolger:
- Wenn Strafverfolger einen Internetserver mit illegalem Material (kopiergeschützte Inhalte, Kinderpornographie) entdecken, können diese versuchen, dessen Datenverkehr abzuhören und erfahren damit die IP-Adressen von dessen (illegalen) Benutzern.
- Über die IP-Adressen kann der Internetprovider der Benutzer gefunden werden, bei dem dann angefragt werden kann, welcher Name dazu gehört.
- Bei Call-by-Call-Verbindungen kennt der Provider nur die Telefonnummer und man muss auch noch die Telefongesellschaft fragen.
- Wenn die Daten inzwischen gelöscht sind, ist eine Strafverfolgung nicht möglich.

Beim Gesetzentwurf zur Vorratsdatenspeicherung ging es darum, die Internetprovider zu verpflichten, die Verbindungsdaten von Handy, Telefon und E-Mail sowie zugeordnete IP-Adressen für ein halbes Jahr zu speichern, damit die Strafverfolgung nicht aufgrund zwischenzeitlich gelöschter Daten fehlschlägt.

Ein wesentliches Problem dabei ist die grobe Unverhältnismäßigkeit zwischen
- den riesigen Mengen gespeicherter Verbindungsdaten, die generell die Bildung von Surf- und damit von Persönlichkeitsprofilen erlauben und
- den außerordentlich geringen Mengen strafrechtlich relevanter Daten

Interessant ist dabei die Frage, ob eine IP-Adresse ein personenbezogenes Datum ist. Die Antwort ist bisher noch umstritten. Ein Ansatz könnte sein:
- Wenn die IP-Adresse Rückschlüsse erlaubt, wer damit im Internet aktiv ist oder war, dann ist sie personenbezogen.
- Andernfalls ist sie es nicht (mehr), z.B. wenn alle Zuordnungsinformationen gelöscht sind.

Beispiel Google Street View

Frage: Sind Fotos von Häusern personenbezogen?

Antwort: Ja, weil sie mit Hilfe von Adressbüchern den Personen zugeordnet werden können, die dort wohnen und damit einen Rückschluss auf deren persönliche Verhältnisse und Vorlieben erlauben.

Frage: Ist es denn dann auch personenbezogen, wenn ich beim Spazierengehen auf dem Klingelschild eines bestimmten Hauses nachsehe, wer da wohnt?

Antwort: Ja, aber es ist ein großer Unterschied in der Verhältnismäßigkeit, ob Sie zu jedem Haus, das Sie sehen möchten, hingehen müssen, oder ob sie alles auf einmal zuhause serviert bekommen.

Beispiel Social Media / Facebook

Übung 9:
- Was bekomme ich von Social Media Plattformen wie Facebook?
- Was gebe ich dafür an eine Social Media Plattform?
- Welche Nachteile können sich für mich ggfs. ergeben bzw. welche Effekte könnte ich als Nachteil betrachten?

- Wie war das eigentlich früher und was ist der Unterschied dazu?

5 Informationssicherheitsmanagement

5.1 Definition

„Informationssicherheitsmanagement“ bezeichnet die betriebliche Aufgabe, die Informationssicherheit im Unternehmen auf ein angemessenes Niveau zu bringen und dort zu halten.

Unter Berücksichtigung von 2.2, 2.3, 2.5 und 2.6 umfasst dies die **kontinuierliche** Ermittlung, Umsetzung und Kontrolle geeigneter und **angemessener** (personeller, organisatorischer und technischer) Sicherheitsmaßnahmen zum Schutz der Vertraulichkeit, Integrität und Verfügbarkeit von Informationen und IT-Systemen im Unternehmen.

5.2 Angemessenheit einzelner Sicherheitsmaßnahmen

5.2.1 Gefährdungen und Maßnahmen

Wie wir in 2.4 und 2.6 bereits betrachtet hatten, stehen Sicherheitsmaßnahmen und Sicherheitsgefährdungen in einem engen Zusammenhang:

- Zu jeder Gefährdung ist in der Regel eine Maßnahme denkbar, die diese Gefährdung verhindert oder zumindest reduziert.
- Jede Maßnahme adressiert mindestens eine Gefährdung, andernfalls wäre die Maßnahme aus Sicherheitssicht irrelevant.

Die Ermittlung geeigneter Sicherheitsmaßnahmen (wie in 5.1 gefordert), geht also einher mit der Ermittlung bzw. Kenntnis der bestehenden Sicherheitsgefährdungen.

5.2.2 Beurteilung der Angemessenheit

Um die Angemessenheit einer Sicherheitsmaßnahme beurteilen zu können, sollten möglichst genaue Informationen vorliegen über

- **Business Impact**: Die Höhe des wirtschaftlichen Schadens, der eintritt, falls die Gefährdung tatsächlich zu einem Sicherheitsvorfall führt. Hierbei sollten die möglichen Schadensszenarien für Vertraulichkeitsverlust, Integritätsverlust oder Verfügbarkeitsverlust betrachtet werden.
- **Eintrittswahrscheinlichkeit**: Die Wahrscheinlichkeit, dass die Gefährdung tatsächlich zu einem (absichtlichen oder unabsichtlichen, vgl. 2.4.2) Sicherheitsvorfall führt.
- **Aufwand**: Wieviel Geld und Arbeitszeit muss aufgewendet werden, um die Maßnahme umzusetzen, die die betrachteten Gefährdungen verhindert. Um eine monetäre Kennzahl zu erhalten, sollte auch die Arbeitszeit in Geld umgerechnet werden.

Aus Business Impact und Eintrittswahrscheinlichkeit kann nun der „erwartete Verlust“ (expected loss oder composite risk index) berechnet werden gemäß

- **Erwarteter Verlust** = Business Impact x Eintrittswahrscheinlichkeit

Das Basiskonzept der Risikoanalyse besteht nun darin, den erwarteten Verlust mit dem Aufwand zu vergleichen und daraus die Angemessenheit der Maßnahme abzuleiten.

Übung 10: Führen Sie die genannten Betrachtungen für mindestens drei Sicherheitsmaßnahmen Ihrer Wahl durch.

5.2.3 Schwierigkeiten

Das dargestellte Verfahren zur Risikoanalyse birgt mehrere Schwierigkeiten:

- Der Business Impact und noch mehr die Eintrittswahrscheinlichkeit eines möglichen Sicherheitsvorfalls sind nur sehr ungenau zu beziffern.
- Die sehr große Anzahl möglicher Sicherheitsvorfälle, Gefährdungsvektoren (vgl. 2.4.6) und Sicherheitsmaßnahmen machen das Verfahren selbst sehr aufwändig.

Insbesondere gilt für die Ermittlung des Business Impact die Problematik, dass es worst case Szenarien geben könnte, die eher unwahrscheinlich sind oder Standardszenarien, die eher wahrscheinlich sind.

Wenn im Unternehmen mehrere Personen mit der Risikoanalyse beschäftigt werden, könnten dadurch Ungenauigkeiten auftreten, dass diese unterschiedliche Beertungskriterien einsetzen.

Durch die große Anzahl von Maßnahmen, die beurteilt werden sollen, wird es einige Zeit dauern, bis alle Risikoanalysen beendet sind. Bis dahin kann sich aber die Bedrohungslage bereits wieder geändert haben aufgrund neuer Systeme oder neuer erkannter Gefährdungen.

Zusammengefasst ergibt sich, dass die Ergebnisse der Risikoanalyse, wenn sie flächendeckend durchgeführt wird, sehr ungenau sind, so dass der resultierende Wert für den erwarteten Verlust mit mehr als 100% Fehler behaftet sein kann. Die Nutzung dieses Verfahrens zur Ermittlung der Angemessenheit einer Sicherheitsmaßnahme ist also nicht unbedingt präziser als eine intuitive und spontane Entscheidung eines erfahrenen Sicherheitsexperten oder eines Sicherheitsgremiums.

5.3 Die Managementaufgabe

Die bisher geschilderten Vorgehensweisen rechtfertigen nicht den Begriff des Sicherheits-"managements". Insbesondere können die genannten Risikobetrachtungen von Nicht-Managern durchgeführt werden. Wir wollen deshalb hier beleuchten, worin tatsächlich die Managementaufgaben bestehen.

5.3.1 Informationssicherheit als Qualitätsziel

Die wichtigste Managementaufgabe besteht darin, sicherzustellen, dass die in 5.1 genannten Aufgaben auch tatsächlich durchgeführt werden und zwar mit der notwendigen Kontinuität.

Da Sicherheit nicht direkt zur Wertschöpfung eines Unternehmens beiträgt, muss Informationssicherheit vom Top Management als Qualitätsziel festgelegt werden. Andernfalls würde jede nachgeordnete Abteilung die Informationssicherheit bei der Verfolgung der ihr zugeordneten Ziele zurückstellen, so dass im Endeffekt keine Aktionen durchgeführt werden.

Die Festlegung der Informationssicherheit als Qualitätsziel erfolgt in der Regel in den folgenden Schritten:

- Ausarbeitung und Verabschiedung einer „Unternehmensrichtlinie zur Informationssicherheit", die allen Mitarbeitern des Unternehmens bekanntgegeben wird, vgl. 5.3.2
- Einrichtung und Besetzung der Stelle eines unternehmensweiten Informationssicherheitsbeauftragten, der direkt an den Vorstand berichtet, vgl. 5.3.3
- Bereitstellung von Mitteln für den Sicherheitsbeauftragten sowie zur Schaffung einer Sicherheitsorganisation in den Abteilungen bzw. Konzernbestandteilen, vgl. 5.3.3
- Delegation der Aufgaben sowie Zuweisung der Mittel und der notwendigen Befugnisse (zur Ermittlung und Umsetzung der angemessenen Sicherheitsmaßnahmen) an den Sicherheitsbeauftragten und die Mitglieder der Sicherheitsorganisation. Hier finden sich auch die Aufgaben unter 5.2.2 zum risikozentrierten Vorgehen wieder, vgl. 5.4.

Für jeden dieser Schritte gibt es verschiedene Durchführungsmöglichkeiten, deshalb werden diese im Folgenden nur eher oberflächlich dargestellt.

5.3.2 Informationssicherheitsrichtlinie

Mit der Informationssicherheitsrichtlinie (laut ISO 27002, Kap. 5 „Informationssicherheitsleitlinie") drückt das Management eines Unternehmens (oder die Leitung einer anderen Organisation) aus, dass

im Unternehmen keine Informationsverarbeitung ohne angemessene Sicherheit betrieben werden soll. Es handelt sich dabei um eine Absichtserklärung ohne allzu viele Details, die vom Management verabschiedet und allen internen und externen Mitarbeitern bekanntgemacht werden soll.

In der Informationssicherheitsrichtlinie wird grob skizziert,

- die Unterstützung des Managements für die Informationssicherheit, insbesondere die Bereitschaft, die notwendigen Stellen und Mittel dafür zur Verfügung zu stellen,
- dass die Informationssicherheit anhand von Risikoeinschätzungen eingerichtet werden soll,
- was die Ziele und Rahmenbedingungen der Informationssicherheit sind, z.B. gesetzliche Anforderungen, Schutz von Unternehmenseigentum, speziell Informationen und IT-Anlagen,
- die Mitwirkungspflichten jedes Mitarbeiters und evtl. Konsequenzen bei Nichtmitwirkung,
- die Zuständigkeiten, z.B. die Stelle des Informationssicherheitsbeauftragten und weiterer speziell eingerichteter Stellen oder Gremien (Informationssicherheitsorganisation),
- evtl. ein Überblick über die zu übenden Sicherheitsprozesse

Übung 11: Entwickeln Sie die Gliederung einer Sicherheitsrichtlinie für die HS Lu und formulieren Sie ein Kapitel aus.

5.3.3 Informationssicherheitsorganisation

Die Einrichtung einer Informationssicherheitsorganisation in einem Unternehmen soll sicherstellen, dass die nötigen Arbeiten im richtigen Umfang auch tatsächlich durchgeführt werden. Dafür werden die benötigten Mitarbeiter benannt und deren Zuständigkeiten festgelegt.

In der Regel wird zunächst ein Informationssicherheitsbeauftragter für das Unternehmen festgelegt. Je nach Größe des Unternehmens stehen diesem dann weitere Mitarbeiter und Gremien zur Verfügung.

Informationssicherheitsbeauftragter

Der Informationssicherheitsbeauftragte ist zuständig für das Management aller notwendigen Arbeiten zur Erreichung einer angemessenen Informationssicherheit. In kleinen Unternehmen kommen hier noch viele operative IT-Aufgaben, speziell IT-Sicherheitsaufgaben, hinzu. In großen Unternehmen wird der Informationssicherheitsbeauftragte vorwiegend mit dem Management seiner Mitarbeiter beschäftigt sein.

Für den Informationssicherheitsbeauftragten sind, je nach Unternehmen, verschiedene Abkürzungen üblich, z.B. ISB, ISO oder CISO ((chief) information security officer).

Der CISO befindet sich oft in einem Zielkonflikt mit dem IT-Leiter bzw. CIO (chief information officer):

Ziele	**CIO**	**CISO**
Innovationstempo	Hoch: Neue Möglichkeiten und Technologien möglichst schnell zur Verfügung stellen	Angemessen: Erst Sicherheit prüfen und ggfs. sicherstellen
Benutzerfreundlichkeit	Möglichst hohe Zufriedenheit der Benutzer erreichen	Auf keinen Fall Sicherheitsprüfungen weglassen

Um also die Informationssicherheit auf Unternehmensebene sicherzustellen, darf der CISO nicht dem CIO unterstellt sein. Andernfalls besteht die Möglichkeit, dass der CIO dem CISO zu wenig Mittel oder Manpower zur Verfügung stellt.

In der Praxis sollte deshalb der CISO entweder selbst Mitglied des Top Managements sein oder z.B. in einem Bereich Qualität oder Risikomanagement angesiedelt sein.

Weitere Mitarbeiter und Gremien

Speziell in großen Unternehmen kann der CISO nicht sämtliche sicherheitsrelevanten Arbeiten selbst erledigen. In der Regel wird er also mit weiteren Mitarbeitern ausgestattet. Um wichtige oder kostspielige Entscheidungen fundiert und realistisch treffen zu können, ist ein Beirat zur Informationssicherheit sinnvoll. Dieser könnte z.B. die Kompetenz bekommen, die Mittel für größere

Sicherheitsprojekte freizugeben oder diese zumindest gegenüber dem Top Management zu befürworten.

In Konzernen, die aus einer, z.T. großen, Zahl von Tochterunternehmen bestehen, sind in der Regel auch Sicherheitsgremien auf der Ebene der Tochterunternehmen notwendig, die dann durch ein Dachgremium im Mutterkonzern konsolidiert und gesteuert werden. In den Tochterunternehmen werden dann auch oft eigene Informationssicherheitsbeauftragte (ISO) eingesetzt.

Es ergibt sich eine Sicherheitsorganisation als Gremienstruktur, die letztlich durch den CISO gemanaged wird.

Übung 12: Entwerfen Sie eine Sicherheitsorganisation für die Hochschule Ludwigshafen mit all ihren Standorten und Fachbereichen. Formulieren Sie das entsprechende Kapitel in Ihrer Sicherheitsrichtlinie.

5.4 Risikozentriertes Vorgehen

In 5.2 haben wir skizziert, weshalb vollständige IT-Risikoanalysen zu aufwändig und ungenau sind, um darauf das Informationssicherheitsmanagement aufzubauen. Dennoch besteht Einigkeit bei allen Beteiligten und Verantwortlichen, dass „risikozentriert“ vorgegangen werden muss. Das bedeutet, dass die größten Risiken zuerst angegangen werden müssen: „Was bringt mir ein weiteres Schloss an der Haustür, wenn die Kellertür offen steht?“

5.4.1 Identifizierung von Risiken

Es geht also weniger darum, alle Risiken zu identifizieren, als darum, die größten Risiken zu identifizieren, so dass diese behoben werden können. Hierfür gibt es die folgenden Verfahren

- Erfassung sämtlicher Informationswerte (information assets) und Klassifizierung in bestimmte Klassen der Wichtigkeit hinsichtlich Vertraulichkeit, Integrität und Verfügbarkeit. Dabei sollten einzelne Assets nicht ohne Grund als „besonders“ oder „hoch“ schützenswert klassifiziert werden.
- Erfassung bekannter Risiken durch Einbeziehung des IT-Personals, der Benutzer und weiterer Experten, z.B. Berater, Hersteller von Sicherheitsprodukten, öffentliche Stellen, etc..
- Einbeziehung von Risikobetrachtungen bei der Neueinführung von IT-Systemen. Je früher dies im Einführungsprozess erfolgt, umso größer ist der Erfolg und umso kleiner ist der Aufwand.
- Turnusmäßige Risikobetrachtung für „wichtige“ bzw. „besonders schützenswerte“ IT-Systeme.

5.4.2 Risikoprozess

Da alle Unternehmen unterschiedlich strukturiert sind, kann die Erfassung und Behandlung von Risiken unterschiedlich erfolgen. Jedes Unternehmen sollte deshalb seinen spezifischen „Risikoprozess“ festlegen.

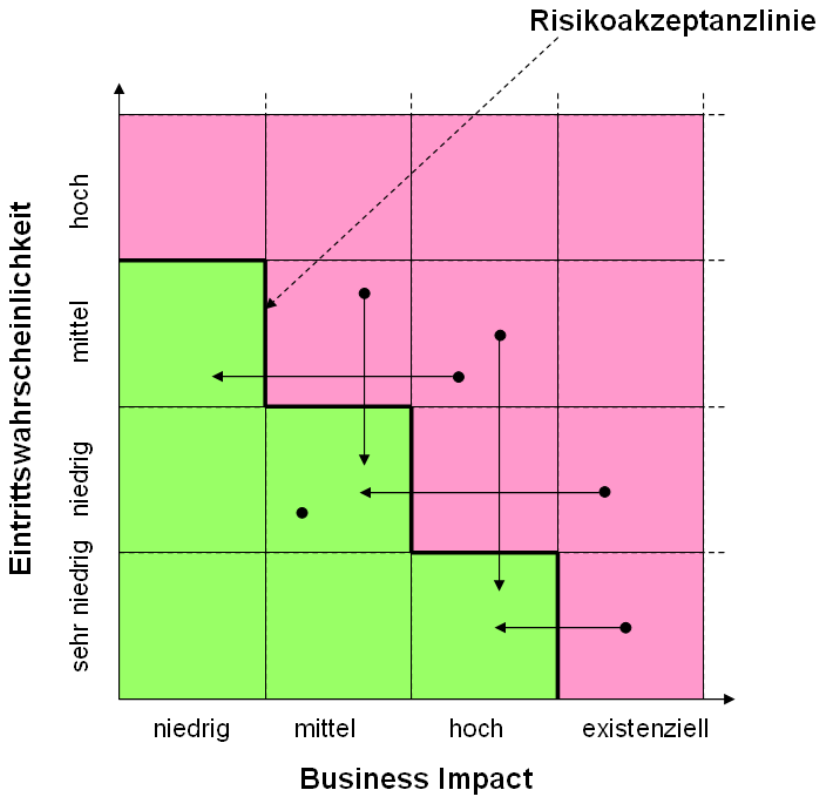
In der Informationssicherheitsrichtlinie sollte der Risikoprozess bereits gefordert oder sogar in Grundzügen beschrieben werden. Der Informationssicherheitsbeauftragte ist für die Durchführung und Weiterentwicklung des Prozesses verantwortlich.

5.4.3 Akzeptierte Risiken

Aufgrund der großen Anzahl unterschiedlicher Detailrisiken können nicht alle Risiken gleichermaßen bearbeitet werden. Risiken, die nicht allzu groß sind, werden in der Regel akzeptiert. Allerdings steht es nicht in der Kompetenz eines einzelnen Mitarbeiters, ohne Weiteres ein Risiko für das Gesamtunternehmen zu akzeptieren. Es müssen also vom Management Vorgaben gemacht werden, wann einzelne Risiken akzeptierbar sind. Dies kann im Zusammenhang mit dem folgenden Beurteilungsschema erfolgen.

5.4.4 Risikobeurteilungsschema

Entsprechend 5.2.2 werden Risiken nach Business Impact, Schadenswahrscheinlichkeit und Aufwand zur Bearbeitung beurteilt. Risiken könnten demnach in ein Schema wie folgt eingeordnet werden:



Die Festlegung der Stufen für Business Impact und Eintrittswahrscheinlichkeit kann jedes Unternehmen spezifisch machen und z.B. für den Business Impact mit EUR-Werten unterlegen (hoch könnte z.B. einen Schaden zwichen EUR 500.000 und EUR 5.000.000 bedeuten).

Die Risikoakzeptanzlinie muss vom Management festgelegt werden.

Risiken unterhalb der Risikoakzeptanzlinie werden als irrelevant akzeptiert und nicht weiter bearbeitet, bis sich entweder die Risikoeinschätzung oder die Risikoakzeptanzlinie ändert.

Risiken oberhalb der Akzeptanzlinie müssen vermindert werden. Dies erfolgt, indem Sicherheitsmaßnahmen ergriffen werden, die entweder die Eintrittswahrscheinlichkeit reduzieren (im Idealfall nahe 0) oder die Auswirkungen eines eventuellen Schadens (Business Impact) verringern.

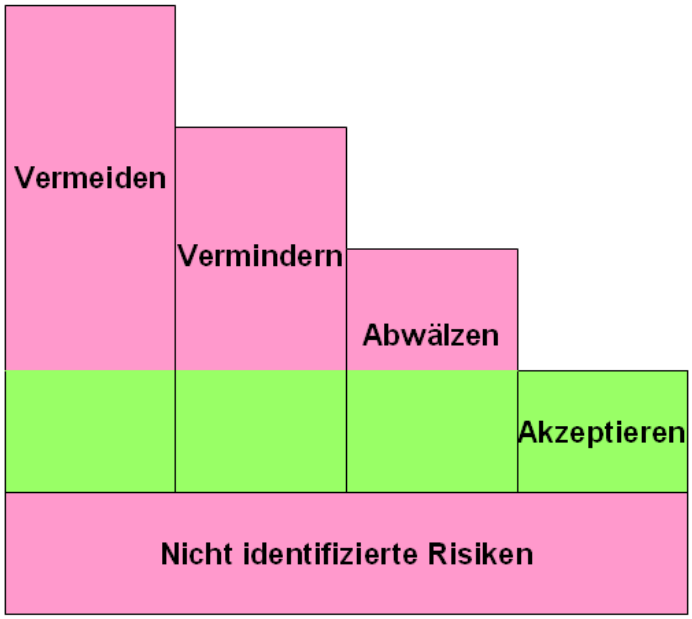
Existenzielle Risiken können in der Regel nicht akzeptiert werden. Dies führt zu der speziellen Aufgabe des Business Continuity Management, vgl. 6.2.

5.4.5 Risikostrategien

Die folgende Darstellung kommt aus dem allgemeinen Risikomanagement, soll Ihnen hier aber nicht vorenthalten werden. Risiken können demzufolge

- durch Maßnahmen vermieden oder
- vermindert werden.
- Alternativ können Risiken durch den Abschluss von Versicherungen abgewälzt werden.
- Risiken, die nicht allzu groß sind können akzeptiert werden.

Problematisch ist allerdings der Bereich der „nicht identifizierten Risiken“, über deren Umfang und Risikohöhe nichts bekannt ist. Der Bereich der nicht identifizierten Risiken sollte demnach möglichst klein gehalten werden.



Übung 13: Entwickeln Sie einen Risikoprozess für die HS Lu. Formulieren Sie das entsprechende Kapitel in Ihrer Sicherheitsrichtlinie.

Übung 14: Führen Sie den Risikoprozess in einem Ihnen bekannten Kontext durch, z.B. im Betrieb eines Bekannten oder Verwandten oder einfach für Ihren Haushalt.

5.5 Informationssicherheitsmanagementsystem (ISMS)

Unter einem Informationssicherheitsmanagementsystem, kurz ISMS, versteht man sämtliche Richtlinien (vgl. 5.3.2), Verantwortlichkeiten (vgl. 5.3.3) und Vorgehensweisen (vgl. 5.4 und 5.6) eines Unternehmens, die dazu dienen, die Informationssicherheit auf das notwendige angemessene Maß zu bringen und dieses dauerhaft einzuhalten.

Das ISMS umfasst also sämtliche Aufgaben im Zusammenhang mit der Informationssicherheit von den Managementaufgaben bis zur Umsetzung und dem Betrieb konkreter (technischer, personeller und organisatorischer) Sicherheitsmaßnahmen.

Zum Aufbau eines ISMS gibt es Sicherheitsnormen auf verschiedenen Ebenen (international, national, konzernintern), von denen die bekannteste die internationale Norm ISO 27001 (und folgende) ist.

5.5.1 Sicherheitsnormen

In einer Sicherheitsnorm werden Vorgaben zum Erreichen von Informationssicherheit zusammengefasst. Ursprünglich (ab ca. 1980) handelte es sich dabei um Maßnahmenkataloge, später zunehmend um Vorgaben zum Sicherheitsmanagement, bei denen die Maßnahmenkataloge nur noch als Anhang mitgeliefert wurden. Die relevantesten Normen sind:

- ISO/IEC 27001 und folgende
- In Deutschland: „IT-Grundschutz“

Weitere Standards sind z.B. (vgl. Wikipedia):

- ISO 13335: Normenreihe zum Sicherheitsmanagement in der Informations- und Kommunikationstechnik
- ISO 27799: Health informatics – Security management in health using ISO 17799 speziell für den Gesundheitsbereich
- Common Criteria for Information Technology Security Evaluation
- Trusted Computer System Evaluation Criteria
- Information Technology Security Evaluation Criteria
- Payment Card Industry Data Security Standard
- die Benchmarks des Center For Internet Security

Auch im ITIL-Standard ist ein Prozess „Information Security Management“ enthalten. Dieser enthält aber nur rudimentäre Aussagen, die in den genannten Normen ausführlicher und aussagekräftiger ausgeführt werden. Deshalb betrachten wir ITIL hier nicht als relevante Norm.

Ein weiterer relevanter Standard ist COBiT, ein Framework zur IT-Governance, das u.a. auch Vorgaben zur Informationssicherheit enthält, die auf Managementebene häufig sehr beachtet werden. Weitere Infos dazu z.B. in Wikipedia.

5.5.2 Zertifizierung

Ein Unternehmen, das seine Informationssicherheit an einer der genannten Normen ausrichtet, kann sich von einem unabhängigen Auditor zertifizieren lassen, um seine Qualität auch nach außen glaubwürdig darstellen zu können.

Übung 15: Für welche Unternehmen oder Unternehmensteile könnte eine Zertifizierung nach einer allgemeinen Norm sinnvoll sein?

5.5.3 ISO/IEC 27001 und folgende

ISO/IEC 27001 ist eine internationale Norm zum Aufbau und Betrieb eines ISMS, die sich weltweit durchgesetzt hat. Sie wird begleitet durch einige ergänzende Normen, insbesondere die ISO 27002, die im Wesentlichen einen Maßnahmenkatalog enthält, der beim Aufbau des ISMS nach ISO 27001 berücksichtigt werden soll. Eine Zertifizierung ist nach ISO 27001 möglich.

Weitere ergänzende Normen sind z.B. die ISO/IEC 27003 (Leitfaden zur Umsetzung der ISO 27001) und ISO/IEC 27004 (Information Security Management Measurement), die aber bisher weder die Qualität noch die Akzeptanz der ISO 27001 erreicht haben. Da alle diese Normen unter dem Nummernkreis ISO 27001 und folgende zusammengefasst wurden, spricht man von der Normenfamilie ISO 2700x.

Historisch ist der Maßnahmenkatalog die ältere der Normen. Dieser erschien ursprünglich 1995 als BS (British Standard) 7799, während die Norm zum Aufbau des ISMS erst 1999 als BS 7799, Part 2 erschienen ist. Zur Klärung der Bezeichnungen hier der historische Entwicklungspfad:

- BS 7799-2 (1998) → ISO 27001 (2005)
- BS 7799-1 (1995) → ISO 17799 (2000) → ISO 27002 (2007)

Seit 2008 gibt es beide Normen auch offiziell in deutsch als DIN ISO/IEC 27001 und DIN ISO/IEC 27002.

In Kapitel 6 werden wir die internationale Norm ISO/IEC 27001 noch näher betrachten.

5.5.4 In Deutschland: „IT-Grundschutz“

In Deutschland hat das „Bundesamt für Sicherheit in der Informationstechnologie“ seit den 1990 Jahren die sogenannten „Grundschutzkataloge“ erarbeiten lassen, in denen jeweils für einen bestimmten Typ von IT-Systemen (z.B. IBM-Großrechner, Windows Rechner, Netzwerke, etc.) die Gefährdungen und möglichen Sicherheitsmaßnahmen zusammengestellt wurden. Es handelt sich also im Wesentlichen um Maßnahmenkataloge.

Mit der Verschiebung des Fokus in Richtung Sicherheitsmanagement und ISMS wurden auch Grundschutzkataloge für Sicherheitsmanagement und Risikomanagement entwickelt. Außerdem wurden diese an die internationalen Normen angeglichen, da ein deutscher Alleingang nicht mehr sinnvoll war.

Die Maßnahmenkataloge nach IT-Grundschutz sind viel detaillierter als die entsprechenden Maßnahmenkataloge der ISO 27002. Damit sind die Grundschutzkataloge schwieriger umzusetzen und bieten auch weniger Freiheitsgrade, können aber sehr gut als Nachschlagewerke dienen.

Heute ist eine Zertifizierung „ISO/IEC 27001-Zertifizierung auf der Basis von IT-Grundschutz“ möglich, die aber bei internationalen Unternehmen aus dem genannten Grund wenig Anklang findet.

Übung 16: Welche Gründe könnte ein Unternehmen haben, sich in Bezug auf ISO 27001 mit IT-Grundschutz zertifizieren zu lassen.

5.6 Informationssicherheitsprozesse

Um klarzumachen, dass Informationssicherheit nicht durch eine einmalige Anstrengung erreicht werden kann, sondern kontinuierlich verfolgt werden muss, wird im Informationssicherheitsmanagement zunehmend der Prozesscharakter der Sicherheitsaufgaben hervorgehoben. Spezielle Beispiele dafür sind:

- Plan – Do – Check – Act – Kreislauf im Sicherheitsmanagement bzw. im ISMS. Grundsätzlich bedeutet dies, dass jede Aktion turnusmäßig überprüft und ggfs. verbessert werden sollte.
- Der Risikomanagementprozess erfordert regelmäßige Wiederholungen der Risikoerfassung und –reduktion.
- Berechtigungsmanagement ist der Prozess der Beantragung, Genehmigung, Vergabe und schließlich dem Entzug von System-Berechtigungen (speziell Zugriff und Benutzung).
- Incident Management ist der Prozess der Erfassung und Bearbeitung von Sicherheitsvorfällen.
- Business Continunity Management ist der Prozess zur Sicherstellung des Geschäftsbetriebs.
- Information Security Management als Prozess im ITIL,
- Sicherheitsbetrachtungen als Bestandteil von Entwicklungs- und Change Prozessen,
- Release und Patch Management um sicherzustellen, dass immer Systemversionen mit möglichst wenig Sicherheitslücken eingesetzt werden.

5.7 Messbarkeit der Informationssicherheit

In 2.3 haben wir aufgeführt, dass Informationssicherheit eine Produkteigenschaft bzw. Qualität ist, die immer nur zu einem bestimmten Umfang gegeben ist. Es liegt deshalb nahe, wissen zu wollen, „wie groß“ unsere Informationssicherheit tatsächlich ist. Dies wäre speziell zur Unterstützung des Managements hinsichtlich Entscheidungen zur Informationssicherheit wünschenswert. Dies führt zu der Frage nach geeigneten Kennzahlen, also der Messbarkeit der Informationssicherheit.

5.7.1 Schwierigkeiten

Dabei treffen wir aber auf dieselben Schwierigkeiten wie schon im Zusammenhang mit den Risikobetrachtungen:

- Es gibt viele verschiedene Systeme, die alle einen unterschiedlichen Grad der Sicherheit haben können.
- Zur Quantisierung der Sicherheit müssten wieder Business Impact und Eintrittswahrscheinlichkeit von möglichen Sicherheitsvorfällen betrachtet werden, die aber nur sehr schwer zu quantisieren sind (vgl. 5.2.3).
- Für jedes der zu betrachtenden Details könnte evtl. eine Sicherheitskennzahl festgelegt werden, dies würde aber eine unübersichtliche Menge von Kennzahlen produzieren, woraus kein betrieblicher Nutzen folgt.

5.7.2 Aggregation

Da die große Menge von Detailkennzahlen für das Management keine gute Entscheidungshilfe darstellt, müssten diese aggregiert werden. Hierfür gibt es wiederum viele (mathematischen) Möglichkeiten, z.B.:

- Addition der Risiken,
- Angabe des gefährlichsten Risikoszenarios (maximales Risiko),
- Anzahl von Risiken, die einen vorgegebenen Schwellenwert überschreiten

Sobald aber mathematische Operationen auf die Risiken angewendet werden, verstellt dies den Blick auf die Schätzungenauigkeiten hinter den einzelnen Zahlen, deshalb ist auch eine solche Aggregation gefährlich.

Außerdem fehlt evtl. ein Maßstab für das Sicherheitsziel, die „Angemessenheit“.

5.7.3 Prozesskennzahlen

Die Sicht auf Informationssicherheitsmanagement als eine Sammlung von Prozessen (vgl. 5.6) ermöglicht auch die Entwicklung von Key Performance Indicators (KPI) zur Messung der Qualität der

Sicherheitsprozesse. In diesem Sinne könnte ein Kennzahlensystem nicht auf den Stand der Sicherheit bzw. der Risiken sondern auf die Qualität der Sicherheitsprozesse, also den Umgang mit Risiken, abzielen.

Auch hier entstehen allerdings wieder viele Detailkennzahlen. Außerdem ist es noch nicht gelungen, sämtliche Aufgaben der Informationssicherheit, z.B. entsprechend dem Maßnahmenkatalog der ISO 27002, als Sicherheitsprozesse zu formulieren.

Übung 17: Überlegen Sie sich Kennzahlen für die Informationssicherheit der HS Lu.

6 ISMS nach ISO/IEC 2700x

6.1 ISO 27001

→ PDF-Datei von Emanuel Hein.

6.2 ISO 27002

Der Standard ISO 27002 bezeichnet sich als „Code of Practice“ bzw. „Leitfaden“ für das Informationssicherheitsmanagement. Er wird im ISO 27001 als Anhang aufgeführt und muss für eine Zertifizierung zwingend vollständig berücksichtigt werden.

ISO 27002 enthält Informationen über zahlreiche Sicherheitsmaßnahmen in einer gewissen Detaillierungsstufe. In der Version von 2013 ist der Standard gegliedert in 19 Kapitel (incl. der 5 einleitenden Kapitel 0 - 4). Ab Kap. 5 werden Sicherheitsmaßnahmen beschrieben, so genannte „Controls“ zu je einer der folgenden „Sicherheitskategorien“. Das Eintragen der deutschen Ausdrücke wird als Übungsaufgabe empfohlen.

	deutsch	**englisch**
Kap. 5		Information Security Policies
Kap. 6		Organization of information security
Kap. 7		Human resource security
Kap. 8		Asset management
Kap. 9		Access Control
Kap. 10		Cryptography
Kap. 11		Physical and Environmental Security
Kap. 13		Communications Security
Kap. 14		System acquisition, development and maintenance
Kap. 15		Supplier relationships
Kap. 16		Information security incident management
Kap. 17		Information security aspects of business continuity management
Kap. 18		Compliance

Die angegebenen Maßnahmen werden in der Regel nicht systemspezifisch formuliert, sondern sind für jedes IT-System neu zu interpretieren. Dabei ist zunächst aus Risikoüberlegungen heraus zu beurteilen, ob die jeweilige Maßnahme im gegebenen Kontext überhaupt relevant ist. Anschließend ist die konkrete, angemessene Umsetzung zu planen, durchzuführen und zu dokumentieren.

Zur Umsetzung gehört dabei in der Regel auch eine Anleitung, wie die Sicherheitsmaßnahmen zu administrieren ist und wie sich beteilte Benutzer oder Administratoren zu verhalten haben. Evtl. kann auch eine Benutzerschulung dazu gehören.

Details zur Umsetzung finden sich in der ISO 27002 nur bis zu einem gewissen Grad, solche können aber zum Beispiel in den Grundschutzkatalogen (vgl. 5.5.4) gefunden werden.

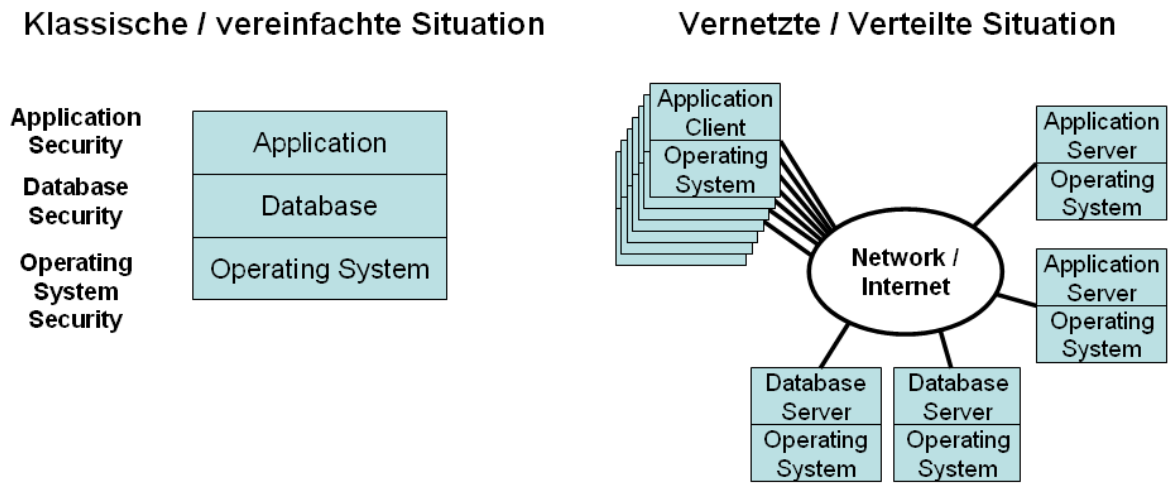
7 Applikatorische Sicherheit

An dieser Stelle beschließen wir unsere Ausführungen zum Informationssicherheitsmanagement und gehen über zum Themenkomplex der applikatorischen Sicherheit und speziell der Webanwendungen.

Wir werden sehen, dass die Sicherheitsanforderungen für Webanwendungen einerseits die Entwickler und andererseits die Betreiber der Anwendung betreffen. Diese Trennung ist aber keineswegs neu, sondern betrifft Softwareentwicklung und –betrieb in jedem Kontext. Wir betrachten deshalb zunächst die allgemeine Situation.

7.1 Komplexität

IT-Systeme entwickeln sich zu immer komplexeren Infrastrukturen. Die wesentlichen Aufgaben der Informationssicherheit bleiben dennoch dieselben, die Umsetzung erfordert allerdings mehr Blick für das Gesamtbild.



7.1.1 Einfache Situation

In der einfachsten Situation, z.B. bei Großrechnern der 1980er Jahre läuft eine Anwendung auf genau einem Rechner mit genau einem Betriebssystem und speichert ihre Daten in einer Datenbank auf demselben Rechner. Allerdings laufen auf demselben Rechner viele Anwendungen, so dass z.T. mehrere tausend Benutzer auf diesen Rechner Zugriff haben müssen.

7.1.2 Vernetzte Situation

In der vernetzten Situation kommt hinzu, dass Anwendungen aus mehreren Bestandteilen bestehen können, z.B. Client- und Server-Komponenten, die jeweils auf eigenen Rechnern mit eigenen Betriebssystemen laufen können. Auch die Datenbank kann auf einem oder mehreren Rechnern mit eigenen Betriebssystemen laufen.

Der größte Unterschied ist aber, dass durch eine Anbindung an das Internet aus den vielen tausend potenziellen Angreifern nun mehrere Milliarden potenzielle Angreifer werden und dass die Datenverbindungen im Zusammenhang mit unserer Anwendung nicht mehr im eigenen Haus verlaufen, sondern über mehrere Provider / Dienstleister, deren Sicherheit und Vertrauenswürdigkeit wir in der Regel nicht einschätzen können.

Es entstehen deshalb neue Sicherheitsanforderungen im Hinblick auf Angreifer, die Zugriff auf das Netzwerk haben.

7.2 Aufgaben der applikatorischen Sicherheit

7.2.1 Grundaufgaben

Die klassischen Aufgaben der applikatorischen Sicherheit sind

- Authentisierung – wer darf auf die Anwendung zugreifen?
- Autorisierung bzw. Zugriffsschutz – wer darf was mit der Anwendung tun (welche Funktionen nutzen)?
- Protokollierung – wer hat was getan (englisch logging oder auditing)?

Eine weitere Sicherheitsaufgabe ist die

- Verhinderung ungeplanter Zugriffsmöglichkeiten.

Mit Authentisierung und Zugriffsschutz ist zu gewährleisten, dass die von einer Applikation zur Verfügung gestellten Funktionen nur den dafür berechtigten Personen zugänglich sind. Mit Protokollierung ist zu gewährleisten, dass es entdeckt wird, falls ein berechtigter Benutzer seine Berechtigungen missbraucht.

Übung 18: Überlegen Sie sich ein Beispiel für den Fall wo ein berechtigter Benutzer seine Berechtigungen missbraucht.

Zusätzlich zu den geplanten Funktionen der Anwendung können ungeplante Zugriffsmöglichkeiten vorhanden sein, die evtl. nicht der Authentisierung, dem Zugriffsschutz und der Protokollierung unterliegen. Deshalb müssen solche Mechanismen möglichst vollständig verhindert werden. In diesen Zusammenhang fallen z.B. Viren, Trojaner, missbräuchliche Eingaben, etc. Für Webanwendungen werden wir diese noch in einigem Detail besprechen.

7.2.2 Datenbanksicherheit

Da eine Datenbank (genauer Datenbankmanagementsystem DBMS) selbst eine Anwendung ist, sollte auch diese über Authentisierung, Autorisierung und Protokollierung verfügen.

Wenn eine Anwendung auf einem DBMS aufsetzt und darin ihre Daten speichert, ist die Anwendung in der Regel nicht sicherer als die Datenbank, denn ein Angreifer, der Zugriff auf die Datenbank hat, kann die Daten dort auslesen, ohne sich an der Anwendung anzumelden.

Falls auch die Anmeldedaten der Anwendung in der Datenbank stehen, kann ein Angreifer diese entweder ebenfalls auslesen (Verletzung der Vertraulichkeit der Datenbank) und sich damit an der Anwendung anmelden. Alternativ könnte er einfach neue Anmeldedaten erfinden, in der Datenbank eintragen (Verletzung der Integrität der Datenbank) und sich damit an der Anwendung anmelden.

Mögliche zusätzliche Schutzmaßnahmen, die bereits in der Anwendung vorgenommen werden können, wären deshalb

- Verschlüsselte Speicherung der Daten
- Verschlüsselte Speicherung der Anmeldedaten (Schutz der Vertraulichkeit)
- Digitale Signatur der Anmeldedaten (Schutz der Integrität)

Übung 19: Wie verwalten Sie die Schlüssel, wenn die Datenbank unsicher ist?

7.2.3 Betriebssystemsicherheit

Da sowohl die Anwendung als auch die Datenbank auf einem Betriebssystem installiert sind, ist auch die Sicherheit dieses Betriebssystems grundlegend für sowohl die Datenbank als auch die Anwendung. Falls ein Angreifer Zugriff auf das Betriebssystem hat, kann er z.B. wie in 7.2.2 Daten, speziell Anmeldedaten, auslesen oder manipulieren.

Hinzu kommt hier noch die Möglichkeit eines Angreifers, die Softwarekonfiguration zu manipulieren oder ganze Softwaremodule auszutauschen, so dass z.B. die Authentisierung immer das Ergebnis „erlaubt“ liefert.

Mögliche zusätzliche Schutzmaßnahme der Anwendung wäre

- Digitale Signatur von Programmcode und Konfigurationsdateien

Übung 20: Wie verwalten Sie die Schlüssel, wenn Datenbank und Betriebssystem unsicher sind?

7.2.4 Sicherer Betrieb

Selbst wenn eine Anwendung die genannten Maßnahmen, z.B. Authentisierung und Autorisierung korrekt und vollständig realisiert hat, können im Betrieb Sicherheitslücken entstehen, wenn diese Maßnahmen unsicher genutzt werden. Z.B. können zu kurze Passwörter zu einer unsicheren Authentisierung führen oder eine Autorisierung ist nutzlos, wenn der Betreiber einfach allen Benutzern die vollen Zugriffsrechte einräumt.

7.3 Abgrenzung zur Systemsicherheit

7.3.1 Allgemein

Zur applikatorischen Sicherheit zählen wir alle Aufgaben und Maßnahmen, die Softwareentwickler realisieren können, damit die Software sicher betrieben werden kann.

Ob die Software dann tatsächlich sicher betrieben wird und ob die zugrundeliegenden Netzwerke, Datenbanken und Betriebssysteme sicher genug sind, zählen wir nicht zum Bereich der applikatorischen Sicherheit.

7.3.2 Websicherheit

Insbesondere für Webanwendungen müssen wir davon ausgehen, dass ein Angreifer Zugriff auf die Netzwerkverbindung hat und Netzwerkpakete in jeder Form direkt an unsere Anwendung schicken kann.

Maßnahmen der Netzwerksicherheit wie Firewalls sind in diesem Zusammenhang wünschenswert, wir zählen diese aber nicht zur applikatorischen Sicherheit.

7.3.3 Vergleich

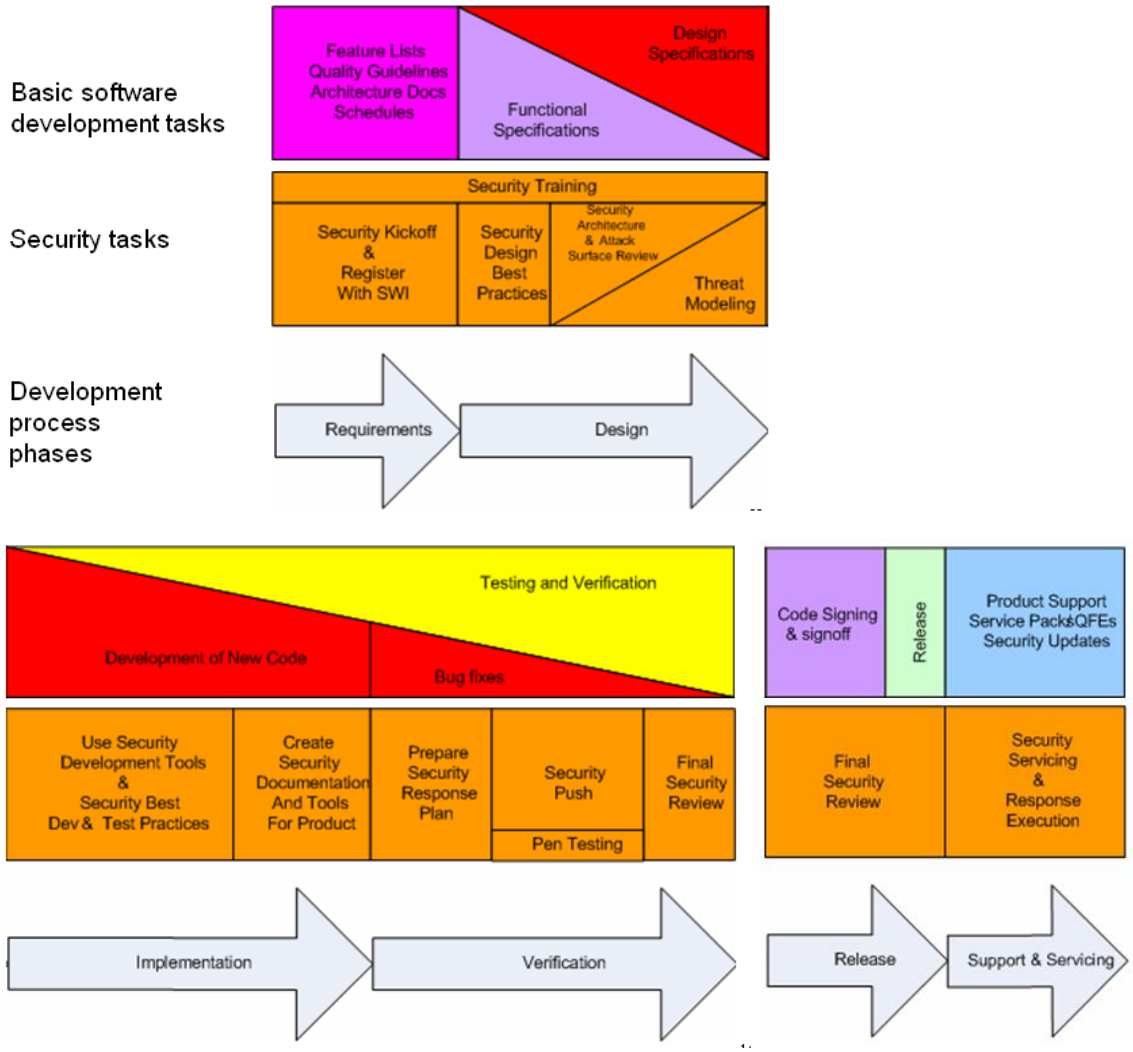
Die folgende Tabelle enthält einen Vergleich zwischen den applikatorischen Sicherheitsmaßnahmen wie sie einerseits von Microsoft und andererseits vom OWASP (Open Web Application Security Project, vgl. 8.1.1) empfohlen werden. Es ist erkennbar, dass der Unterschied nicht besonders groß ist:

Microsoft	**OWASP**
Compartmentalize	Compartmentalization (Separation of Privileges)
Use least privilege	Least Privilege
Apply defense in depth	Defense in depth
Check at the gate	
Do not trust user input	**Validate input and output**
Fail securely	Fail securely (closed)
	Keep it simple
	Use and reuse trusted components
Secure the weakest link	Only as secure as the weakest link
	Security by obscurity won‘t work
Create secure defaults	
Reduce your attack surface	

7.4 Sicherer Softwarentwicklungsprozess

Zur Entwicklung sicherer Software gehört ein Entwicklungsprozess, der sicherstellt, dass die relevanten Risiken frühzeitig erkannt werden und entsprechende Maßnahmen bereits in der Konzeptphase in die Entwicklung einfließen.

Da dies einen Zusatzaufwand gegenüber „unsicherer Softwareentwicklung“ bedeutet, ist es ganz wichtig, dass das Management und alle beteiligten Mitarbeiter dies mittragen („Security Awareness“). Die folgenden Folien zeigen Microsoft’s Entwurf eines „Trustworthy Computing Security Development Lifecycle“:



7.5 Authentisierung

Authentisierung ist nicht irgendeine Sicherheitsmaßnahme, sondern in gewissem Sinne die allererste. Mit Authentisierung wird die Identität eines Benutzers festgestellt.

7.5.1 Vertrauenskette

Nach der Authentisierung sind alle weiteren Entscheidungen über Zugriffsberechtigungen von dieser festgestellten Identität abhängig zu machen. Mit der Authentisierung beginnt die so genannte Vertrauenskette (chain of trust). Eine Kette ist nur so stark wie ihr schwächstes Glied. Hier heißt das: Keine spätere Zugriffsentscheidung kann besser sein, als die vorausgegangene Authentisierung.

Auch Protokollierungsmaßnahmen hängen direkt von der Authentisierung ab: Wenn der Benutzer nicht richtig identifiziert wurde, können die Benutzereinträge in allen Protokollen falsch sein.

7.5.2 Authentisierungsverfahren

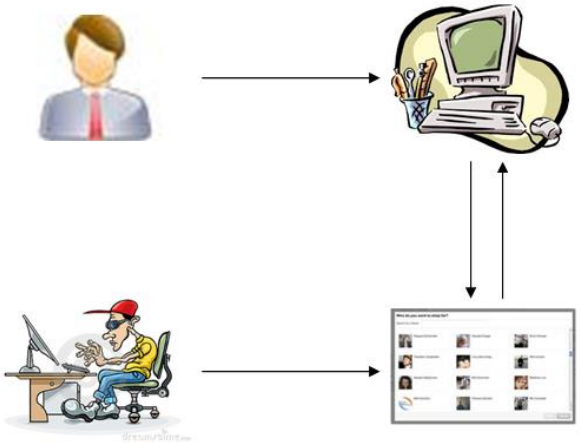
Authentisierung ist keineswegs neu! Lang bevor es Computer gab, war Authentisierung das wichtigste Sicherheitsverfahren an der mittelalterlichen Stadtmauer.

Übung 21: Nennen Sie mindestens 3 Authentisierungsverfahren ohne Einsatz von Computern.

Zu einem Authentisierungsverfahren gehören verschiedene Komponenten:

- Der Benutzer, der sich authentisieren möchte,
- die authentifizierende Stelle,
- eine Sammlung (z.B. eine Liste oder ein Verzeichnis) von Benutzern, die authentifiziert werden dürfen,

- ein Verfahren zur Pflege dieser Liste (bzw. des Verzeichnisses)



7.5.3 Arten von Authentisierungsverfahren

Übung 22: Nennen Sie mindestens 3 Authentisierungsverfahren mit Einsatz von Computern.

Einige Authentisierungsverfahren sind z.B.

- Verwendung von Userids (Zugangskennungen) und Passwörtern (sehr weit verbreitet)
- Benutzung von Hardware-Tokens zur Generierung von Einmalpasswörtern
- Benutzung von Smartcards, die quasi als „elektronischer Ausweis“ dienen sollen.
- Authentisierung anhand biometrischer Merkmale, z.B. Fingerabdruck, Auge (Iris), etc.

Verallgemeinert spricht man von

- Authentisierung auf der Basis von Wissen (z.B. Passwörter),
- Authentisierung auf der Basis von Besitz (z.B. Hardware-Tokens oder Smartcards)
- Authentisierung auf der Basis von Eigenschaften (z.B. Biometrie)

7.5.4 Angriffe auf Authentisierungsverfahren

Um ein Authentisierungsverfahren anzugreifen, kann jede der oben beschriebenen Komponenten angegriffen werden, z.B.:

- Der Benutzer könnte vorgeben, ein anderer zu sein, z.B. mit einem geklauten Passwort.
- Der authentifizierende Rechner könnte so manipuliert werden, dass er den Angreifer fälschlicherweise als berechtigten Benutzer akzeptiert.
- Der Administrator könnte bestochen werden, damit er dem Angreifer die Zugriffsberechtigung einräumt.
- Das Verzeichnis von Zugriffsberechtigungen könnte
  - gelesen werden (Vertraulichkeitsverlust), so dass einem Angreifer z.B. Passwörter in die Hände fallen.
  - manipuliert werden (Integritätsverlust), so dass ein Angreifer z.B. Userid / Passwort einschleusen kann, mit denen er sich später anmelden kann.
  - sabotiert / zerstört werden (Verfügbarkeitsverlust), so dass der Betreiber des Systems beschließt, das System vorübergehend ohne Authentisierung zu betreiben.

Darüberhinaus können die Übertragungswege (oben dargestellt als Pfeile) angegriffen (abgehört oder manipuliert) werden.

Übung 23: Nennen Sie zu jeder der Komponenten und zu jedem Übertragungsweg mindestens 1 Angriffsmöglichkeit.

Übung 24: Nennen Sie zur Authentisierung durch Fingerabdruck mögliche Schwachstellen.

7.5.5 Stärke von Authentisierungsverfahren

Wenn man vollen Zugriff auf sämtliche beteiligten Systeme hat, ist jedes Authentisierungsverfahren zu knacken. Es könnte dann ja z.B. einfach eine Smartcard für den Angreifer ausgestellt werden. Dies entspricht im wirklichen Leben etwa einem Spion im Kanzleramt.

Vor diesem Hintergrund besagt die Stärke eines Authentisierungsverfahrens, wie aufwändig es für einen Angreifer ist, das Verfahren zu umgehen. Passwort-Verfahren gelten dabei als ziemlich unsicher, da es viele einfache Schwachstellen gibt:

- Passwörter können evtl. erraten werden.
- Passwörter werden oft aufgeschrieben (damit sie nicht vergessen werden).
- Passwörter werden oft weitergesagt (freiwillig oder mit Erpressung oder Gewalt).
- Passwörter können evtl. im Netz abgehört werden.
- Passwörter können evtl. aus einer Datenbank gelesen werden.
- Die Prozesse zur Passwort-Vergabe (Benutzer-Registrierung), zum Passwort-Reset oder zur Löschung von Benutzerkonten könnten unsicher sein.

Übung 25: Nennen Sie zu jeder der angegebenen Schwachstellen mindestens eine Sicherheitsmaßnahme, die diese beheben kann.

Bei einer Authentisierung auf der Basis von Besitz besteht die einfache Möglichkeit:

- Das Hardware-Token könnte gestohlen (oder verloren und gefunden) werden.

Deshalb werden Verfahren, die auf Besitz beruhen, meistens noch mit einem Passwortverfahren kombiniert, dies gilt als ziemlich sicher.

Die Stärke eines Authentisierungsverfahrens spielt sich nicht nur zwischen Benutzer und System ab. Vielmehr ist der Administrationsprozess genauso wichtig, wenn nicht noch wichtiger! Grund: Wenn ein Angreifer sich als Administrator ausgeben kann, dann kann er sich beliebig viele Passwörter selbst einrichten. Allerdings gibt es sehr viel weniger Administratoren als Endbenutzer, so dass diese mit weniger Aufwand geschützt werden können. Z.B. Besitz und Wissen für Administratoren, Passwörter mit vorgegebener Stärke für Benutzer.

7.5.6 Kollektive Zugangsberechtigungen

Bei Authentisierungsverfahren muss jeder berechtigte Benutzer im System eindeutig bekannt sein. In weniger sicherheitsrelevanten Fällen ist es möglich, dass ein Zugangscode zur Verfügung gestellt wird, den sich alle berechtigten Benutzer teilen. Ein Beispiel ist der Zugang zu meinen Unterlagen für alle Studierende der HS Lu. Es geht hierbei also nur um den Ausschluss der großen Öffentlichkeit.

Ein solches Verfahren ist natürlich nicht sehr sicher, besitzt aber den Vorzug, dass es wesentlich einfacher zu administrieren ist. Unter der Beachtung der Angemessenheit könnte dies aber der optimale Schutz in weniger sicherheitskritischen Fällen sein.

7.5.7 Authentisierungsverfahren im ISO 27002

Der Standard ISO 27002 stellt Anforderungen an das Vorhandensein und die Qualität von Authentisierungsverfahren in den folgenden Kapiteln:

- Kap. 6.2: Organisation → Vereinbarungen und Maßnahmen für Zugang externer Personen (Dienstleister, Kunden, Weitere)
- Kap. 8.3.3: Personalsicherheit → Zurücknahme von Zugriffsrechten
- Kap. 10.6.2: Kommunikationsmanagement → Sicherheit von Netzdiensten
- Kap. 11: Zugangskontrolle

Übung 26: Analysieren Sie Kap. 11 der ISO 27002. Welche Unterabschnitte beziehen sich konkret auf Authentisierungsverfahren? Welche beziehen sich auf Passwortverfahren?

7.5.8 Pseudoauthentisierung

Gelegentlich werden Sicherheitsmaßnahmen eingeführt, wonach ein Benutzer sich nicht persönlich identifizieren muss, sondern z.B.

- eine Userid / Passwort –Kombination von mehreren Personen genutzt wird (Gruppen-Login),

- gar keine Userid verwendet wird, sondern nur ein Zugangscode bekannt sein muss,
- der Zugang gewährt wird, wenn der Zugriff aus einem bestimmten Netzwerksegment heraus erfolgt.

Bei solchen Verfahren handelt es sich nicht um Authentisierung, weil die Identität des Benutzers nicht festgestellt wird. Das größte Sicherheitsproblem dabei ist, dass eine Aktion nicht mehr nachträglich einem Benutzer zugeordnet werden kann wodurch die Hemmschwelle für einen Angreifer deutlich gesenkt wird.

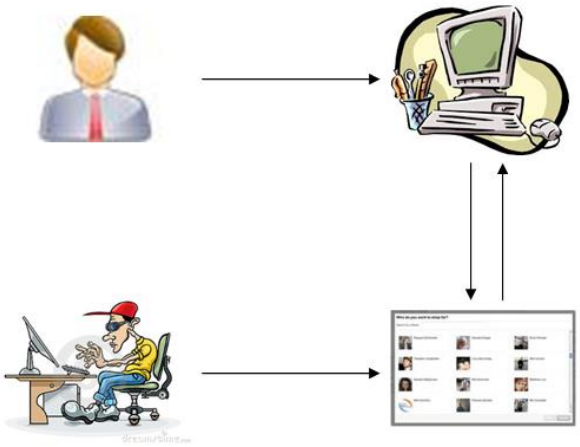
7.6 Zugriffsschutz

Bei kleineren Anwendungen oder im Fall kollektiver Zugangsberechtigungen (vgl. 7.5.6) ist es möglich, dass jeder zugangsberechtigte (authentisierte) Anwender im System dieselben Rechte besitzt.

Für Betriebssysteme und umfangreichere Anwendungen wie z.B. SAP besitzen unterschiedliche Benutzer auch unterschiedliche interne Berechtigungen zur Durchführung von Funktionen und Transaktionen. Dies erfordert Zugriffsschutzverfahren innerhalb der Anwendung bzw. des Betriebssystems. Wir sprechen von jetzt an nur noch von „Anwendungen“, für Betriebssysteme gilt aber dasselbe.

7.6.1 Schematische Darstellung

Grundsätzlich besitzt Zugriffsschutz dieselbe Architektur wie die Authentifizierung. Deshalb werden in Anwendungen beide Verfahren meistens gemeinsam behandelt.



Die Komponenten sind dieselben wie in 7.5.2:

- Ein Benutzer versucht, eine Funktion der Anwendung zu nutzen.
- Die Anwendung muss entscheiden, ob der Benutzer die nötigen Berechtigungen besitzt.
- Die Berechtigungen werden in einem Verzeichnis hinterlegt.
- Ein Administrator verwaltet die Berechtigungen in dem Verzeichnis.

Da für den Zugriffsschutz dieselben Benutzer berücksichtigt werden müssen wie für die Authentisierung, wird hier in der Regel auf dieselben Daten zugegriffen.

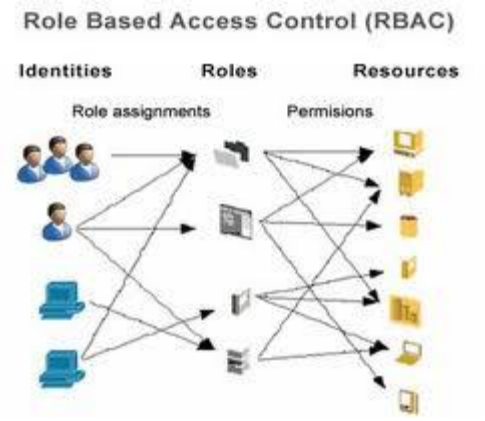
7.6.2 Arten von Zugriffsschutzverfahren

Der wesentliche Unterschied verschiedener Zugriffsschutzverfahren liegt darin, wie die Berechtigungen im Verzeichnis abgelegt werden. Davon abhängig sind die programmatische Umsetzung in der Anwendung und die Administration. Der Benutzer selbst hat keine spezifischen Aufgaben, er versucht nur, seine Transaktion auszuführen.

Im Verzeichnis der Berechtigungen müssen sowohl die zugreifenden Benutzer bekannt sein, als auch die Ressourcen, auf die zugegriffen wird. Jeder Benutzer soll nur auf die Ressourcen zugreifen dürfen, die er für seine Arbeit im Unternehmen benötigt. Es gibt drei Strukturen:

- Discretionary access control – Benutzern werden die benötigten Ressourcen direkt zugewiesen, z.B. mit access control lists (ACL)

- Mandatory access control – die Berechtigungen werden regelbasiert geprüft. Z.B. kann sowohl den Benutzern als auch den Ressourcen eine Schutzstufe zugeordnet werden und ein Zugriff ist erlaubt, wenn die Schutzstufe des Benutzers höher ist als die der Ressource. Dieser Mechanismus kommt ursprünglich aus dem militärischen Bereich.
- Role-based access control – es werden Rollen eingeführt, die bestimmten Tätigkeiten im Unternehmen entsprechen. Eine Rolle enthält die Ressourcen, die für die Tätigkeit notwendig sind und wird den Benutzern zugewiesen, die diese Tätigkeit durchführen sollen.



Übung 27: Wieviele direkte Verbindungen zwischen Benutzern und Ressourcen müssten administriert werden, wenn die im Bild dargestellte Situation mit discretionary access control abgebildet werden sollte?

Hauptprobleme bei der Administration des Zugriffsschutz sind:
- Sehr große Anzahl von Benutzern und Ressourcen
- Ähnliche Berechtigungen in unterschiedlichen Systemen und Anwendungen
- Häufige Änderungen
- Entzug von Berechtigungen, die nicht mehr gebraucht werden

7.6.3 Stärke von Zugriffsschutzverfahren

Die Stärke eines Zugriffsschutzverfahrens besteht darin, dass die Berechtigungen möglichst korrekt sind und möglichst zeitnah angepasst und entzogen werden. Dies ist kein technisches Problem, sondern ergibt sich aus der Qualität des Administrationsprozesses.

Wie im Fall der Authentisierung gehört hierzu auch die Sicherheit der Administratorkonten, vgl. 7.5.5.

7.6.4 Zugriffsschutz im ISO 27002

Der Standard ISO 27002 stellt Anforderungen an das Vorhandensein und die Qualität von Zugriffsschutzverfahren in den Kapiteln:
- Kap. 6.2: Organisation → Vereinbarungen und Maßnahmen für Zugang externer Personen (Dienstleister, Kunden, Weitere)
- Kap. 11: Zugangskontrolle

Übung 28: Analysieren Sie Kap. 11 der ISO 27002. Welche Unterabschnitte beziehen sich konkret auf Zugriffsschutzverfahren?

Anm.: Diverse Artikel zum Role-Based access control finden sich unter http://www.roeckle.info/frm_u.htm.

7.7 Protokollierung

Protokollierung (englisch: Logging) kann zwar keine Angriffe verhindern, kann aber ermöglichen, nachträglich herauszufinden, wie ein Angriff passiert ist und kann damit evtl. helfen, den eintretenden Schaden zu vermindern.

7.7.1 Anwendung und Administration

Anwendungen können z.B. protokollieren, wenn sich ein Benutzer anmeldet (authentifiziert) und welche Aktionen er in der Anwendung ausführt.

Genauso wichtig ist die Protokollierung administrativer Aktionen. Falls ein Benutzerkonto nur für den Zweck eines Angriffs angelegt wird, soll nachprüfbar sein, welcher Administrator dies getan hat.

7.7.2 Stärke von Protokollierungsverfahren

Ein Protokollierungsverfahren kann die folgenden Schwächen aufweisen:

- Ein Angreifer könnte das Verfahren abstellen. Wenn z.B. die Protokollierung durch eine Konfigurationsdatei gesteuert wird, könnte ein Angreifer diese Konfigurationsdatei manipulieren.
- Die Protokolldatensätze werden evtl. nicht lange genug gespeichert, da diese ein großes Datenvolumen annehmen können. Für sicherheitskritische Protokolle wird die Verwendung von nicht-überschreibbaren Medien empfohlen (WORM: write-once-read-multiple)

7.7.3 Auswertung

Anwendungsprotokolle können in zweierlei Situationen ausgewertet werden:

- im Fall eines erkannten Angriffs,
- regelmäßig, um Angriffsmuster zu erkennen.

Eine regelmäßige Auswertung fällt in den Bereich der Systemüberprüfung (Audit) und erfordert Richtlinien, Prozesse und Verantwortlichkeiten.

7.7.4 Protokollierung im ISO 27002

Der Standard ISO 27002 stellt Anforderungen an das Vorhandensein und die Qualität von Protokollierungsverfahren in den Kapiteln:

- 10.6.1: Kommunikationsmanagement → Maßnahmen für Netze
- 10.10: Kommunikationsmanagement → Überwachung, bestehend aus 10.10.1 Auditprotokolle, 10.10.3 Schutz von Protokollinformationen, 10.10.4 Administrator- und Betreiberprotokolle, 10.10.5 Fehlerprotokolle
- 12.4.3: Entwicklung von Informationssystemen → Zugangskontrolle zu Quellcode
- 13.2.1, c): Umgang mit Sicherheitsvorfällen → Verantwortlichkeiten und Verfahren

8 Web Application Security

Unter Web Application Security verstehen wir alle Gefährdungen, die ein Softwareentwickler bei der Entwicklung von Web-basierter Software berücksichtigen sollte und alle Sicherheitsmaßnahmen, die er implementieren sollte, um diese Gefährdungen auszuschließen.

8.1 Einordnung

8.1.1 Spezialfall der applikatorischen Sicherheit

Web Application Security ist als Spezialfall der applikatorischen Sicherheit einzuordnen. Allerdings hat dabei neben den zentralen Aufgaben der Authentisierung, Zugriffsschutz und Protokollierung die „Verhinderung ungeplanter Zugriffsmöglichkeiten“ (vgl. 7.2.1) eine besondere Bedeutung, weil es sehr viele ungeplante Zugriffsmöglichkeiten und extrem viele potenzielle Angreifer gibt.

8.1.2 Quellen

Mit dem Thema Web Application Security beschäftigen sich u.a. die beiden Web Präsenzen

- https://www.owasp.org – Open Web Application Security Project
- http://www.webappsec.org – Web Application Security

Bei beiden handelt es sich um non-profit-Organisationen, die Informationen und Unterstützung zur Web Application Security anbieten. Beide Portale gehen von Gefährdungen und Maßnahmen aus.

Das OWASP ist für die etwas plakative Darstellung einer „Top Ten-Liste“ der Sicherheitsprobleme im Web bekannt. Die Inhalte des WebAppSec sind weitgehend ähnlich aber weniger imposant dargestellt. Das OWASP betreibt darüber hinaus noch unterstützende Projekte wie eine Sicherheitsbibliothek für Java (ESAPI 8.5) oder ein unsicheres Websystem zum Üben (WebGoat).

Übung 29: Surfen Sie OWASP und WebAppSec an, finden Sie die OWASP-Top Ten Liste und die „attacks, weaknesses and solutions“ nach WebAppSec.

In gedruckter Form gibt es seit 2012 (deutsche Ausgabe 2013) das Buch

- Michal Zalewsky: Tangled Web, Der Security-Leitfaden für Webentwickler

Das Buch wird in Fachkreisen hoch gelobt. Es geht von der Architektur der unterschiedlichen Web-Techniken und Architekturen aus und leitet davon die Gefährdungen und Maßnahmen ab. Es ist ziemlich anspruchsvoll.

Ein Foliensatz meiner Wenigkeit aus dem Jahr 2011 stellt außerdem einige Themen der Web Application Security dar, gegliedert nach den Top Ten des OWASP aus dem Jahr 2010, die übrigens andere sind als die OWASP-Top 10 des Jahres 2013. Darin enthalten sind auch die weiteren Quellen

- Essential PHP Security von Chris Shiflett (2005)http://www.amazon.de/Sichere-Webanwendungen-mit-Tobias-Wassermann/dp/3826617541/ref=sr_1_7?s=books&ie=UTF8&qid=1287560817&sr=1-7
- Sichere Webanwendungen mit PHP von Tobias Wassermann (2007)http://www.amazon.com/Web-Security-Testing-Cookbook-Systematic/dp/0596514832/ref=sr_1_1?ie=UTF8&qid=1287560899&sr=8-1
- Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast by Paco Hope and Ben Walther (2008)http://www.amazon.com/Web-Application-Hackers-Handbook-Discovering/dp/0470170778/ref=sr_1_3?ie=UTF8&qid=1287560899&sr=8-3
- The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws by Dafydd Stuttard and Marcus Pinto (2007)http://www.amazon.com/Developers-Guide-Web-Application-Security/dp/159749061X/ref=sr_1_7?ie=UTF8&qid=1287560899&sr=8-7
- Developer's Guide to Web Application Security by Michael Cross (2007)http://www.amazon.com/Web-Security-Step---Step-Reference/dp/0201634899/ref=sr_1_8?ie=UTF8&qid=1287560899&sr=8-8
- Web Security: A Step-by-Step Reference Guide by Lincoln D. Stein (1998)http://www.amazon.com/Web-2-0-Security-Defending-AJAX/dp/1584505508/ref=sr_1_9?ie=UTF8&qid=1287560899&sr=8-9
- Web 2.0 Security - Defending AJAX, RIA, AND SOA by Shreeraj Shah (2007)http://www.amazon.com/Improving-Web-Application-Security-Countermeasures/dp/0735618429/ref=sr_1_3?ie=UTF8&qid=1287561087&sr=8-3
- Improving Web Application Security: Threats and Countermeasures by Microsoft Corporation (2003)http://www.amazon.com/Deadliest-Application-Attacks-Syngrass-Deadlest/dp/1597495433/ref=sr_1_4?ie=UTF8&qid=1287561087&sr=8-4
- Seven Deadliest Web Application Attacks (Syngrass Seven Deadlest Attacks) by Mike Shema (2010)
- The World Wide Web Consortium (W3C) publishes the http://www.w3.org/Security/Faq/
- Security guru Bruce Schneier recommends e.g. http://www.smashingmagazine.com/2010/01/14/web-security-primer-are-you-part-of-the-problem/
- 2010 CWE/SANS Top 25 Most Dangerous Software Errors: http://cwe.mitre.org/top25/
- Lots of consulting companies: Google „Web Security“ or „Web application security“
- Google Security Scanner: http://www.heise.de/security/meldung/Web-Security-Scanner-von-Google-959931.html --> Skipfish
- More Security Scanners: http://projects.webappsec.org/w/page/13246988/Web-Application-Security-Scanner-List

8.1.3 Abgrenzung

Zur Application Security gehören bekanntlich die Maßnahmen der Entwickler aber nach 7.3.1 nicht die Maßnahmen der Betreiber. Im Fall der Web Application Security sehen wir dies generell genau so, stehen damit aber im Widerspruch zum OWASP, das als Top 5 die „sichere Konfiguration“ aufführt. In diesem Sinne sind die Übergänge zwischen sicherer Entwicklung und sicherem Betrieb fließend. Auf jeden Fall gehört aber zur Application Security, dass eine Anwendung so entwickelt werden muss, dass sie sicher betrieben werden **kann**.

8.1.4 Web (Application) Security im ISO 27002

Die ISO 27002 behandelt Web Anwendungen kaum separat mit Ausnahme von Kap. 10.9: E-Commerce Anwendungen. Im Wesentlichen steht die ISO 27002 damit auf dem Standpunkt, dass die Probleme der Web (Application) Security nur inhaltlich andere sind aber nicht strukturell andere. Einerseits ist dies korrekt, andererseits liefert uns die ISO 27002 damit keine Unterstützung.

8.2 Injection

Das Thema der Injection-Schwachstellen steht im OWASP seit Jahren auf Platz 1. Es besteht allerdings eine enge Verwandtschaft zwischen Injection und Cross-Site-Scripting, was zu der Gefährlichkeit von Injection-Angriffen entscheidend beiträgt.

8.2.1 Beispiel / Übung

8.2.1.1 Vorbereitung

Legen Sie sich in Eclipse ein dynamic web project „BW5S“ an und darin

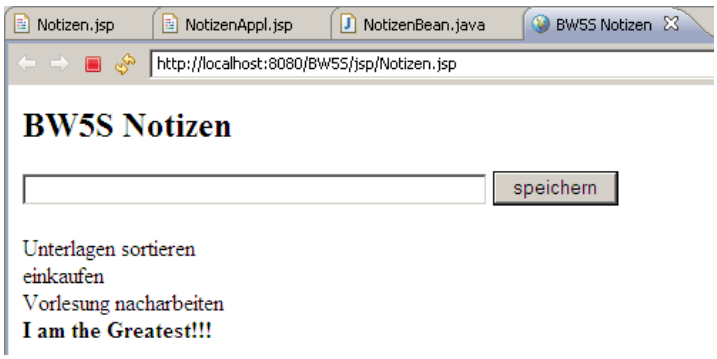
- eine Bean zur Speicherung von Notizen in einem Vector<String>
- eine JSP-Seite zur Texteingabe und –anzeige
- eine JSP-Appl, um den eingegebenen Text in die Anzeige zu übertragen

Das vom Entwickler vorgesehene Verhalten besteht darin, dass die oben eingegebenen Notizen unten angezeigt werden.

8.2.1.2 Erster Ausnahmefall

Was passiert, wenn Sie im Eingabefeld eintippen:

Ergebnis:

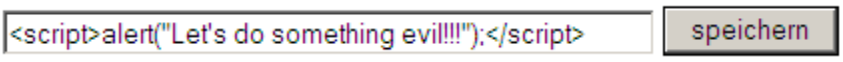


Die Eingabe erscheint fett auf der Listenseite. Entgegen der Vorstellung des Entwicklers wird also nicht nur der Text sondern auch die Formatierung von oben nach unten übertragen

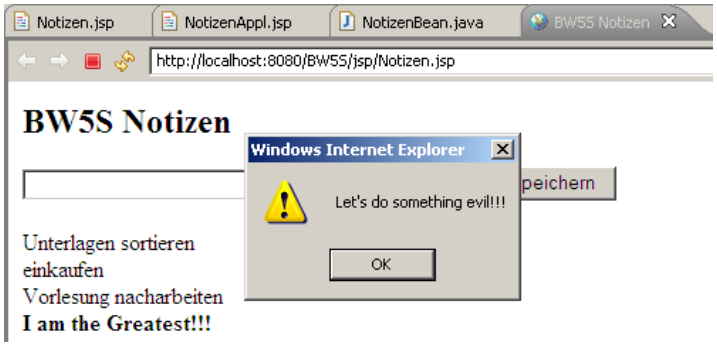
Übung 30: Analysieren Sie im Detail, wie dies passieren konnte.

8.2.1.3 Zweiter Ausnahmefall

Was passiert, wenn Sie im Eingabefeld eintippen:



Ergebnis:



Entgegen der Vorstellung des Entwicklers konnte oben JavaScript-Code eingegeben werden, der ohne Weiteres ausgeführt wird.

Übung 31: Analysieren Sie im Detail, wie dies passieren konnte.

8.2.1.4 Zweiter Ausnahmefall Revisited

Geben Sie danach wieder normalen Text ein:

Nachdem Sie auf „speichern“ klicken erscheint wieder der JavaScript-alert.

Übung 32: Analysieren Sie im Detail, wie dies passieren konnte.

8.2.1.5 Auswertung

Die eingegebenen Zeilen wurden in die Ergebnisseite übernommen und dort zur Interpretation an den Browser übergeben:

```
...
<body>

<h2>BW5S Notizen</h2>
<form action="./NotizenAppl.jsp" method="get" >
      <input type="text" name="eingabe" value="" size="50" />
      <input type="submit" name="speichern" value="speichern" />
</form>
<p>
      Unterlagen sortieren<br />
einkaufen<br />
Vorlesung nacharbeiten<br />
<b>I am the Greatest!!!</b><br />
<script>alert("Let's do something evil!!!");</script><br />
Weitere Tests durchführen<br />

</p>
</body>
</html>
```

Dabei wurde nicht berücksichtigt, dass der Browser

- mitgelieferte Formatierungsinformationen interpretiert und
- mitgelieferte Scripte interpretiert und ausführt.

8.2.2 Definition und Analyse

Eine Injection-Schwachstelle liegt vor, wenn ein Angreifer Eingaben entgegen den Vorstellungen des Entwicklers an einen Interpreter schicken kann. Wir haben jetzt kennengelernt:

- HTML-Injection, vgl. 8.2.1.2 und
- Script-Injection, vgl. 8.2.1.3

8.2.2.1 Die Eingabeproblematik

Ein Software-Entwickler hat immer eine Vorstellung, wie seine Software genutzt werden kann und soll. Auf der Basis dieser Vorstellung wird die Software entwickelt, so dass sie funktioniert wie geplant, wenn sie genutzt wird wie geplant. Diese Art der Softwareentwicklung ist eigentlich ziemlich einfach.

Ein fortgeschrittener Softwareentwickler wird zumindest in Betracht ziehen, welche Fehler seine Benutzer (in der Regel unabsichtlich) machen könnten, so dass die Software im Fall eines solchen vorhergesehenen Fehlers angemessen reagiert, z.B. mit einer Fehlermeldung und ohne eine Aktion durchzuführen. Dies stellt eine Behandlung vorausgesehener Ausnahmesituationen dar.

Ein Angreifer dagegen versucht, eine nicht vorausgesehene Ausnahmesituation herbeizuführen und damit die Anwendung zu einer ungeplanten Aktion zu bringen. Der entscheidende Unterschied liegt darin, dass es sich dabei nicht um unabsichtliche Fehler von Benutzern handelt, sondern um ausgeklügelte schädliche Eingaben.

8.2.2.2 Analyse des Beispiels

In einem Browser gibt es zunächst verschiedene Eingabemöglichkeiten, z.B.

- URL-Eingabe in der URL-Zeile,
- Parametereingabe in Formularen,
- Menüeingaben, z.B. Aufruf von Bookmarks, Rücksprung auf die letzte Seite, etc.

Außerdem arbeiten im Browser unterschiedliche Interpreter zusammen, z.B.

- HTML-Interpreter,
- CSS-Interpreter,
- JavaScript-Interpreter.

Weitere (externe) Interpreter oder Programme können von einem Browser bei Bedarf aufgerufen werden, z.B.

- SQL-Interpreter bei Aufruf von Datenbankfunktionen,
- PDF, Word, Excel, etc. zur Dateianzeige
- Plug-Ins

Für Entwickler ist es deshalb fast unmöglich, immer alle Interpreter im Blick zu behalten. Verschärft wird die Situation dadurch, dass unterschiedliche Browser häufig unterschiedlich arbeiten und z.B. URL-Eingaben unterschiedlich interpretieren.

8.2.2.3 Lösungsansatz für das Beispiel

Offensichtlich gelingt der Angriff auf das Beispiel nur deshalb, weil der Angreifer durch Eingabe der Sonderzeichen „<“ und „>“ den Fließtext verlassen hat und einen Interpreter ansprechen konnte. Lösungsmöglichkeiten für diese Schwachstelle sind also

- Herausfiltern und Ignorieren / Löschen aller eingegebener Zeichen „<“ und „>“
- Codieren der Zeichen „<“ und „>“ z.B. als ASCII-Codes %60 und %62 bzw. als < und >

8.2.3 SQL Injection

SQL Injection liegt vor, wenn ein Angreifer es schafft, eine Datenbank zu einer ungeplanten Aktion zu verleiten. Dafür ist eine unsichere Anwendung nötig. Im Folgenden konstruieren wir eine solche.

8.2.3.1 Beispiel / Übung

Wir wollen unser Beispiel von oben erweitern, indem wir die eingegebenen Notizen auch in einer Datenbanktabelle speichern.

Vorbereitung

Wir brauchen eine Datenbank und darin eine Tabelle NOTIZEN der folgenden Form:

```
create table notizen(
id    integer    not null primary key
      generated always as identity,
notiz clob(4096) not null
)
```

Außerdem brauchen wir in unserem Projekt die nötigen Klassen für eine JDBC-Connection (z.B. DB2Access, etc.).

Wir kopieren Notizen.jsp, NotizenAppl.jsp und NotizenBean zu Notizen1.jsp, Notizen1Appl.jsp und Notizen1Bean.

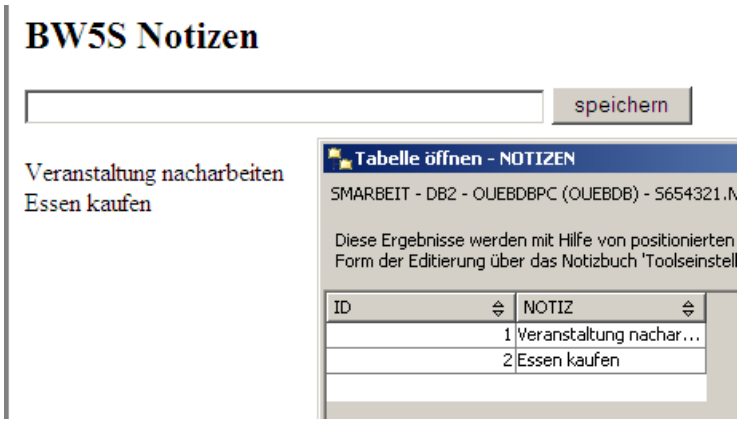
Wir erweitern die Notizen1Bean um eine Methode storeToDB(String notiz), die die angegebene Notiz in der oben angegebenen Tabelle speichert. Um ein Beispiel mit einer Sicherheitslücke zu schaffen, verwenden wir kein PreparedStatement:

```
public void storeToDB(String notiz) throws NoConnectionException, SQLException {
    Connection dbConn = new DB2Access().getConnection();
    String sql = "INSERT INTO NOTIZEN (NOTIZ) " +
                 "VALUES ('" + notiz + "')";
    System.out.println(sql);
    dbConn.createStatement().execute(sql);
}
```

In der Notizen1Appl.jsp rufen wir die neue Methode direkt hinter der add(notiz)-Methode auf. Falls eine Exception auftritt, zeigen wir den Stack Trace an:

```
if (speichern.equals("speichern")){
    nb.add(eingabe);
    try{
        nb.storeToDB(eingabe);
    }catch(Exception e){
        e.printStackTrace();
    }
}else{
}
response.sendRedirect("./Notizen1.jsp");
```

Im Regelfall wird nun die eingegebene Notiz nicht nur in die Liste aufgenommen, sondern zusätzlich in der Datenbank gespeichert:



Theoretischer Ausnahmefall

In der Literatur wird als typischer Ausnahmefall etwa die Eingabe

```
bla');drop table notizen;--
```

aufgeführt.

Übung 33: Was müsste dabei passieren und warum?

Dies funktioniert aber mit JDBC nicht, da JDBC nicht mehrere Kommandos hintereinander erlaubt.

Mehr Vorbereitung

Um einen echten Schaden zu kontruieren, erweitern wir unsere Anwendung noch weiter, nämlich so, dass nach der Eingabe einer Notiz sämtliche Notizen aus der Datenbank gelesen und auf dem Bildschirm angezeigt werden. In der Notizen1Bean ergänzen wir folgende Methode:

```
public String getDbNotizenAlsHtml() throws NoConnectionException, SQLException{
    Connection dbConn = new DB2Access().getConnection();
    String sql = "SELECT NOTIZ FROM NOTIZEN";
    System.out.println(sql);
    ResultSet dbRes = dbConn.createStatement().executeQuery(sql);
    String dbAlsHtml = "";
    while (dbRes.next()){
        dbAlsHtml += dbRes.getString(1).trim() + "<br />" + "\n";
    }
    return dbAlsHtml;
}
```

In der Notizen1.jsp binden wir diese ein:

```
<p>
      <jsp:getProperty name="nb" property="alleNotizenAlsHtml" />
      <jsp:getProperty name="nb" property="dbNotizenAlsHtml" />
</p>
```

Die eingegebenen Notizen erscheinen jetzt doppelt: Einmal aus dem Hauptspeicher und einmal aus der Datenbank:

Ausnahmefall

Geben Sie jetzt als Notiz die folgende ein:

```
bla'),(select userid || password from user fetch first 1 row only),('bla
```

Übung 34: Was müsste passieren und warum?

Ergebnis: Wir schaffen es, durch Zugriff auf eine andere Tabelle eine geheime Userid/Passwort-Kombination auf den Schirm zu bekommen (vorausgesetzt, wir wissen wie die Tabelle heißt, etc.)

Weiterer Ausnahmefall

Übung 35: Konstruieren Sie den folgenden Ausnahmefall selbstständig:

Verwenden Sie die Anmeldefunktionalität des letzten Semesters oder entwickeln Sie ein Anmeldeformular mit Userid und Password. Prüfen Sie mit dem folgenden SQL-Kommando, ob ein passender User vorhanden ist:

```
String sql = "SELECT * FROM USER " +
             "WHERE USERID = '" + userid + "' " +
               "AND PASSWORD = '" + password + "'";
```

Sie müssen ggfs. die Methode checkUseridPassword() zeitweilig ändern.

Geben Sie nun eine beliebige Userid ein und als Passwort: 1’ OR ’1’ = ’1

Prüfen Sie ob Sie angemeldet sind.

8.2.3.2 Lösungsansatz

Analog zu 8.2.2.3 könnte man auf die Idee kommen, die Problematik zu lösen, indem man Sonderzeichen Hochkomma, Semikolon und Bindestrich verbietet, ignoriert oder filtert. Es könnte aber sein, dass eine andere Datenbank wieder andere Sonderzeichen benutzt, deshalb wäre dies hier unsicher.

Der wesentliche Lösungsansatz gegen SQL Injection ist deshalb die Benutzung von Prepared Statements wie Sie es in der Vorlesung „Webanwendungen“ gelernt haben. Da dies inzwischen weithin bekannt ist, kann SQL Injection prinzipiell als gelöstes Problem gelten.

8.2.4 URL Eingabe

Eine Eingabe in ein Web-Formular führt zu einem http-Request der Form …?feldname=value. Wie besprochen kann die Eingabe eines bösartigen value zu einem Angriff führen.

Natürlich ist es für einen Angreifer auch möglich, manuell einen http-Request auszuführen, anstatt ein Formular abzuschicken. Grundsätzlich hat der Angreifer dadurch mindestens dieselben Möglichkeiten wie durch Formulareingabe.

Zwei Anmerkungen:

- Ein Angriff durch eine manipulierte URL mag handwerklich aufwändiger sein als ein Angriff über ein Formularfeld, aber wenn der Angreifer genügend talentiert ist, ist es ihm egal, ob er seine URL mit GET oder POST abschicken soll. Er wird dazu in der Lage sein.
- Angriffe über URLs lassen sich sehr einfach automatisieren. D.h. ein Angreifer kann Skript-gesteuert viele URLs hintereinander abschicken.

Zusammengefasst sind Angriffe über URLs mindestens so gefährlich wie Angriffe über Formularfelder.

8.3 Architektur der Angriffe

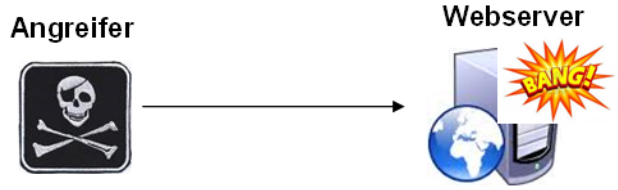
Als Angriffsarchitekturen unterscheiden wir:

- Unidirektionale Angriffe, bei denen nur der jeweilige Server angegriffen wird,
- Sogenannte „Cross-Site“ Angriffe, bei denen andere Benutzer angegriffen werden. In diesem Fall unterscheidet man zwischen „stored“ und „reflected“ Angriffen.

8.3.1 Unidirektional

8.3.1.1 Einfach

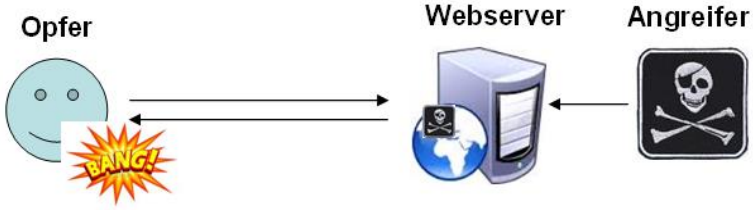
Besonders einfach ist ein Angriff, wenn er nach einem Klick bereits beendet ist:



Dabei wird nur der Server angegriffen, z.B. um sich eine Anmeldung zu erschleichen (vgl. abschließendes Beispiel von 8.2.3.1) oder Daten zu manipulieren.

8.3.1.2 Durch Server-Manipulation

Ebenfalls unidirektional ist ein Angriff, wenn ein Angreifer sich Zugriff auf eine Datenbank (oder einen anderen Server-basierten Datenspeicher) verschafft und dort bösartigen Code hinterlegt, der dann von Benutzern aufgerufen wird:



Je nachdem, um was für einen Datenbestand es sich dabei handelt, kann auf diese Art z.B. eine sehr große Zahl von Benutzern angegriffen werden oder ein bestimmter Benutzer ganz spezifisch angegriffen werden.

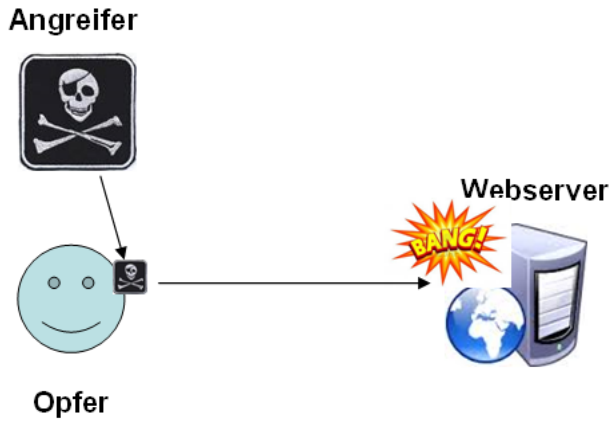
Da Benutzer und Administratoren nicht mit bösartigem Code in der Datenbank rechnen, können solche Angriffe sehr effizient sein.

8.3.1.3 Heimliche Delegation

Evtl. kennt ein Angreifer einen Weg, einen Server anzugreifen, kann diesen aber nicht selbst ausnutzen, weil ihm dazu die Berechtigungen fehlen. Eine Möglichkeit, den Angriff trotzdem auszuführen kann ggfs. sein:

- Der Angreifer spielt einem anderen Benutzer einen bösartigen Request zu und
- dieser führt den Request unwissend aus.

Der Benutzer, der den Request ausführt, schadet dabei meistens seinen eigenen Daten auf dem Server, deshalb bezeichnen wir diesen dabei als Opfer:



Opfer erhalten den bösartigen Request als Link, z.B. in einer Mail oder auf einer bösartigen Webseite. Der Request wird ausgeführt, wenn der Benutzer die Berechtigung dazu besitzt, z.B. weil er

- sich in einem dafür autorisierten Netzwerkbereich befindet,
- sich bereits in einer authentisierten Session befindet (vgl. 8.7.3.2),
- seine korrekten Anmeldedaten in ein sich öffnendes Anmeldefenster eingibt.

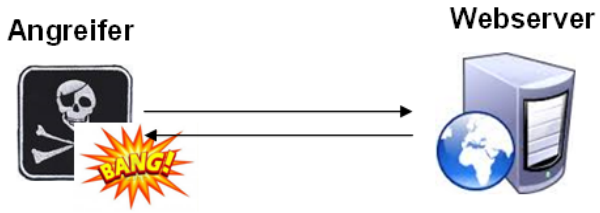
Um möglichst viele Opfer zu treffen, müsste ein Angreifer also sehr vielen Benutzer seinen bösartigen Request unterschieben (Spam). Da die meisten Internet Benutzer inzwischen recht sensibel auf Spam reagieren, ist die Infektionsrate eher gering, was aber durch die immense Zahl von Benutzern wettgemacht wird.

Wenn der Angreifer ein Opfer kennt und gezielt angreifen möchte, kann er den Link speziell für das Opfer aufbereiten, um die Wahrscheinlichkeit zu erhöhen, dass dieses darauf hereinfällt.

8.3.2 Bidirektional

8.3.2.1 „Selbstangriff“

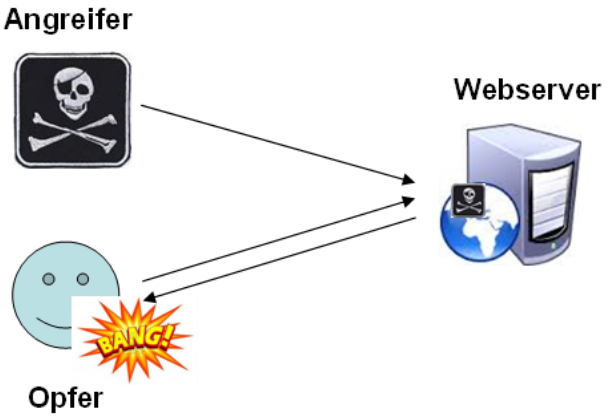
Ein Angreifer hat immer das Problem, dass er die Antwort auf einen HTTP-Request immer selbst bekommt. Wenn dadurch ein Schaden bei ihm selbst entsteht, ist das für ihn sinnlos. Wir könnten dies als „Selbstangriff“ bezeichnen.



8.3.2.2 „Stored“ Angriff

Ein „Stored Cross-Site“-Angriff erfolgt, indem

- ein Angreifer es schafft, mit einem request bösartigen Code auf einem Server zu speichern, z.B. in einer Datenbank, einem Forum, Gästebuch, o.ä. und
- ein Benutzer diesen bösartigen Code ausführt



Zu diesem Angriff gehören unsere Beispiele mit HTML-Injection und Script-Injection, wenn der injizierte Code zunächst auf einem Server gespeichert wird und dann bei einem anderen Benutzer ausgeführt wird. Wenn es sich dabei um injizierten JavaScript Code handelt, spricht man von „Stored Cross-Site Scripting“.

Um möglichst viele Opfer zu treffen, müsste ein Angreifer also Server mit möglichst vielen Benutzern angreifen.

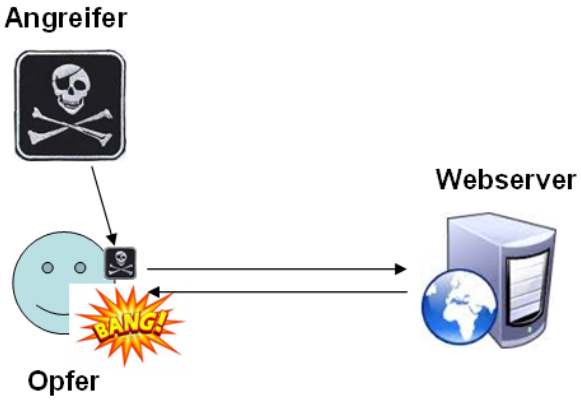
Um ein Opfer gezielt anzugreifen, müsste ein Angreifer einerseits wissen, auf welche Server das Opfer zugreift und anderseits einen dieser Server erfolgreich angreifen können.

8.3.2.3 „Reflected“ Angriff

Ein „Reflected Cross-Site“-Angriff erfolgt in der Art einer heimlichen Delegation (vgl. 8.3.1.3), indem

- ein Angreifer einem anderen Benutzer einen bösartigen Request zuspielt,
- den dieser dann unwissend ausführt.

Wenn der bösartige Request aus JavaScript besteht, spricht man von „Reflected Cross-Site Scripting“. In diesen Zusammenhang fällt auch „DOM-based Cross-Site Scripting“. Beim zweiten Teil eines reflected Angriffs handelt es sich um einen versehentlichen Selbstangriff des Opfers (vgl. 8.3.2.1):



Die weiteren Anmerkungen aus 8.3.2.1 gelten auch hier.

8.4 Cross-Site Scripting (XSS)

Das Thema Cross-Site Scripting, abgekürzt XSS[6], steht in den Top 10 des OWASP auf Platz 3. Ein Angriff besteht darin, dass ein Angreifer sein Opfer dazu bringt, bösartigen JavaScript-Code auszuführen.

[6] Früher wurde Cross-Site Scripting mit CSS abgekürzt, aber um Verwechslungen mit Cascading Stylesheets zu vermeiden, wird heutzutage die Abkürzung XSS verwendet.

8.4.1 Arten von XSS

entsprechend der in 8.3 dargestellten Architekturen kann XSS auf dreierlei Arten erfolgen:

- Unidirektionales XSS mit Server-Manipulation,
- Stored XSS,
- Reflected XSS incl. DOM-based XSS.

Die zweite und dritte Möglichkeit stehen dabei im Zusammenhang mit Injection, wobei Sicherheitsmaßnahmen gegen Injection darauf abzielen, dass der bösartige Code erst gar nicht auf den Server kommt. Im ersten Fall helfen solche Maßnahmen nicht.

8.4.2 Lösungsansätze

8.4.2.1 XSS mit Server-Manipulation

Zunächst ist zu beurteilen, ob das Risiko einer Server-Manipulation überhaupt so hoch ist, dass dedizierte Schutzmaßnahmen ergriffen werden müssen.

Wenn eine Web Anwendung auf eine Datenbank zugreift, bei der davon ausgegangen wird, dass bösartiger Code enthalten sein kann, dann könnte die Web Anwendung beim Aufbau der HTML-Seiten wieder Sonderzeichen wie <, >, &, ;, … ausfiltern oder codieren. Im Gegensatz zur Filterung von Input-Daten (als Schutz gegen Injection) handelt es sich also dabei um die Filterung von Output-Daten.

Wenn der bösartige Code direkt als Link auf einer statischen Webseite eingebaut wurde, gibt es keine Schutzmöglichkeit.

8.4.2.2 Stored XSS

Bei Stored XSS muss der bösartige Code des Angreifers sowohl den Hinweg (input) als auch den Rückweg (output) bewältigen. Sowohl Schutzmaßnahmen gegen Injection als auch Output-Filtermaßnahmen können also hier helfen.

8.4.2.3 Reflected XSS

Reflected XSS kann zum Einsatz kommen, wenn ein Eingabefeld gar nicht erst gespeichert, sondern direkt zurückgegeben wird.

Übung 36: Entwickeln Sie eine Webseite, die einen Namen abfragt, sich selbst aufruft und dann „Hallo <name>“ ausgibt. Rufen Sie dann diese Webseite auf, indem Sie anstelle des Namens eingeben `<script>alert('Kuckuck');</script>`

Übung 37: Schreiben Sie nun eine HTML-Mail an ein Opfer, in der Sie einen bösartigen Link auf Ihre Seite einfügen.

Auch bei Reflected XSS können Schutzmaßnahmen gegen Injection sowie Output-Filtermaßnahmen helfen.

Übung 38: Identifizieren Sie die Stellen, wo Sie die input- oder output-Prüfung anbringen müssten.

8.4.2.4 Detailliertere Lösungsansätze…

… finden Sie unter
https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet

Übung 39: Schauen Sie sich diese an.

8.5 ESAPI

Bei der letzten Übung werden Sie schon auf den Begriff ESAPI gestoßen sein. Schauen wir uns diesen näher an.

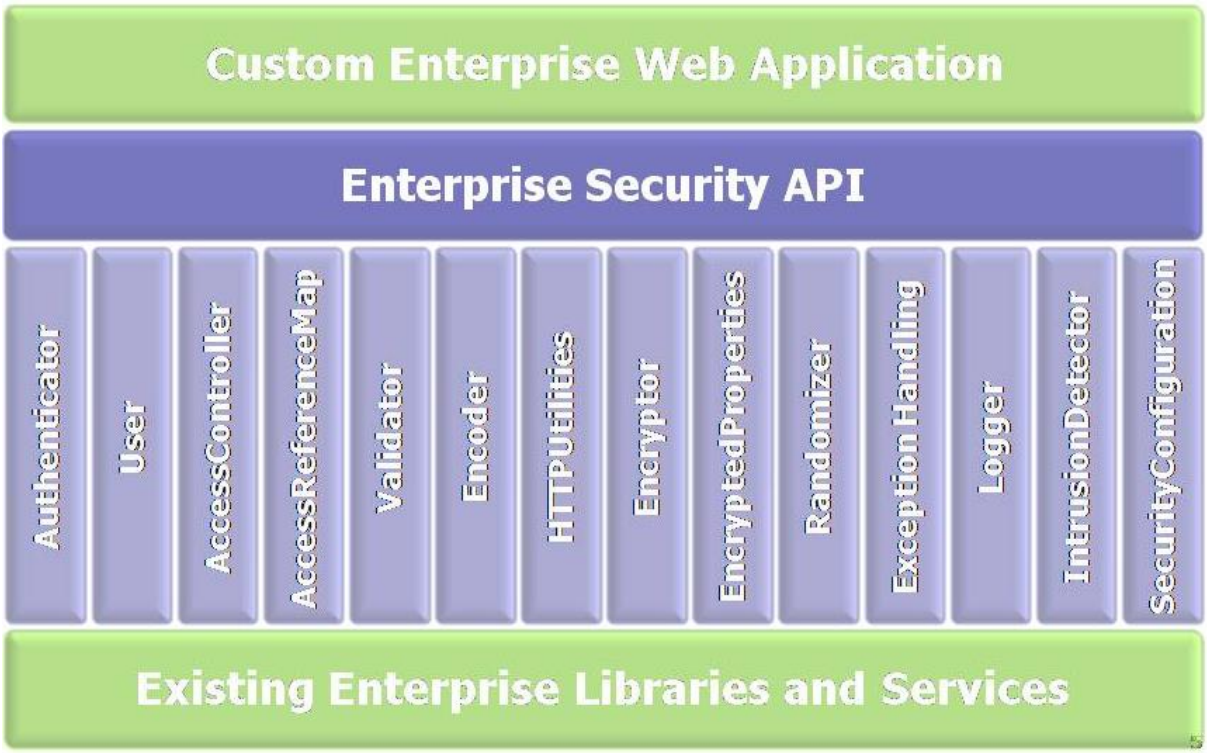
Bei der OWASP-Entwicklung ESAPI (Enterprise Security API) handelt es sich nach https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API um „… a free, open source, web application security control library that makes it easier for programmers to write lower-

risk applications. The ESAPI libraries are designed to make it easier for programmers to retrofit security into existing applications."

Es wird davon ausgegangen, dass nicht jeder Webentwickler das nötige Know How besitzt, um sichere Webanwendungen zu implementieren, insbesondere um immer an alle vorgenannten Sicherheitsmaßnahmen zu denken. So soll ESAPI

- Webentwicklern einigen Aufwand zur Sicherheit ihrer Anwendungen abnehmen und
- ein Framework bereitstellen, das auch mit eingeschränktem Sicherheits-Know How anzuwenden ist.

Nach http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/package-summary.html besteht ESAPI aus den folgenden Paketen:



Vor dem Hintergrund der aktuellen Enthüllungen über die NSA und andere Geheimdienste kann natürlich nicht unbedingt davon ausgegangen werden, dass ESAPI frei von Hintertüren ist. Einerseits interessiert uns an dieser Stelle aber die generelle Funktionsweise, andererseits ist ESAPI Open Source, kann also theoretisch gegen Hintertüren geprüft werden.

ESAPI besteht aus Paketen, Klassen, Interfaces und Dokumentation, die Entwickler bei der Entwicklung eigener Webanwendungen frei verwenden dürfen. Diese liegen in verschiedenen Programmiersprachen vor. Wir beschränken uns hier auf Java. Die API-Dokumentation findet sich unter http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/index.html.

ESAPI muss im System installiert und konfiguriert werden. Dabei werden z.B. die zu verwendenden Verschlüsselungsalgorithmen angegeben.

8.6 Validierung

8.6.1 Allgemeines

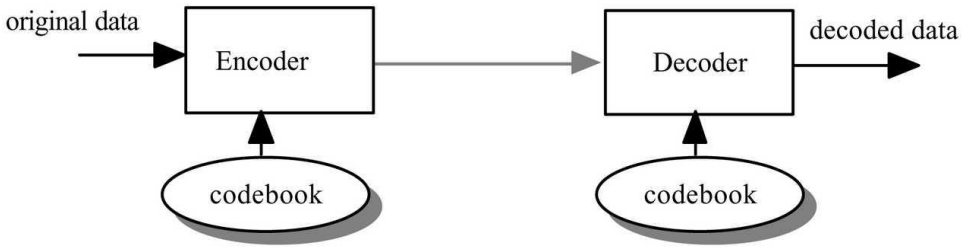
In 8.2 und 8.4 haben wir gesehen, dass Daten sowohl bei der Eingabe als auch bei der Ausgabe validiert werden müssen:

- Eingabe: Gegen bösartige Benutzer-Eingabe
- Ausgabe: Gegen Angriffe mit Server-Manipulation

8.6.2 Encoding und Decoding

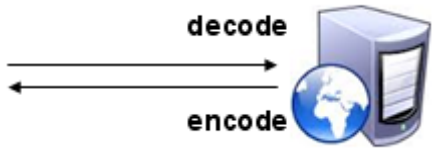
Grundsätzlich versteht man unter Encoding und Decoding die Veränderung von Nachrichten vor und nach einer Übertragung. Dies umfasst ganz verschiedene Ebenen:

- Gesellschaftlich: Ein Gedanke wird in Sprache ausgedrückt (encodiert) und vom Hörer interpretiert (decodiert). Gelegentlich kommt der Gedanke dann beim Hörer anders an, als er vom Sprecher gemeint war.
- Im Netzwerk: Daten werden derart encodiert (serialisiert), dass sie in einem Netzwerk verlustfrei übertragen werden können. Auf Empfängerseite werden die Daten wieder decodiert und ggfs. zusammengesetzt. Wenn alles gut läuft, gibt es hier keine Ausfälle.
- In Client-Server Anwendungen:
  - Ein Benutzer gibt Daten ein, die vom Client encodiert und an den Server übertragen werden. Dieser decodiert die Daten und liefert sie an die Anwendung.
  - Die Anwendung liefert Daten. Diese werden vom Server encodiert, vom Client decodiert und vom Benutzer angenommen.



Diese Beispiele illustrieren encoding / decoding aus Netzwerk- bzw. Übertragungssicht. Aus der Sicht eines einzelnen Teilnehmers, z.B. einer Anwendung kann auch gesagt werden:

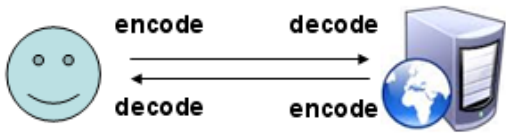
- Wenn Daten von einem Sender angenommen werden, müssen diese decodiert werden. Nach der Verarbeitung und vor der Ausgabe müssen die Daten für die Rückübertragung encodiert werden.



Encoding und decoding hängen immer davon ab, wer der Kommunikationspartner ist. Im obigen Bild entspricht dies dem „codebook“. In dem Sinne ist

- Decoding immer Sender-spezifisch und
- Encoding immer Empfänger-spezifisch

8.6.3 Situation bei Webanwendungen



8.6.3.1 Encoding im Web-Client

Bei Webanwendungen findet zunächst ein automatisches Encoding im Browser statt, der aus den Benutzereingaben einen sinnvollen http-Request erzeugt. Der Browser berücksichtigt dabei verschiedene Datenquellen:

- Die URL des Requests, die üblicherweise oben in der URL-Zeile steht,
- Parameter, die aus einem Formular gelesen werden und nach einem „?“ in der Form „name=value“, ggfs. getrennt durch „&“ an die URL-Zeile angehängt werden,
- Cookies, die im Browser gespeichert sind und im Header des http-Requests mitgeschickt werden.

Der Browser kann dabei Umwandlungen vornehmen, z.B. indem er Sonderzeichen in ASCII-Zeichen umwandelt, Leerzeichen in „+“ oder „%20“ o.ä.

Anmerkungen:

- Sonderzeichen können bereits im HTML codiert sein, z.B. „<“ als „<“. Da der Browser nicht weiß, ob dies der Fall ist, wendet dieser darauf keine Encodierung an.
- Wir haben kaum Einfluss darauf, was der Browser automatisch macht, müssen dies aber möglichst gut verstehen, um auf Server-Seite darauf reagieren zu können.
- Eingabeprüfungen mit Java Script können benutzerfreundlich sein, bieten aber keine Sicherheit, da sie von einem Angreifer umgangen werden können, indem dieser direkt auf http-Ebene mit dem Server kommuniziert.

8.6.3.2 Decoding auf Server-Seite

Der Webserver nimmt die Daten an und liefert sie an die Webanwendung. Sowohl Webserver als auch Webanwendung haben eine Gelegenheit eine Decodierung vorzunehmen (Eingabe-Validierung). Mindestens sollten Sonderzeichen ausgefiltert werden, die der Browser nicht als gefährlich betrachtet, z.B. „<“ oder single quotes, vgl. 8.2.

Anmerkungen:

- Aus Sicherheitsgründen wird empfohlen,
  - nur bekannt ungefährliche Zeichen zuzulassen (white listing) anstatt nur bekannt gefährliche Zeichen zu verbieten (black listing),
  - eine geprüfte Sicherheitsbibliothek (z.B. ESAPI) zu verwenden, anstatt seinen Sicherheitscode selbst zu entwickeln.
- Falls nicht ausschließlich Browser als Sender in Frage kommen, sondern auch andere Anwendungen, sollte überlegt werden, ob darauf evtl. noch eine spezielle Decodierung anzuwenden ist.

8.6.3.3 Encoding auf Server-Seite

Die Ausgabe einer Webanwendung ist in der Regel eine HTTP-Response, deren Body eine HTML-Seite enthält, die wiederum aus Daten zusammengesetzt ist, die zumindest teilweise aus einer Datenbank stammen. Da in der Datenbank HTML-Sonderzeichen gespeichert sein können – entweder versehentlich oder nach einem erfolgreichen Angriff – sollte die Ausgabe keine uncodierten HTML-Sonderzeichen enthalten, also z.B. „<“ anstatt „<“. Dies nennt man HTML-Encoding.

Wenn die Ausgabe Inhalte enthält, die nicht an den HTML-Interpreter gehen, müssen diese ggfs. separat encodiert werden, z.B. CSS-Encoding, Datenbank-Encoding, JavaScript-Encoding.

8.6.3.4 Decoding auf Client-Seite

Die Decodierung auf Client-Seite erfolgt in der Regel automatisch im Browser, d.h. weder Benutzer noch Entwickler haben darauf Einfluss. Deshalb müssen alle möglichen Decoding-Probleme bereits beim Encoding auf Server-Seite berücksichtigt werden.

8.6.3.5 Encoding / Decoding gegenüber der Datenbank

Da die Webanwendung in der Regel mit einer Datenbank kommuniziert, können dabei weitere spezifische Codierungsanforderungen auftreten.

Auf der Seite der Datenbank laufen diese in der Regel automatisch ab, die Anwendung sollte aber berücksichtigen, dass ein Angreifer evtl. bösartigen Code in der Datenbank versteckt hat.

Übung 40: An welcher Stelle müsste dann eine Validierung erfolgen?

8.6.4 Canonicalization

Ein beliebter Trick bei Angreifern ist es, Daten so zu codieren, dass deren Bedeutung auf den ersten Blick nicht erkennbar ist, im Idealfall weder für Menschen noch für Prüfsoftware. Bsp.:

- GET target.de/public/index.html/../../private/pw.txt
- Mein Name ist %3Cb%3EBig Mouth%3C%2Fb%3E

Übung 41: Worin besteht jeweils der Trick?

Ein Mechanismus, um solche Codierung zu erzeugen, ist Mehrfach-Encoding, das natürlich nur dann funktioniert, wenn die Daten dann auch mehrere Decoding-Schritte durchlaufen.

Um solche Tricks zu umgehen, verwendet man die sogenannte „Canonicalization" bzw. „Kanonisierung", mit der man die untersuchten Daten zunächst in eine „kanonische Form" bzw. „Normalform" bringt. Das bedeutet, dass man alle Codierungen auflöst, um sie danach durch Validierung erkennen und ggfs. entfernen zu können. Auch hierfür soll möglichst eine geprüfte Sicherheitsbibliothek verwendet werden.

8.6.5 Exkurs: URL-Syntax[7]

Die beispielhafte Struktur einer absoluten URL könnte lauten

- http://roeckle.geheim@roeckle.de:80/public/index.php?bla=blubb&blabla=blubbblubb#kap2

Sie besitzt 8 Bestandteile:

Schema-/Protokollname	http:
Indikator für hierarchische URLs	//
Anmeldeinformationen, hier Benutzername „roeckle", Passwort „geheim"	roeckle.geheim@
Server	roeckle.de, alternativ auch IPv4, z.B. 127.0.0.1 oder IPv6, z.B. [0:0:0:0:0:0:0:1]
Port	:80
Hierarchischer Dateipfad	/public/index.php
Query-String	?bla=blubb&blabla=blubbblubb
Fragment-Identifier	#kap2

Damit eine solche URL eindeutig analysiert werden kann, dürfen die verwendeten Sonderzeichen :, /, ., @, ?, =, &, #, [,] nicht an anderen Stellen auftreten als im Beispiel dargestellt.

Falls doch, werden diese encodiert, und zwar standardmäßig mit %-Zeichen und Hexadezimal(-ASCII-)code, z.B. %40 für „@". Damit ist auch das Prozentzeichen selbst als Textzeichen verboten und muss ggfs. codiert werden (%25). Übrigens ist es erlaubt auch „normale" Zeichen zu codieren, was einen Benutzer sehr verwirren kann, z.B. roeckle%2e%64%65.

Solange die Möglichkeit besteht, dass eine URL weitergeleitet werden muss, darf diese nicht decodiert werden.

Da Browser benutzerfreundlich sein wollen, führt nicht alles zu einem Fehler, was eigentlich verboten ist, wird aber von verschiedenen Browsern z.T. unterschiedlich interpretiert, z.B. a@b@c. Auch bei nichtdruckbaren Zeichen ist oft unklar, was passieren soll: roeckle.%0A%0Dde

Als Server-Adresse kann auch eine IP-Adresse in anderer Form angegeben werden, z.B. http://example.com&kauderwelsch=1234@0177000000001/

Übung 42: Welcher Server wird hier NICHT angesprochen? Für Genießer: Welcher Server wird hier angesprochen?

Sicherheitstipp für Entwickler:

- Wenn Werte aus URLs gelesen werden, muss immer davon ausgegangen werden, dass diese unerlaubte Zeichen enthalten. Deshalb zuerst kanonisieren, dann decodieren, analysieren, filtern, etc.
- Wenn URLs aus Benutzereingaben zusammengebaut werden, sollten nur Buchstaben und Ziffern zugelassen werden. Alles andere sollte in Prozentzeichen umgewandelt werden.

[7] Die Inhalte dieses Kapitels sind ein kleiner Teil von Kap. 2 des Buches „Tangled Web" von Michal Zalewski.

- Relative URLs sollten zu voll-qualifizierten URLs verlängert und dann neu geprüft werden.
- Es sollten nur bekannte Schemanamen (meistens http://) zugelassen werden.
- Für Anmeldeinformationen und Server sollten nur Buchstaben, Ziffern, - und . zugelassen werden.

8.6.6 Encoding und Decoding in ESAPI

Die ESAPI-Bibliothek bündelt die Methoden für encoding, canonicalization und decoding in den Packages org.owasp.esapi.codecs und org.owasp.esapi.reference, speziell in der Klasse org.owasp.esapi.reference.DefaultEncoder[8].

8.7 Authentisierung

Inzwischen gibt es in Webarchitekturen sämtliche Arten von Authentisierung, die wir in 7.5 beschrieben haben. Wir fokussieren hier nicht auf eine vollständige Beschreibung, sondern beschreiben nur die einfachsten Mechanismen, Angriffsmöglichkeiten und Sicherheitsmaßnahmen.

8.7.1 Basic Authentication

8.7.1.1 Grundsätzliches

Das Verfahren der Basic Authentication ist das ursprünglich vorgesehene Authentisierungsverfahren für Webseiten (RFC 2617). Es delegiert die Authentisierung an Webserver und Browser und erfordert keine Programmierung seitens der Webentwickler.

Da es aber gewisse Sicherheitslücken besitzt und nicht optisch an einzelne Webseiten anzupassen ist, wird es heute fast gar nicht mehr verwendet. Trotzdem sollte man es IMHO mal ausprobiert haben.

8.7.1.2 Funktionsweise

Vorbereitung

Auf dem Webserver werden zwei (oder mehr) Dateien hinterlegt:

- Eine Datei (oder mehrere Dateien) `.htaccess`, die besagen, welche Verzeichnisse bzw. Dateien überhaupt durch Authentisierung geschützt werden sollen.[9]
- Eine Datei, die die Benutzer und deren Passwörter enthält. Diese Datei heißt im Standardfall `.htpasswd`.

Der Webserver wird so eingerichtet, dass

- die `.htaccess` auch beachtet wird. Im Apache Webserver muss dazu innerhalb von `<Directory ".../htdocs">` die Direktive `AllowOverride All` gesetzt werden (anschließend Server-Neustart).

Falls ein Zugang von außerhalb erfolgen soll, z.B. um den Netzwerktraffic mit Wireshark mitzuschneiden, dann ist ggfs. auch die Öffnung der Firewall für Port 80 notwendig.

Anmerkungen:

- Wenn die Datei `.htaccess` im Stammverzeichnis der Webpräsenz liegt, gilt sie zunächst für die gesamte Webpräsenz. Diese Gültigkeit in Unterverzeichnissen kann aber durch weitere `.htaccess`-Dateien in diesen Unterverzeichnissen wieder überschrieben werden.
- Es wird empfohlen, die Datei `.htpasswd` außerhalb der Web-zugreifbaren Verzeichnisstruktur zu speichern.
- Webserver können noch weitere Mechanismen zur Konfiguration der Authentisierung bereitstellen.

Ablauf

Wenn ein Benutzer per Browser eine Seite aufrufen möchte, die per .htaccess geschützt ist, dann

[8] http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/reference/DefaultEncoder.html
[9] In .htaccess-Dateien können außerdem noch viele andere Eigenschaften eines Webservers hinterlegt werden: http://de.selfhtml.org/servercgi/server/htaccess.htm

- schickt der Server anstatt der HTML-Seite eine HTTP-Response mit Rückgabewert 401 (Unauthorized) und einem Header „WWW-Authenticate“.
- Der Browser öffnet daraufhin ein Eingabefenster für Userid und Passwort. Wenn der Benutzer dieses ausfüllt und bestätigt, wird der ursprüngliche Request noch einmal geschickt, aber mit einem Header „Authorization“, der Userid und Passwort in base64-codierter Form enthält.

Übung 43: Installieren Sie einen Apache-Server. Ermitteln Sie im Web, wie .htaccess und .htpasswd einzurichten sind und bringen Sie das prototypisch zum Laufen. Beobachten Sie die Netzwerkpakete mit Wireshark. Schließen Sie dafür – falls nötig – zwei Rechner zusammen.

Schwachstellen

Die Basic Authentication hat grundsätzlich zwei typische Schwachstellen:
- Wenn ein Angreifer, die Netzwerkverbindung abhört, kann er das Userid/Passwort lesen und base64-decodieren.
- Wenn ein Angreifer die Datei .htpasswd vom Webserver lesen kann, sieht er alle Userids und je nach Konfiguration die Passwörter entweder im Klartext oder gehashed (vgl. 8.7.1.4).

Die typischen Schwächen des Passwort-Managements (vgl. 7.5.5) bestehen in Webanwendungen natürlich ebenfalls, diese werden aber an dieser Stelle nicht mehr behandelt.

8.7.1.3 Sicherheitslücke: Passwortübertragung im Netz

Exkurs: Base64-encoding

Bei base64-encoding handelt es sich um ein Verfahren, das 8-Bit-Daten in eine 6-Bit-Darstellung codiert, um Zeichensatzproblem bei der Übertragung zu vermeiden. Ein base64-codierter Text ist zwar nicht ohne Weiteres lesbar aber auch nicht verschlüsselt, sondern kann von jedermann wieder decodiert werden. Der Zweck der base64-Codierung ist Übertragungsstabilität (Verfügbarkeit) aber nicht Verschlüsselung (Vertraulichkeit).

Verfahren:
- Aus je 3 * 8 Bit (= 24 Bit) aufeinanderfolgenden Quellcode werden (24 Bit =) 4 * 6 Bit gebildet, die als Binärcode für die Zahlen 0 – 63 betrachtet werden.
- Jede dieser 64 Zahlen wird entsprechend einer vorgegebenen Umwandlungstabelle durch ein Zeichen A-Z, a-z, 0-9, + oder / codiert.
- Diese Zeichen werden als ASCII-Zeichen interpretiert und durch den entsprechenden ASCII-Code ersetzt. Als solche sind sie Zeichensatz-unabhängig, weil sie im ASCII-Code kleiner als 128 sind.
- Die Zeichen werden im ASCII-Code über das Netz übertragen. Es besteht kein Umwandlungsrisiko.
- Auf Empfängerseite läuft das Verfahren rückwärts:
  - ASCII-Code → ASCII-Zeichen
  - ASCII-Zeichen entsprechend Umwandlungstabelle in base64-Code
  - 4 base64-Codes ergeben 3 * 8 Bit der Originalmessage.

Wenn die Zeichen des Ausgangstexts nicht auf einen 24-Bit Block enden, wird der Ausgangstext mit Nullen aufgefüllt, die base64-codierte Nachricht mit „=“.

Übung 44: Suchen Sie sich im Netz einen base64-decoder. Nehmen Sie userid/password aus dem Netzwerkpaket der letzten Übung und decodieren sie es.

Sicherheitsmaßnahme gegen Angreifer im Netz

Falls die Gefahr besteht, dass ein Angreifer die Authentisierung im Netz abhört, gibt es zwei Schutzmöglichkeiten:
- Verschlüsselung der gesamten Kommunikation, z.B. mit SSL/TLS,
- Vermeiden, dass ein Passwort gesendet wird.

SSL/TLS ist ein Thema, das nicht hierher gehört, aber das Vermeiden kann erreicht werden mit dem Verfahren „Digest Authentication“. Dabei sendet der Server zunächst einen Zufallswert. Der Browser

kombiniert den Zufallswert mit dem Passwort und bildet daraus einen kryptografischen Hashwert, der dann über das Netz gesendet wird. Der Server kann den Hashwert prüfen, weil er das Passwort kennt.

Übung 45: Zeichnen Sie den Ablauf der Digest Authentication schematisch. Warum kann ein Angreifer nichts damit anfangen, wenn er den Hashwert mitliest.

Anmerkung: Leider kann ein Angreifer auch dieses Verfahren aushebeln, wenn er die Netzwerkpakete nicht nur lesen, sondern auch manipulieren kann. In diesem Fall kann er den Response Header dahingehend ändern, dass dieser nur Basic Authentication anfordert. Wenn der Browser daraufhin Userid/Passwort im Klartext (bzw. base64-codiert) sendet, hat der Angreifer sein Ziel erreicht.

8.7.1.4 Sicherheitslücke: Gestohlene Passwörter

Ein weiterer wichtiger Angriffskomplex ist der, dass ein Angreifer die Datei `.htpasswd` lesen kann, z.B. weil er beim Internet-Provider einen Mitarbeiter bestochen hat. Wenn die sogenannten „Credentials“ (Userid/Passwort) im Klartext in der Datei stehen, hat der Angreifer schon gewonnen.

Passwort-Hash

Üblicherweise geht man deshalb so vor, dass zumindest das Passwort nicht lesbar sein soll. Hier hat sich das Verfahren durchgesetzt, einen kryptografischen Hashwert zu verwenden, d.h. aus dem Passwort wird eine binäre Zeichenfolge gebildet, aus der das Passwort nicht ermittelt werden kann.

Bei der Anmeldung gibt der Benutzer sein Passwort an, dieses wird auf dem Server gehashed und dann mit dem gespeicherten Hash verglichen. Wenn diese identisch sind, ist der Benutzer authentisiert.

Angriffsmöglichkeiten

Ein Angreifer, der die Datei `.htpasswd` besitzt hat nun noch die folgenden Möglichkeiten:

- Er kann versuchen, das kryptografische Hashverfahren zu knacken. Das klappt meistens nicht.
- Er kann versuchen, mit möglichst großer Rechenpower (z.B. Rechnerverbund, Cloud-Computing, etc.), sogenannter roher Gewalt (**brute force**), möglichst viele Zeichenketten durchzuprobieren, ob sie als Passwort in Frage kommen. Für jede Zeichenkette ermittelt er selbst den kryprografischen Hashwert, wenn dieser mit dem übereinstimmt, der in der `.htpasswd` steht, hat er gewonnen. Wenn das Verfahren stark genug ist, dauert das jahrelang.
- Er kann versuchen eine riesige Datenbank aufzubauen, in der zu vorgegebenen Hashwerten mögliche Passwörter gespeichert sind, eine sogenannte **Rainbow Table**. Wenn er das Verfahren mehrere Jahre laufen lässt, hat er bereits einen großen Vorrat an geknackten Hashes. Angeblich gibt es fertige Rainbow Tables im Netz zu kaufen.
- Anstatt alle möglichen Zeichenkombination durchzuprobieren, kann ein Angreifer nur solche verwenden, die mit großer Wahrscheinlichkeit als Passwort benutzt werden, z.B. Namen, Begriffe aus einem Wörterbuch, einfache Ziffernkombinationen und/oder das Ganze kombiniert mit Sonderzeichen. Man spricht dann von einer **Wörterbuch-Attacke**.
- In Kombination sind Rainbow Tables für Wörterbuch-Attacken effizient.

Sicherheitsmaßnahmen

Aus diesen Angriffsmöglichkeiten sind folgende Sicherheitsmaßnahmen abzuleiten:

- Wir brauchen ein Verfahren, das noch nie geknackt wurde.
- Das Verfahren muss genügend lange Hashwerte erzeugen, um gegen Brute Force-Attacken und Rainbow-Tables sicher zu sein, solange die Rainbow Tables nicht auf der Basis von Wörterbüchern gebildet werden.
- Ausnahmsweise hilft ein Verfahren, das „langsam“ ist. Wenn ein Verfahren z.B eine Zehntelsekunde benötigt, um einen Hashwert auszurechnen, dann wäre dies für die normale Benutzung schnell genug, ein Angreifer könnte aber brute force nur 10 Versuche pro Sekunde machen. Ein Verfahren, das nur eine tausendstel Sekunde braucht, würde einem Angreifer einen 100-fachen Vorteil einräumen.
- Um Wörterbuch-basierende Rainbow Tables auszuhebeln, wird ein sogenanntes „Salt“ verwendet. Dafür wird das zu hashende Passwort mit einigen zufälligen Zeichen verlängert, damit es in

keinem Wörterbuch mehr vorkommt. Anschließend wird der Hashwert gebildet. Diesem werden die zufälligen Zeichen vorangestellt, weil der reguläre Prüfalgorithmus diese ebenfalls benötigt.

Bekannte Hash-Verfahren

Einige Hash-Verfahren aus Vergangenheit und Gegenwart sind crypt, MD5, SHA-1, SHA-3, SHA-256, SHA-512, BCRYPT, PBKDF2, scrypt

Übung 46: Finden Sie im Netz die Vor- und Nachteile der Hash-Verfahren crypt, MD5, SHA-1, SHA-3, SHA-256, SHA-512, BCRYPT, PBKDF2, scrypt für die Passwortverschlüsselung.

Basic und Digest Authentication verwenden übrigens nur die Verfahren crypt oder MD5, sind also aus diesem Grund ziemlich unsicher.

8.7.1.5 Cross-Site Authentication (XSA) Angriff

Vorbereitung

Wenn der Angreifer einen eigenen Server betreibt, kann er diesen so konfigurieren, dass dieser Basic Authentication erfordert. Beim Zugriff wird also der Benutzer nach Userid/Passwort gefragt.

Ablauf

Der Angreifer kann nun z.B. in ein Forum oder in ein Gästebuch einen Eintrag schreiben, in dem er den Link zu einem Bild einbindet, das er auf seinem eigenen Server gespeichert hat.

Wenn ein anderer Forums- (oder Gästebuch-)Benutzer das Bild laden möchte, geht das Browserfenster zur Basic Authentication auf. Wenn der Benutzer denkt, er sei immer noch auf der Seite des Forums, wird er evtl. seine Userid/Passwort-Kombination für das Forum eintippen. Diese landen dann aber beim Server des Angreifers und können dort gespeichert werden. Damit kommt der Angreifer in den Besitz der Credentials des Benutzers

Schutz

Foren, Gästebücher, Bewertungen, etc. sollten so konfiguriert werden, dass keine Links auf externe Daten eingebaut werden können.

8.7.2 Selbst entwickelte Passwort-Authentisierung

Selbst entwickelte Passwort-Authentisierung entspricht dem, was wir in der Vorlesung des letzten Semesters entwickelt haben.

Übung 47: Gehen Sie die oben genannten Sicherheitslücken noch einmal durch und beurteilen Sie, welche davon bei unserer selbst entwickelten Authentisierungslösung vorkommen können.

Antwort: Alle. Ein Angreifer könnte
- Userid/Passwort bei der Übertragung im Netz abhören,
- die Passwörter aus der Datenbank lesen,
- einen Benutzer evtl. mit XSA hereinlegen.

Mögliche Sicherheitsmaßnahmen wie oben:
- Netzwerkverschlüsselung mit SSL/TLS,
- Sichereres Authentisierungsverfahren, bei dem Passwörter nicht übertragen werden,
- Passwörter mit einem sicheren Verfahren hashen, bevor wir es in der Datenbank speichern.

Übung 48: Implementieren sie die genannten Sicherheitsmaßnahmen in die Webanwendung aus Ihrem AS-Praktikum.

8.7.3 Aufbau einer Session

8.7.3.1 Sessions als Dauer-Authentisierung

In der Originalfassung von HTTP gab es noch keine Sessions. Basic Authentication ging deshalb davon aus, dass Userid/Password vom Browser bei jedem Request an den Server wieder mitgeschickt

werden. Dadurch entstand natürlich die Sicherheitslücke, dass man nur einen einzigen Request mitlesen musste, um die Credentials bis zur nächsten Passwortänderung zu kennen.

Mit der Einführung von Sessions etablierte sich deshalb das Vorgehen, dass die Authentisierung an die Session gebunden wird. So müssen Userid/Passwort nur noch einmal über das Netz geschickt werden. Falls ein Angreifer dabei UserId / Passwort mitlesen kann, ist das Verfahren natürlich trotzdem gebrochen, aber die Gelegenheit dazu wird deutlich reduziert.

An die Stelle der übertragenen UserId / Passwort tritt nun eine SessionId, die natürlich ebenso geheim gehalten werden sollte, aber falls eine SessionId bekannt wird, ist die Sicherheit nur bis zum Ablauf der Session gebrochen und nicht bis zur nächsten Passwortänderung.

Zusammengefasst: Der Übergang von der Basic-Authentisierung zu Session-basierten Verfahren reduziert die Gelegenheiten zum Lesen und die Gelegenheit zum Ausnutzen eines Passworts.

8.7.3.2 Nicht-authentisierte und authentisierte Sessions

Eine Session wird z.B. von JSP beim ersten Zugriff auf eine Webseite aufgebaut. Aufbau einer Session bedeutet dabei, dass der Server und der Browser eine gemeinsame SessionId vereinbaren, anhand derer sie sich wiedererkennen. Dies erfolgt in der Regel automatisch, also ohne Zutun des Benutzers.

Zu diesem Zeitpunkt ist der Benutzer noch nicht authentisiert. Der Webserver erkennt zwar, dass ein nachfolgender Zugriff von demselben Benutzer (genauer: Von demselben Browser) kommt, er kann den Benutzer aber noch nicht sicher identifizieren. Die SessionId trägt nur die Information „selber Benutzer wie vorhin“.

Nachdem der Benutzer sich authentisiert hat, kann der Server die Userid mit der SessionId verbinden. Er weiß jetzt, wer der Benutzer ist. Die SessionId trägt jetzt also die Information „authentiserter Benutzer mit Userid …“. Je nachdem, was der Server noch alles über den Benutzer weiß, sind auch diese Informationen indirekt mit der SessionId verbunden, z.B. Alter, Wohnort, Kreditkartennummer, etc.

Wenn ein Angreifer also die SessionId einer authentisierten Session erfährt, kann ihm das Spielraum für Betrügereien einräumen. Wenn er dagegen die Id einer nicht-authentisierten Session erfährt, ist das nicht gefährlicher, als wenn der Angreifer selbst auf den Server zugegriffen hätte.

Zusammengefasst: Die Id einer authentisierten Session ist für einen Angreifer ein lohnendes Ziel.

8.7.3.3 Exkurs: Übertragung von SessionIds

Wie wir im JSP-Kurs gelernt haben, kann eine SessionId z.B. über Cookies, Hidden Fields oder URL Rewriting übertragen werden.

Meistens werden SessionIds über Cookies übertragen. Wir werden deshalb im weiteren Verlauf nicht immer explizit auf alle drei Fälle eingehen, wir können aber davon ausgehen, dass ein Angreifer jede der drei Methoden nutzen kann, um eine gestohlene SessionId zu nutzen.

Übung 49: Lesen Sie einige SessionIds mit Wireshark.

8.7.3.4 Angriffsmöglichkeit: Session Hijacking

Unter Session Hijacking versteht man, wenn ein Angreifer die Id einer authentisierten Session herausbekommt und diese für seine eigenen Zwecke nutzt, indem er bösartige Requests konstruiert, die aufgrund der gestohlenen Sessionid vom Server ausgeführt werden. Dabei handelt es sich um eine ganze Familie von Angriffsszenarien, denen wieder jeweils spezifische Sicherheitsmaßnahmen gegenüberstehen.

Angriffsszenarien

- Wenn die SessionId vom Server erzeugt wird, indem einfach jeweils nur ein Zähler hochgezählt wird, kann ein Angreifer diese einfach erraten.
- Wenn der Angreifer Zugriff auf die Netzwerkverbindung hat, kann er die SessionId mitlesen, falls diese im Netz unverschlüsselt übertragen wird.

- Wenn ein Angreifer Zugriff auf den lokalen Rechner des Benutzers hat, kann er die SessionId evtl. aus dem Browser oder von der Festplatte lesen.
- Wenn ein Angreifer Zugriff auf den Server hat, kann er die SessionId evtl. dort auslesen oder sogar derart manipulieren, dass der Benutzer eine SessionId bekommt, die vom Angreifer vorgegeben wird.

Übung 50: Ermitteln Sie via Internet was DNS-Spoofing und DNS-Poisoning bedeutet und in welchem Zusammenhang dies mit Session Hijacking steht.

Sicherheitsmaßnahmen

Die Sicherheitsmaßnahmen korrespondieren direkt mit den Angriffsszenarien:
- SessionIds müssen vom Server so erzeugt werden, dass sie nicht erraten werden können. Am besten geht das, wenn ein kryptografischen Zufallsgenerator verwendet wird, der genügend lange SessionIds erzeugt.
- Um Angreifer im Netzwerk auszuschließen, sollte die Netzwerkverbindung verschlüsselt sein, z.B. mit SSL/TLS.

Übung 51: Warum bringt es nichts, wenn einfach die SessionId verschlüsselt übertragen wird?
- Angreifer sollten keinen Zugriff auf Browser oder Server haben. Dies ist leichter gesagt als getan, weil wir nicht wissen, wer Angreifer ist. Z.B. haben im Unternehmensumfeld viele Mitarbeiter Zugriff auf PCs in einer Abteilung. Es sind also für Clients und Server folgende elementaren Sicherheitsmaßnahmen denkbar:
  - im Unternehmen Einschränkungen, welche Personen Zugang zu und Zugriff auf PCs oder Server haben dürfen.
  - Generell Schutz vor Viren, Trojanern, etc., die einen Fernzugriff auf Clients und Server ermöglichen könnten.

8.7.3.5 Angriffsmöglichkeit: Session Fixation

Unter Session Fixation versteht man ein Szenario der heimlichen Delegation, allerdings erfolgt der Angriff erst später direkt.

Ablauf
- Ein Angreifer verbindet sich mit einem Server und erhält die SessionId einer nicht-authentisierten Session.
- Der Angreifer schiebt einem Benutzer einen Request mit „seiner“ SessionId unter, der den Webserver dazu bringt, nach Authentisierungsinformationen zu fragen.
- Der Benutzer logged sich ein, wodurch sich die nicht-authentisierte Session in eine authentisierte Session verwandelt, deren Id der Angreifer immer noch kennt.
- Anschließend kann der Angreifer innerhalb der Session Requests ausführen, die unter der Berechtigung des Benutzers laufen.

Sicherheitsmaßnahme

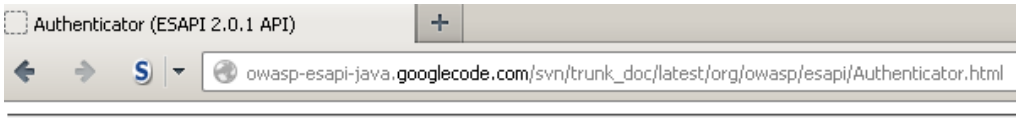
Die Sicherheitsmaßnahme gegen Session Fixation ist einfach:
- Jeder Server muss nach einer erfolgreichen Authentisierung (login) **immer** eine neue SessionId vergeben. Wenn der Angreifer dann nicht zusätzlich das Netzwerk abhören kann, kennt er die Id der authentisierten Session nicht.

Da die SessionId automatisch vergeben wird, reicht es, wenn der Server die alte SessionId invalidiert, z.B. mit dem Java-Befehl `HttpSession.invalidate()`. Eine neue SessionId entsteht dann automatisch.

8.7.4 ESAPI-Authenticator

8.7.4.1 Java Interface

Das Java Interface „Authenticator“ soll dabei helfen, eine sichere Authentisierung zu entwickeln. Es besteht aus den folgenden 19 Methoden, die eine Implementierung ausprogrammieren muss (http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/Authenticator.html):



Method Summary

void	**changePassword**(User user, java.lang.String currentPassword Changes the password for the specified user.
void	**clearCurrent**() Clears the current User.
User	**createUser**(java.lang.String accountName, java.lang.String Creates a new User with the information provided.
boolean	**exists**(java.lang.String accountName) Determine if the account exists.
java.lang.String	**generateStrongPassword**() Generate a strong password.
java.lang.String	**generateStrongPassword**(User user, java.lang.String oldPass Generate strong password that takes into account the user's informatio
User	**getCurrentUser**() Returns the currently logged in User.
User	**getUser**(long accountId) Returns the User matching the provided accountId.
User	**getUser**(java.lang.String accountName) Returns the User matching the provided accountName.
java.util.Set	**getUserNames**() Gets a collection containing all the existing user names.
java.lang.String	**hashPassword**(java.lang.String password, java.lang.String a Returns a string representation of the hashed password, using the accou
User	**login**() Calls login with the *current* request and response.
User	**login**(javax.servlet.http.HttpServletRequest request, javax This method should be called for every HTTP request, to login the curr
void	**logout**() Logs out the current user.
void	**removeUser**(java.lang.String accountName) Removes the account of the specified accountName.
void	**setCurrentUser**(User user) Sets the currently logged in User.
void	**verifyAccountNameStrength**(java.lang.String accountName) Ensures that the account name passes site-specific complexity requirem
boolean	**verifyPassword**(User user, java.lang.String password) Verify that the supplied password matches the password for this user.
void	**verifyPasswordStrength**(java.lang.String oldPassword, java. Ensures that the password meets site-specific complexity requirements,

Ein Blick auf die Methoden zeigt uns, dass der ESAPI Authenticator sich nicht nur um den Login, sondern auch um das Anlegen von Usern kümmern möchte. Entsprechend der Dokumentation verbirgt sich hinter der login(…) Methode auch die Sicherheit der Session.

8.7.4.2 Authenticator-Klassen

ESAPI stellt zwei Klassen bereit:

- AbstractAuthenticator implementiert 9 der 19 Methoden. Entwickler können davon erben und brauchen dann nur noch 10 Methoden selbst implementieren.
- FileBasedAuthenticator ist eine Klasse, die das gesamte Authenticator-Interface implementiert. Die User werden dabei in einer Datei gespeichert.

Im Unternehmensumfeld sollen die User meistens in einer Datenbank oder einem Verzeichnisdienst gespeichert werden. In diesem Fall müssten Entwickler eine eigene Authenticator Klasse schreiben, z.B. indem sie von AbstractAuthenticator oder FileBasedAuthenticator erben.

Übung 52: Setzen Sie ein eigenes Authentisierungsverfahren mittels ESAPI-Authenticator auf.

8.8 Weitere Websicherheitsthemen

CSRF: Wie Fixation, nur ohne Session? REST?

Direct Object Reference

Access Control

Übungsverzeichnis