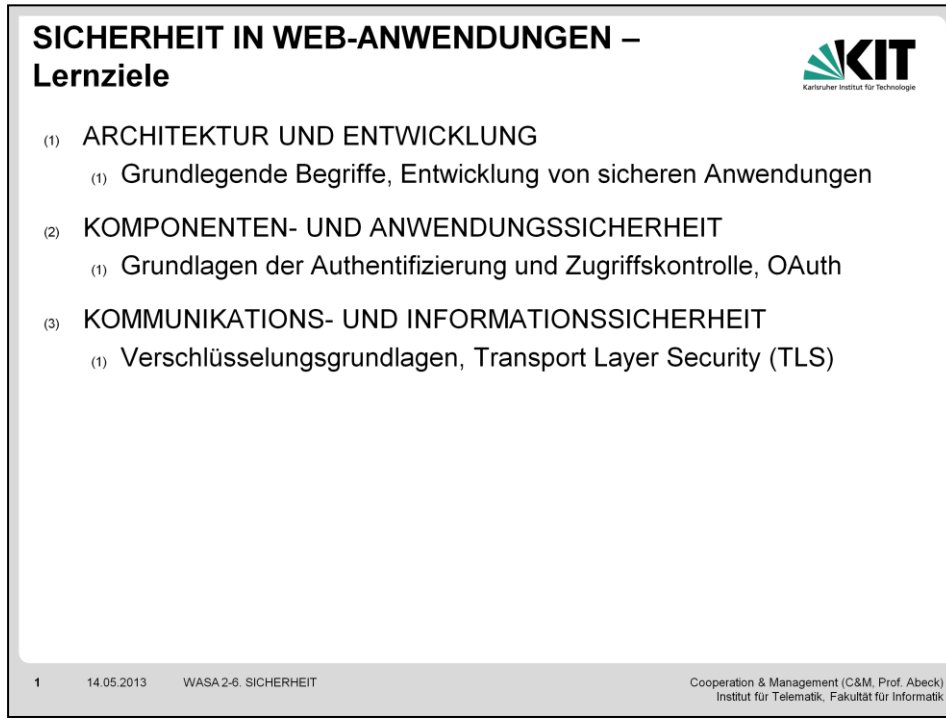
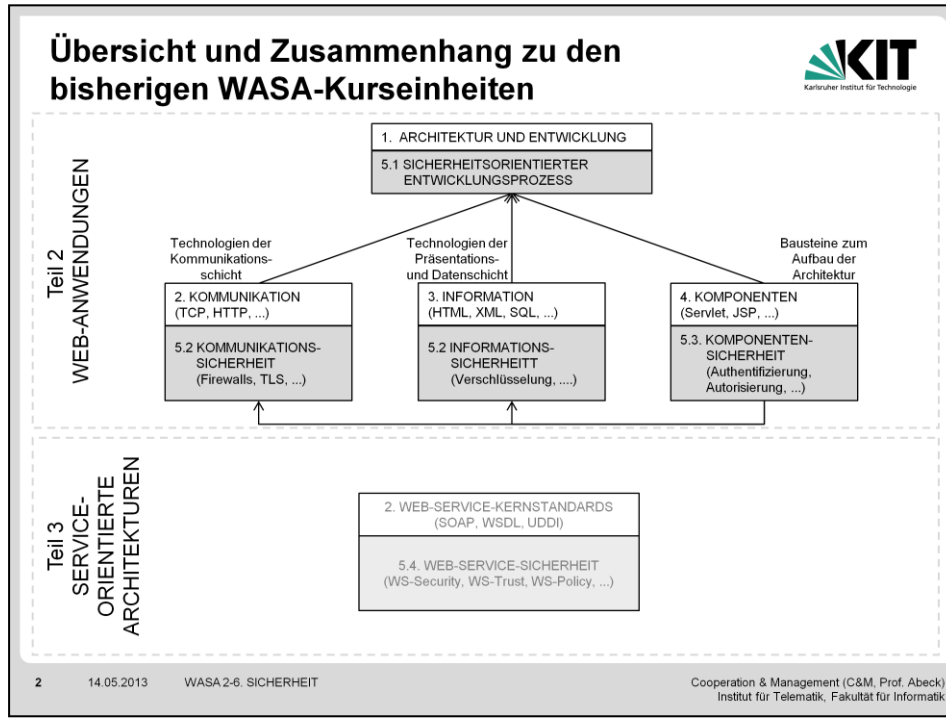


In dieser Kurseinheit soll die Grundlage für das Verständnis von Sicherheit in Web-basierte Anwendungen aufgebaut werden.

(1) Die Thematik wird anhand grundlegender Begriffe der Sicherheitsdomäne eingeführt. Weiterführende Themen sollen anhand eines sicherheitsbasierten Entwicklungsprozesses erläutert werden, welche hier vorgestellt wird.

(2) Authentifizierung und Zugriffskontrolle sind weitere grundlegende Sicherheitsmechanismen. Die Konzepte und deren Anwendung werden in diesem Teil behandelt.

(3) In diesem Teil werden die wesentlichen Konzepte von Verschlüsselungsverfahren vorgestellt und deren Einsatz zur Sicherung von Informationen und Nachrichtenaustausch präsentiert.



Die Kurseinheit "Sicherheit" betrachtet die bisher präsentierten Kurseinheiten unter dem Aspekt der Sicherheit. Ziel der Lerneinheit soll es sein, ein Verständnis dafür zu entwickeln, wie sich der nichtfunktionale bzw. qualitätsorientierte Aspekt der Sicherheit in einem sicherheitsorientierten Entwicklungsprozess einbauen lässt, um ein möglichst weitgehend abgesicherte Web-Anwendung zu entwickeln.

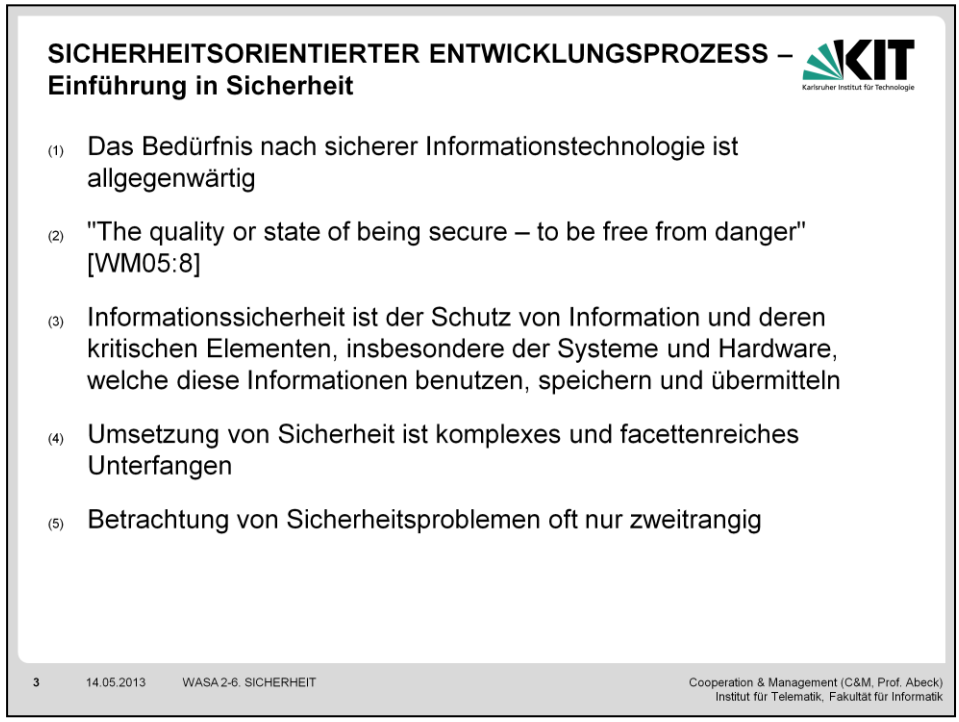
(5.1 SICHERHEITSORIENTIERTER ENTWICKLUNGSPROZESS) Nach einer grundlegenden Einführung in die Sicherheitsthematik, sollen in diesem Teil die Aktivitäten und Werkzeuge eines sicherheitsorientierten Entwicklungsprozesses aufgezeigt werden.

(5.2 KOMMUNIKATIONSSICHERHEIT) Analog zu den vorgestellten Themen in der Kurseinheit 2-2 KOMMUNIKATION, beinhaltet dieser Teil Sicherheitsmechanismen, welche auf den Ebenen von TCP und HTTP zum Zuge kommen.

(5.2 INFORMATIONSSICHERHEIT) In diesem Teil geht es vorallem um kryptographische Methoden zur Absicherung von Informationen.

(5.3 KOMPONENTENSICHERHEIT) Die in heutigen Web-Anwendungen Sicherheitsmechanismen von Authentifzierung und Autorisierung sowie Zugriffskontrolle werden vorgestellt und mittels konkreten Beispielen demonstriert.

(5.4 WEB-SERVICE-SICHERHEIT) In einem Ausblick werden die wesentlichen Mechansimen zur Absicherung von Web-Services vorgestellt.



(1) Schon vor der automatisierten Informationsverarbeitung mittels Rechenmaschinen wurden Mittel zur Schutz von sensiblen Informationen genutzt, um sie vor Einsicht durch nicht-autorisierten Personen zu schützen. Seit dem Betrieb der ersten Mainframes, der Entwicklung des Personal Computers sowie der Entwicklung des Internets wuchsen die Herausforderungen um die Sicherheit der verarbeiteten Informationen zu gewährleisten.

(2) Um Gefahren abzuwenden, werden Schutz- bzw. Sicherheitsmaßnahmen errichtet. Diese sind meist mehrschichtig um alle Gefahren abzuwehren und einen größtmöglichen Schutz zu bieten. Während dies für beliebige Dinge der realen Welt gilt, können für Informationssystem in Unternehmen physische, persönliche, operationelle, Kommunikations- und Netzwerksicherheit als Beispiele für solche Schichten gezählt werden [WM05: 8].

(3) Diese Definition umfasst die vorherigen Beispiele und betrachtet zusätzlich die Verwaltung von Informationssicherheit [WM05: 8].

(4) Die Realisierung von Informationssicherheit erweist sich nicht selten als komplexes Unterfangen, da viele Facetten berücksichtigt werden müssen. Zum einen sind vielfältige Sicherheitsmaßnahmen entstanden, deren Funktionsweise und Anwendungsbereich meist nur von Sicherheitsexperten verstanden wird. Zum anderen müssen alle Ebenen eines Informationssystems berücksichtigt werden. Dies beinhaltet neben der funktionalen Anwendungsschicht auch die darunterliegenden Ausführungsumgebungen und Netzwerke sowie die darüber liegenden Geschäftsprozesse und menschlichen Nutzen. Allerdings dürfen die Sicherheitsmaßnahmen die Benutzbarkeit der Informationssysteme nur unwesentliche beeinträchtigen. Auch diese Abwägung erschwert die Einführung von Sicherheitsmaßnahmen.

(5) Trotz des Sicherheitsbedürfnisses kommt es häufig dazu, dass aufgrund der Komplexität der Sicherheitsdomäne, Sicherheit oft nur zweitrangig bei der Entwicklung von Informationssystemen betrachtet wird. Meist werden erst nach der erfolgreichen Umsetzung der geschäftlichen Funktionen Sicherheitsprobleme betrachtet und dann oft nur unzureichende Maßnahmen umgesetzt.

Referenzen

[WM05] M. E. Whitman and H. J. Mattord, Principles of Information Security, 2nd ed. Boston: Thomson Course Technology, 2005, p. 576.

(1) Um Sicherheitsmaßnahmen in einem Informationssystem zu implementieren, empfiehlt sich ein sicherheitsbasierter Entwicklungsprozess. Hierzu wird ein bestehender fachlichen Entwicklungsprozess um Aktivitäten ergänzt, so dass Sicherheitsmaßnahmen in einem methodischen Vorgehen entwickelt werden können. Ein solches Vorgehen ist jedoch meist nicht etabliert, was dazu führt, dass Sicherheitsmaßnahmen nach der Implementierung der fachlichen Funktionalität umgesetzt werden. Dieses "penetrate and patch"-Vorgehen, d.h. die Sicherheitslücken relativ zufällig entdecken und durch nachträglich eingespielte Aktualisierungen schließen, ist jedoch nicht der Sicherheit einer Anwendung förderlich. Denn wie an der Abbildung zu sehen, erfordert ein nachvollziehbares und qualitätsgesichertes Vorgehen zur Entwicklung sicherer Software vielfältige Methoden und Werkzeuge in allen Entwicklungsphase.

(Anforderungsanalyse, Missbrauchsanwendungsfälle, Sicherheitsanforderungen) UML-Anwendungsfälle sind ein erprobtes Mittels zur Spezifizierung von Anwendungsfällen und Szenarien, aus welchen funktionale Anforderungen abgeleitet werden. Ein ähnlichen Ansatz nutzen Missbrauchsanwendungsfälle um die Bedrohungen für Anwendungsfälle zu spezifizieren. Aus den so dargestellten Bedrohungen für bzw. Missbrauch von Anwendungsfällen werden im Anschluss Sicherheitsanforderungen abgeleitet.

(Entwurf, Sicherheitsmuster, Sicherheitsarchitektur) In der Entwurfsphase werden zu den Anforderungen passende Sicherheitsmaßnahmen modelliert. Sicherheitsmuster helfen, aus dem bereits existierenden Sicherheitswissen erprobte Lösungen zu bekannten Sicherheitsproblemen auszuwählen. Ziel des Entwurfs ist es, sowohl die konzeptionelle Architektur der Sicherheitsfunktionalität als auch deren Feinentwurf zu spezifizieren. Sicherheitsrichtlinien spezifizieren dabei die Funktionsweise von einzelnen Sicherheitskomponenten im Detail.

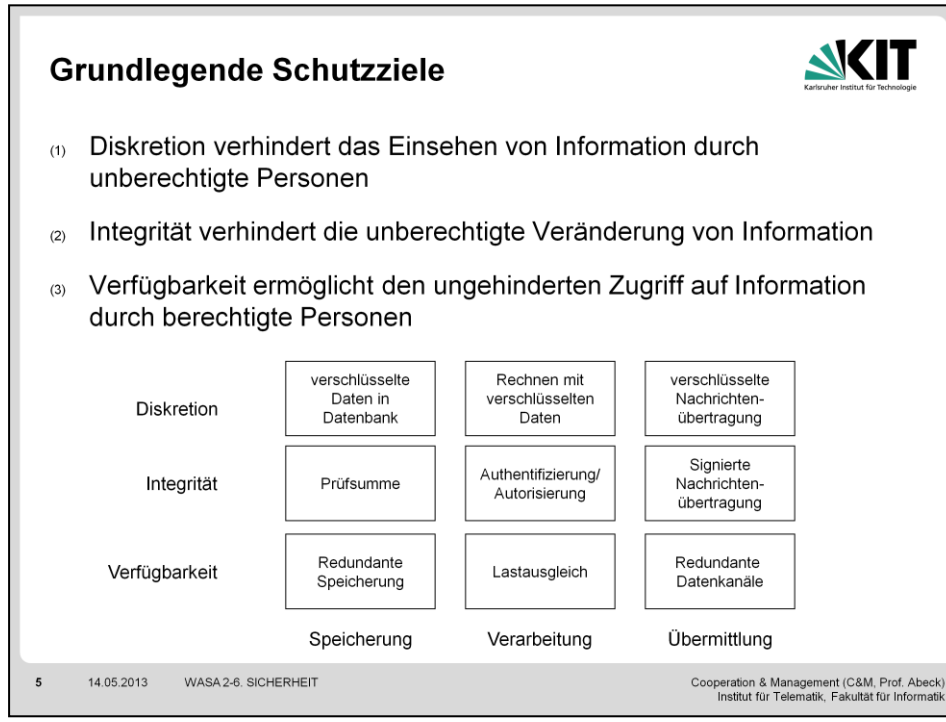
(Implementierung, Sicherheitsstandards, Sicherheitsmechanismen, ...) Die Umsetzung von Sicherheitsmaßnahmen sollte nur im Ausnahmefall durch eine Eigenimplementierung durchgeführt werden. Stattdessen sollten bestehende Sicherheitsstandards, -produkte oder -rahmenwerke eingesetzt werden, welche ein ausgereifte Implementierung von Sicherheitsmechanismen bereitstellen. Bei der Umsetzung von sicherheitsrelevanten sowie auch fachlichen Quelltext müssen Konventionen eingehalten werden, welche die Einführung von sicherheitskritischen Fehlern (engl. bugs) vorbeugen. Dieser Aspekt wird jedoch in dieser Vorlesung nicht weiter vertieft.

(Tests, Penetrationstest, statische Analyse) In der Testphase werden die umgesetzten Software-Artefakte auf Sicherheitslücken überprüft. Mittels Penetrationstests werden bekannte Angriffsszenarien simuliert und die Reaktion der entwickelten Software untersucht. Statische Analyse untersucht den Quelltext der Anwendung auf bekannte Fehler verursachende Quellcodekonstrukte.

(Verwundbarkeiten, Angriffe, Betrieb, etc.) Im Betrieb der entwickelten Anwendung müssen neue Angriffsmethoden erkannt und Verwundbarkeiten analysiert werden, welche durch die bereits etablierten Sicherheitsmaßnahmen nicht abgedeckt wurden. Dementsprechend muss die Sicherheitsfunktionalität der aktuellen Gefährdungslage angepasst werden. Der sicherheitsbasierte Entwicklungsprozess wird dabei erneut durchlaufen.

Referenzen

[Mc04] G. McGraw, “Software Security,” IEEE Security & Privacy, vol. 2, no. 2, pp. 80–83, 2004.



Der Schutzbedarf von Informationsressourcen is bedingt durch den mit ihnen verbundenen unternehmerischen Wert [Ec09:6f; WH05:9ff]. Es existieren unterschiedliche Eigenschaften von Informationen, welche es durch Sicherheitsmaßnahmen zu schützen gilt. Die folgenden drei grundlegenden Eigenschaften sind auch durch die Abkürzung C.I.A. (engl. confidentiality, integrity, availability) bekannt.

(1) Durch die Diskretionseigenschaft wird verhindert, das unbefugte Personen oder Systeme, Informationen einsehen können. Somit wird eine Veröffentlichung und Weitergabe verhindert.

(2) Die Veränderung von Information ist meist nur durch berechtigte Personen oder Systeme erwünscht, z.B. soll der Kontostand eines Bankkontos nur durch den Inhaber des Kontos durch eine Auszahlung verringert werden können. Für unterschiedliche Informationen und informationsverarbeitende Systeme sind unterschiedliche Maßnahmen möglich, z.B. können durch Prüfsummen Veränderungen in Nachrichten und Dateien aufgespürt werden.

(3) Unter Verfügbarkeit versteht man die ungehinderte Bereitstellung von Informationen für berechtigte Nutzer in einem geeigneten Format. Die Feststellung ob eine Person oder ein System entsprechende Berechtigung besitzt, läßt sich durch Authentifizierung und Autorisierungsmaßnahmen feststellen. Somit erfüllen diese mehrere Schutzziele.

(Speicherung, Verarbeitung, Übermittlung) Es ist wichtig zu verstehen, dass dies abstrakte Eigenschaften sind, die je nach Art der Information in unterschiedlicher Ausprägung erreicht werden können. In der Abbildung werden drei Zustände unterschieden, Speicherung, Verarbeitung und Übermittlung, welche Information annehmen kann. In jedem Zustand sind unterschiedliche Angriffe und unterschiedliche Sicherheitsmaßnahmen möglich, welche die Eigenschaften beeinträchtigen bzw. erfüllen können. Auch erfüllen einige Sicherheitsmaßnahmen mehrere Schutzziele.

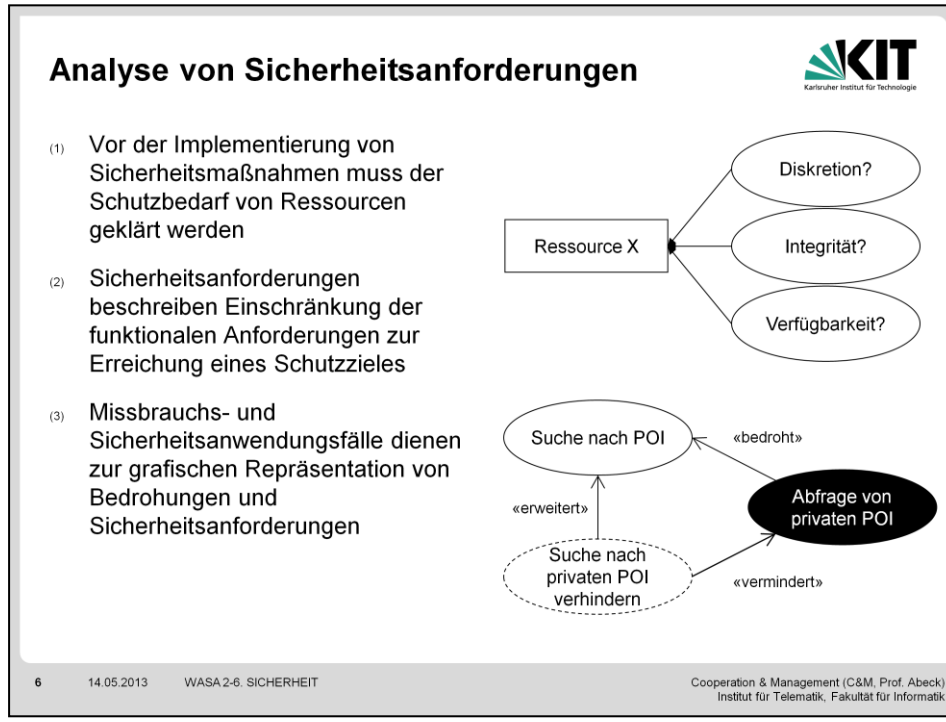
Abkürzungen

C.I.A. Diskretion (engl. confidentiality), Integrität (engl. integrity), Verfügbarkeit (engl. availability)

Referenzen

[Ec09] C. Eckert, IT-Sicherheit, 6th ed. München: Oldenbourg Wissenschaftsverlag, 2009, p. 981.

[WM05] M. E. Whitman and H. J. Mattord, Principles of Information Security, 2nd ed. Boston: Thomson Course Technology, 2005, p. 576.



(1) Eine wesentliche Voraussetzung zur Umsetzung von Sicherheitsmaßnahmen stellt die Erhebung und Spezifzierung von Sicherheitsanforderungen dar. Durch die Erhebung wird der Schutzbedarf der zu entwickelnden Anwendung festgestellt. Hierdurch wird ermittelt, von welche Art die umzusetzenden Maßnahmen sind und welchen Umfang sie einnehmen.

(2) Sicherheitsanforderungen werden häufig mit architekturellen Sicherheitsmaßnahmen ausgedrückt. Hierdurch werden jedoch bereits in der Analysephase Entwurfsentscheidungen getroffen, obwohl die fachliche Architektur noch nicht entworfen wurde. Stattdessen beinhalten Sicherheitsanforderungen das zu schützende Objekt und den Grund für den Schutz im Sinne der Sicherstellung eines Schutzzieles. Durch die Relation zu einer fachlichen Ressource bzw. zu einer funktionalen Anforderungen lassen sich die Sicherheitsanforderungen spezifisch genug ausdrücken, so dass im Entwurf auf geeignete Sicherheitsmaßnahmen geschlossen werden kann [TJ+08].

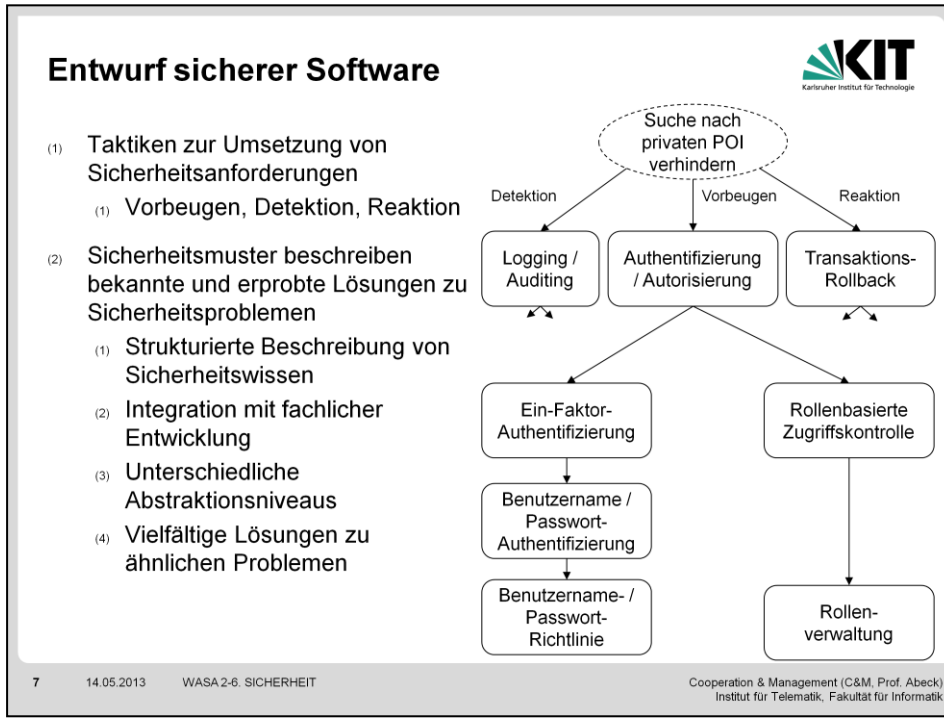
(3) Eine sich an der Erhebung von Anforderungen durch Anwendungsfälle orientierende Methode zur Ermittlung von Sicherheitsanforderungen sind Missbrauchs- [SO05] und Sicherheitsanwendungsfälle [Fi03]. Während Missbrauchsanwendungsfälle zur Beschreibung von Bedrohungen dienen, werden Sicherheitsanwendungsfälle zur Beschreibung von Sicherheitsanforderungen genutzt. Diese Werkzeuge dienen lediglich der grafischen Repräsentation und bieten keine Vorgehensweise zur Erhebung der Bedrohungen und Anforderungen an.

Referenzen

[Fi03] D. G. Firesmith, “Security use cases,” Journal of Object Technology, vol. 2, pp. 53–64, May 2003.

[So05] G. Sindre and A. L. Opdahl, “Eliciting Security Requirements with Misuse Cases,” Requirements Engineering, vol. 10, no. 1, pp. 34–44, 2005.

[TJ+08] I. A. Tøndel, M. G. Jaatun, and P. H. Meland, “Security requirements for the Rest of Us: A Survey,” IEEE Software, vol. 25, no. 1, pp. pp. 20–27, 2008.



(1) Aus den Sicherheitsanforderungen müssen anschließend geeignete Sicherheitsmaßnahmen abgeleitet werden. Je nach Schutzbedarf ergeben sich verschiedenen Möglichkeiten, welche grob in sogenannte Taktiken eingeteilt werden können:
(1.1) Vorbeugen schützt Sicherheitsattribute durch direkte Verhinderung von ungewünschten bzw. kompromittierenden Ereignissen. Detektion identifiziert ungewünschte Ereignisse und stellt entsprechende Benachrichtigungen bereit. Reaktive Maßnahmen adressieren ungewünschte Ereignisse nach ihrem Eintreten.

(2) Sicherheitsmuster beschreiben erprobte Lösungen zu wiederauftretenden Sicherheitsproblemen in einem spezifischen Kontext. Die Lösung besteht aus einer Menge von interagierenden abstrakten Rollen, welche in verschiedene konkrete Entwurfsstrukturen arrangiert werden können [SF+05: 31]. Bei der Integration von fachlicher und sicherheitsbasierter Entwicklung ist ein Musteransatz vorteilhaft [SF+05:34]:
(2.1) Muster beschreiben grundlegendes Sicherheitswissen in strukturierter und nachvollziehbarer Form
(2.2) Muster werden bereits in der fachlichen Entwicklung genutzt. Sicherheitsmuster orientieren sich an der für Softwareentwickler und Architekturen gewohnten Beschreibungsform. Somit einheitliche Behandlung von fachlichen und Sicherheitsaspekten möglich.
(2.3) Sicherheitsmuster können sowohl für die detaillierte Implementierung von Sicherheitsmaßnahmen als auch für abstrakte logische Strukturierung eingesetzt werden. Hierdurch wird eine iterative musterbasierte Verfeinerung der Sicherheitslösung ermöglicht.
(2.4) Obwohl Sicherheitsmuster Lösungen bereitstellen, sollten die zu behandelnden Probleme bzw. Sicherheitsanforderungen nicht außer acht gelassen werden. Ohne ein wohldefiniertes Sicherheitsproblem besteht die Gefahr, dass ein unzureichendes Sicherheitsmuster eingesetzt wird [SF+05:35].

Referenzen:

[SF+05] M. Schumacher, E. B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns. Chichester, England: John Wiley & Sons Ltd, 2005, p. 565.

LZ ARCHITEKTUR UND ENTWICKLUNG – ÜA Entwicklung sicherer Software

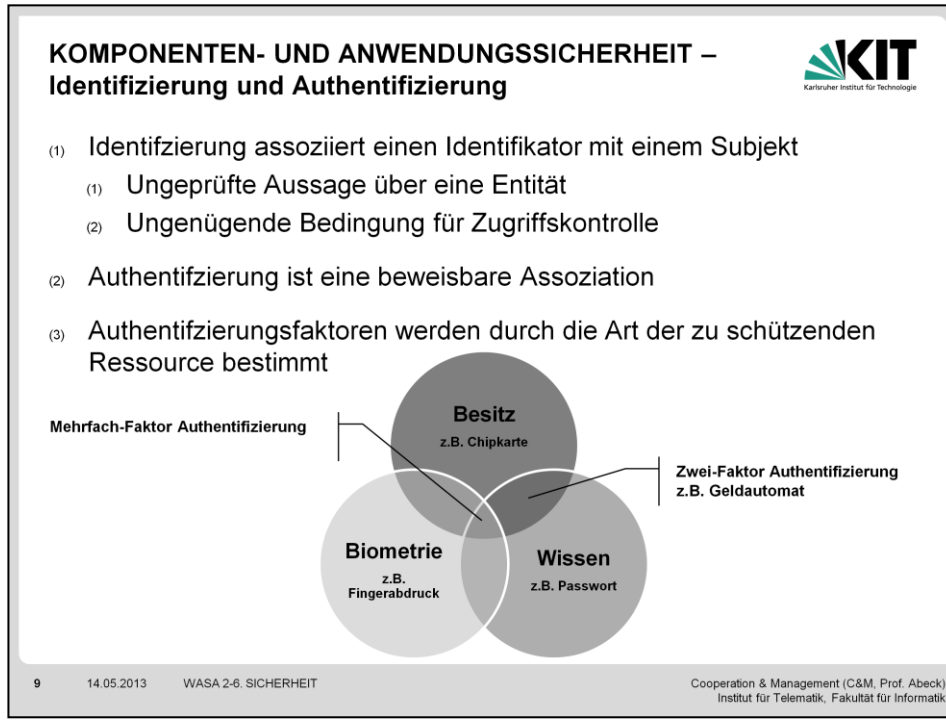
(1) Wie lauten die grundlegenden Schutzziele für Informationen? Erklären Sie sie an einem passenden Beispiel.

(2) Ein sicherheitsbasierter Software-Entwicklungsprozess ist ...
   (1) ein eigenständiger Entwicklungsprozess.
   (2) ein erweiterter Entwicklungsprozess.
   (3) nicht existent.

(3) Welche Methode zur Beschreibung von Angriffen und Sicherheitsmaßnahmen kennen Sie?

(4) Welche Ansätze zur Umsetzung von Sicherheitsmaßnahmen gibt es?

(5) Warum ist der Einsatz von Sicherheitsmustern in der Entwurfsphase sinnvoll?



(1) Nutzer agieren in einem Informationssystem mittels einer digitalen Identität. Eine Identifizierung eines Nutzers (bzw. allgemein eines Subjektes) an einem Informationssystem stellt eine Verbindung zwischen ihm und der von ihm genutzten Identität her. Dabei ist diese Assoziation ungeprüft, d.h. der Nutzer gibt die Verbindung an, kann gegenüber dem System jedoch noch nicht beweisen, das er berechtigt ist, mittels dieser Identität im System zu arbeiten. Eine solche Verbindung kann z.B. durch Angabe eines Benutzernamens erfolgen. Identifizierung ist nicht ausreichend um eine Zugriffsbeschränkung durchzuführen, da die Identitäten gefälscht werden können. Ein Beispiel hierfür ist ein IP-Spoofing-Angriff, bei welchem die IP-Adresse verändert wird, so dass die Identität eines anderen Kommunikationspartners vorgetäuscht wird..

(2) Eine Authentifizierung stellt eine beweisbare Assoziation eines Subjektes mit einer digitalen Identität dar. Authentifizierung ist konzeptionell ein zweistufiger Prozess, bei welchem zunächst ein Identifizierung stattfindet und diese anschließend authentifiziert wird. In der Praxis ist diese Zweistufigkeit selten sichtbar, so wird bspw. durch die Authentifizierung durch Benutzername-Passwort sowohl eine Identifizierung mit der Identität des Benutzernamens durchgeführt und diese Assoziation durch die Angabe des zugehörigen Passwortes auch bewiesen.

(3) Die zur Authentifizierung möglichen Methoden werden in drei Kategorien, sogenannten Faktoren, eingeteilt. Besitzfaktoren ermöglichen die Authentifizierung durch die Vorlage eines Gegenstandes, z.B. Kreditkarte, KIT-Karte, etc. Biometrische Verfahren authentifizieren Subjekte anhand ihrer inhärenten Eigenschaften, z.B. Fingerabdruck, Iris-Scan, etc. Wissensfaktoren authentifizieren Subjekte durch die Abfrage von geheimen Wissen, z.B. Passwörter, PIN, Transaktionsnummern etc. Die Kombination von mehreren Authentifizierungsfaktoren ist möglich und kann je nach Schutzbedarf einer Ressource eingesetzt werden. Somit kann es zu einer Zwei-Faktor-Authentifizierung, z.B. Benutzername-Passwort und KIT-Karte, oder Mehrfachfaktor-Authentifizierung kommen [SN+06, p. 51f; Wi05, p. 50ff].

Abkürzungen

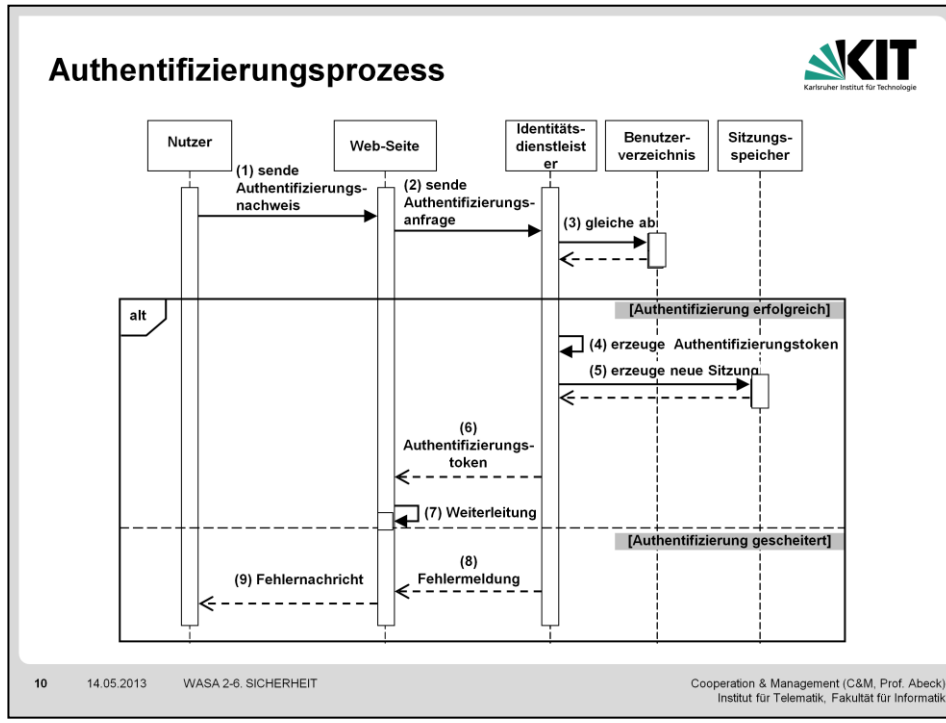
IP Internet Protocol

Referenzen

[Ca04] L. Jean Camp, *Digital Identity*, IEEE Technology and Society Magazine, 2004, vol. 23 (4), pp. 34-41.

[SN+06] C. Steel, R. Naappan, R. Lai, *Core Security Patterns - Best Practices and Strategies for J2EE, Web Services, and Identity Management*, Pearson Education, Upper Saddle River, NJ, 2006.

[Wi05] P. Windley, *Digital Identity*, O'Reilly, August 2005.



Das abgebildete UML-Sequenzdiagramm zeigt den konzeptionellen Ablauf bei einer Authentifizierung.

(1) Ein Benutzer ruft eine Ressource, z.B. eine Web-Seite auf. Um Zutritt zur Ressource zu erhalten muss er sich zunächst authentifizieren. Zur Eingabe seines Benutzernamens und Passwortes wird ihr ein Anmeldeformular angezeigt.

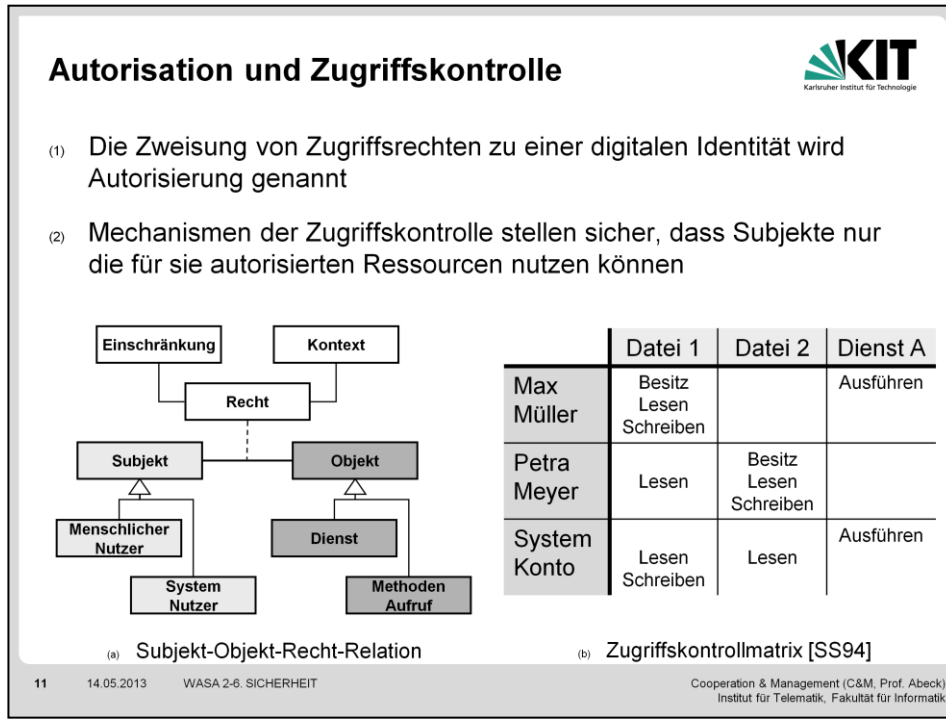
(2) Die Web-Seite führt die Authentifizierung nicht selbst durch, sondern nutzt den Authentifizierungsdienst eines ihr vertrauenswürdigen Identitätsdienstleisters.

(3) Die mitgesendeten Authentifizierungsinformationen werden durch den Identitätsdienstleister gegen die in einem Benutzerverzeichnis abgelegten Benutzerdaten abgeglichen.

(4)-(5) Bei erfolgreicher Authentifizierung wird dem Benutzer ein zeitbegrenztes Authentifizierungstoken ausgestellt, welches auch den Beginn einer Sitzung beschreibt.. Dies bestätigt die erfolgreiche Authentifizierung des Clients gegenüber des Servers und kann bei weiteren Zutrittsanfragen des Clients benutzt werden. Hierdurch wird die kritische Übermittlung von Benutzername und Passwortes umgangen. Die Sitzungsinformation wird in einem Sitzungsspeicher abgelegt.

(6)-(7) Das Token wird dem Benutzer zugestellt, z.B. als Sitzungs-Cookie im Web-Browser. Anschließend wird der Benutzer zur angefragten Ressource weitergeleitet.

(8)-(9) Schlägt die Authentifizierung fehl, so wird eine Fehlermeldung angezeigt und der Zutritt zur Ressource verweigert.



(1) Autorisierung stellt somit eine Entscheidung dar, einem Subjekt bzw. seiner digitalen Identität die Ausführung einer Aktion, z.B. lesen, schreiben, etc., auf einer Ressource durchführen zu lassen. Die Problematik in der Rechtezuweisung liegt in deren Verwaltung. In einem Unternehmen mit mehreren hundert oder mehr Mitarbeitern, müssen effizient Wege gefunden werden, um die einzelnen Rechte den Mitarbeiter zuweisen zu können. Eine erfolgreiche Authentifizierung in Kombination mit Autorisierung stellt eine wesentliche Voraussetzung für eine effektive Zugriffskontrolle dar

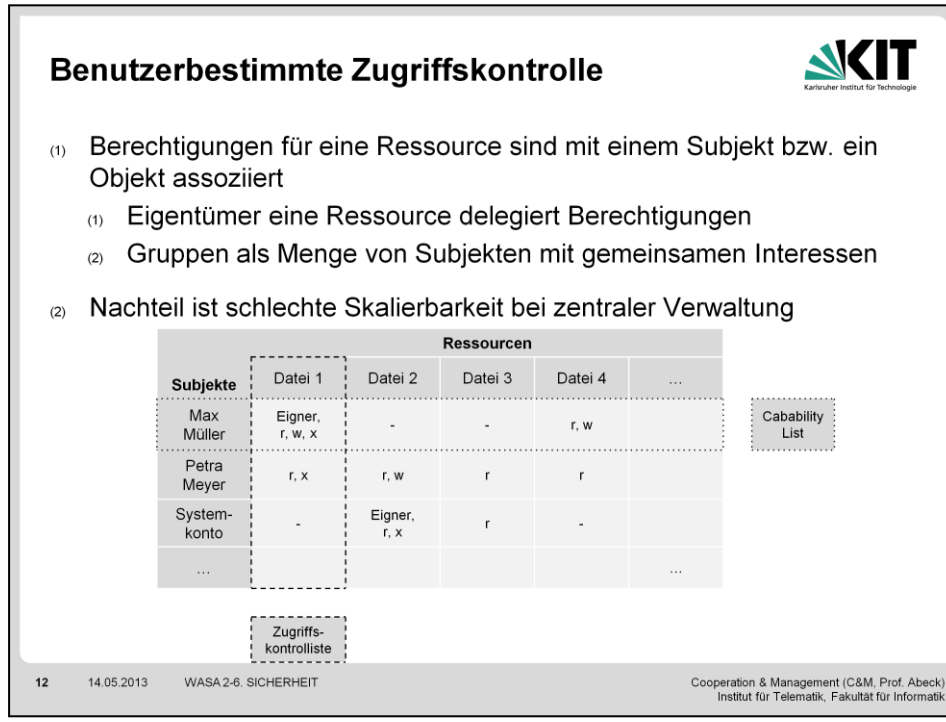
(2) Autorisierung ist der Zugriffskontrolle konzeptionell vorgelagert und beschäftigt sich mit der Zuweisung und Verwaltung von Rechten. In der Praxis wird leider selten zwischen den beiden unterschieden. Zugriffskontrollmechanismen überprüfen zur Anfragezeit, ob ein Subjekt die erforderlichen Rechte besitzt, um auf eine Ressource zuzugreifen.

(Zugriffskontrollmatrix) Es existieren unterschiedliche Arten von Autorisierung und Zugriffskontrolle, welche alle auf dem konzeptionellen Basismodell der Rechtezuweisung, der Zugriffskontrollmatrix basieren. Wie in der Abbildung dargestellt bezeichnen die Zeilen die Subjekte bzw. deren digitalen Identitäten, während die Spalten die Ressourcen darstellen. In den Feldern werden die Rechte eingetragen, welche ein Subjekt auf eine Ressource erhält. Autorisierung und Zugriffskontrolle wird in der Praxis selten durch eine Matrix realisiert, da diese Datenstruktur zu ineffizient ist. [SS04].

Autorisierung und Zugriffskontrolle legen ein abstraktes Modell zugrunde, in welchem ein Subjekt auf ein Objekt zugreifen möchte und dieser Zugriff jedoch nur durch das Vorhandensein der notwendigen Rechte möglich ist. Die Konkretisierung diese abstrakten Modells, durch Festlegung des entsprechenden Zugriffskontrollmodelles ist der wesentliche Aspekt der Modellierung von Zugriffskontrolle im Entwicklungsprozess:
(Subjekt) Es muss festgelegt werden, wer die aktiven und passiven Parteien sind. Dies können sowohl menschliche Nutzer als auch autonome Systeme sein.
(Objekt) Die Granularität des zu schützenden Objektes muss festgelegt werden. Eine hohe Granularität ist z.B. beim Zugriff auf das gesamte System oder auf Dienste angebracht, während ein feingranulare Ansicht z.B. bei Methoden aufrufen oder bei Dateisystemzugriffen genutzt werden sollte.
(Recht) Die Spezifikation der Rechte muss festgelegt werden. Hierzu zählen die eigentlichen Beschränkungen im Zugriff als auch der jeweilige Kontext in dem der Zugriff stattfindet.

Referenzen

[SS04] R. S. Sandhu, Pierangela Samarati – Access Control: Principles and Practice, IEEE Communications Magazine, Vol. 32 (9), pp. 40-48, 1994
[Wi05] P. Windley, *Digital Identity*, O'Reilly, August 2005.



Eine direkte Interpretation der Zugriffskontrollmatrix führt zu einer benutzerbestimmten Zugriffskontrolle (engl. discretionary access control, DAC).

(1) Bei einem DAC-Ansatz, wird die Berechtigung für den Zugriff auf eine Ressource direkt mit der Ressource verknüpft und abgelegt. Üblicherweise wird eine Zugriffskontrollliste (engl. access control lists, ACL) für jede Ressource angelegt, in welchen für jeden Benutzer die jeweiligen Berechtigung abgelegt sind. Dieser Ansatz lässt sich sehr einfach umsetzen und findet vor allem in Betriebssystemen Anwendung. Ein anderer Ansatz assoziiert die Identitäten der Subjekte mit den Berechtigungen für die jeweiligen Ressourcen. Dies führt zu sogenannten capabilities lists (dt. Fähigkeitslisten)
(1.1) Eine Besonderheit des DAC-Ansatzes ist, dass den Ressourcen ein Besitzer zugeteilt wird, welcher den Vollzugriff auf die Ressource hat. Der Besitzer einer Ressource hat ebenso die Möglichkeit die Berechtigungen zur Manipulation der Ressource an andere Benutzer weiterzugeben.
(1.2) Eine Erweitung von DAC ist die Gruppierung von Subjekten zu Gruppen. In diesem Fall werden die Berechtigungen mit den Gruppen assoziiert und an die Mitglieder der Gruppe weitergereicht. Eine Gruppe ist dabei eine Sammlung von Subjekten, welche eine gemeinsames Ziel verfolgen bzw. ein gemeinsames Interesse haben.

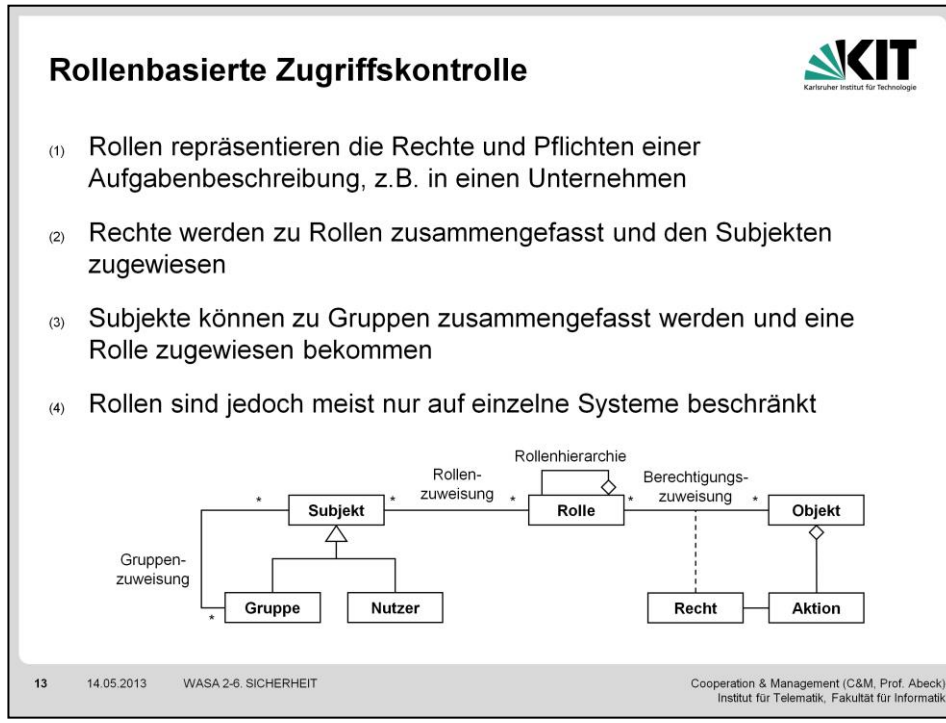
(2) Bei einer zentralen Verwaltung der Benutzer und Subjekte skaliert DAC nur schlecht, da die individuellen Subjekte verwaltet werden müssen. Zudem ist der Gruppenansatz häufig inflexibel und und verstärkt u.a. die Komplexität, da z.B. keine Gruppenhierarchien unterstützt werden. Für einen abgeschlossenen Organisationsbereich ist eine natürlichere Form der Berechtigungsverwaltung die Verwendung von Geschäftsfunktionen bzw. -Rollen [Wi05, p. 67-70].

Abkürzungen

ACL access control list
DAC discretionary access control

Referenzen

[SS04] R. S. Sandhu, Pierangela Samarati – Access Control: Principles and Practice, IEEE Communications Magazine, Vol. 32 (9), pp. 40-48, 1994
[Wi05] P. Windley, *Digital Identity*, O'Reilly, Sebastopol, CA, 2005.



(1) Rollenbasierte Zugriffskontrolle (engl. role-based access control, RBAC) ist ein beliebtes Autorisierungs- und Zugriffskontrollmodell. Es basiert auf der Beobachtung, dass Benutzer auf Ressourcen zugreifen, welche sie zur Erfüllung einer bestimmten Aufgabe in einem Unternehmen oder einer Organisation benötigen.

(2) Die Berechtigungszuweisung ist bei RBAC zweistufig. Zunächst werden die notwendigen Rechte zu Rollen zusammengefasst und anschließend werden den Subjekten bestimmt Rollen zugewiesen. Der Vorteil liegt in der einfachen Verwaltung von Rechten, da sie in aggregierten Form weniger häufig wechseln als die Personen, die die jeweiligen Aufgaben ausüben. Zusätzlich existieren Erweiterungen von RBAC, die es erlauben, Rollenhierarchien zu erzeugen, in welchen eine Rolle die Rechte von ihrer Elternrolle erbt. Dies ist eine weitere Vereinfachung der Rechteverwaltung [Wi05, p. 70; FS+01].

(3) Während Rollen aggregierte Rechte darstellen, fassen Gruppen Subjekte zusammen. Dies ist ein wesentlicher Unterschied [Me08, pp. 45-48].

(4) Trotz der Verbreitung von RBAC, hat dieses Modell auch seine Nachteile. Denn die ursprüngliche Intention, Rechte für Unternehmensaufgaben abzubilden wurde nicht erfolgreich umgesetzt. Eine solche Abbildung müsste die verschiedenen Systeme in einem Unternehmen mit berücksichtigen. Allerdings sind viele Rollen auf einzelne Systeme eingeschränkt und müssen aufwendig in verschiedenen Systemen gepflegt werden. [KM+09].

Abkürzungen

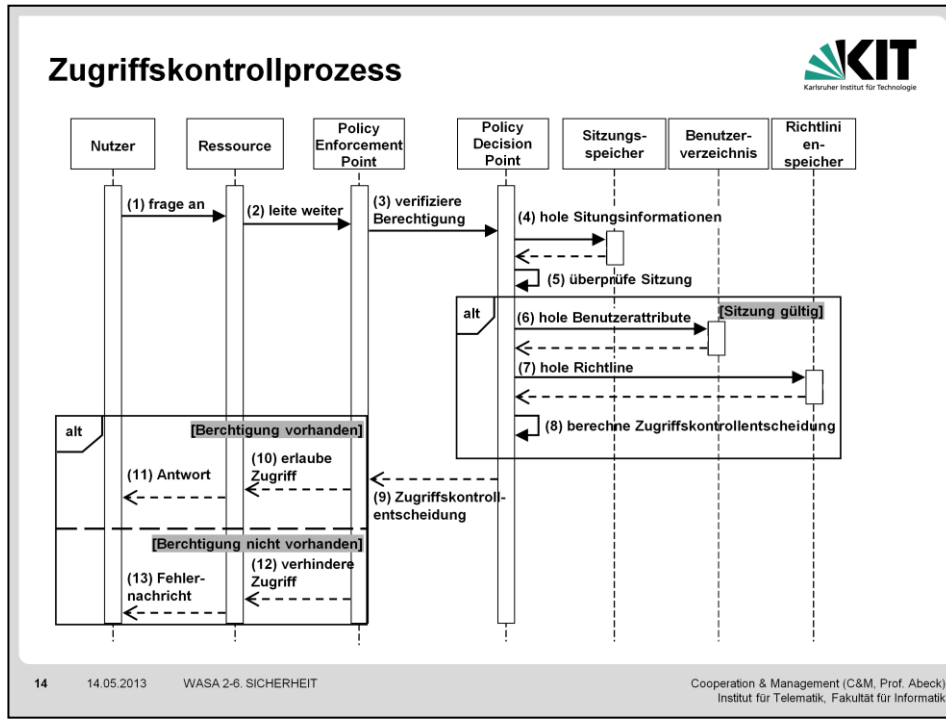
RBAC role-based access control; rollenbasierte Zugriffskontrolle

Referenzen

[FS+01] D. F. Ferraiolo, R. Sandhu, S. Gavrila, *Proposed NIST Standard for Role-Based Access Control*, ACM Transactions on Information and System Security (TISSEC), 2001, Vol. 4 (3), pp. 224-274

[KM+09] H. Klarl, et al., *Extending Role-based Access Control for Business Usage*, to appear in Proceedings of SECURWARE'09, Athens, Greece, June 2009.

[Me08] C. Mezler-Andelberg, *Identity Management - Eine Einführung*, dpunkt Verlag, 2008.



Die Abbildung zeig ein generischen Ablauf der Zugriffkontrolle bei der Anfrage eines Nutzers an eine Ressource als UML-Sequenzdiagramm:

(1) Der Benutzer fragt eine Ressource an. Neben den benötigten Parametern sendet er sein zuvor ausgestelltes Authentifizierungstoken mit, um seine Identität zu bestätigen.

(2) Eine Policy Enforcement Point-Komponente (PEP) kontrolliert den Zugriff auf die Ressource und setzt die gefällten Zugriffskontrollentscheidungen um. In dem dargestellten Fall, leitet die Ressource die Anfrage an den PEP weiter um den Zugriff zu verifizieren.

(3) Der PEP leitet die Anfrage weiter an eine Policy Decision Point-Komponente, welche ausschließlich für die Berechnung der Zugriffskontrollentscheidung verantwortlich ist. Der PEP übernimmt dabei die Kommunikation mit dem PDP und entkoppelt diese Logik von der fachlichen Funktionalität.

(4)-(5) Der PDP validiert zunächst die Authentifizierung und die Sitzung des Nutzers, indem er die entsprechenden Informationen aus dem Sitzungsspeicher abfragt. Bei einer ungültigen Sitzung oder fehlender Authentifizierung, sendet der PDP eine Fehlermeldung an den PEP, welcher den Authentifizierungsprozess erneut anstößt.
Es existieren unterschiedliche Muster für das Zusammenspiel von Ressource, PEP und PDP. Es ist z.B. auch möglich, das der PEP eine aktive Komponente ist, welche aktiv die Abfrage abfängt, ohne dass die Ressource selbst aktiv werden muss bzw. von der Existenz des PEP weiß.

(6)-(8) Bei einer aktiven Sitzung, werden im Anschluss vom PDP Benutzerdaten und Zugriffskontrollrichtlinien für die Ressource aus dem Benutzerverzeichnis bzw. dem Richtlinienspeicher geholt, um die Zugriffskontrollentscheidung zu fällen.

(9) Die Entscheidung wird vom PDP an den PEP gesendet, welcher diese entsprechend umsetzt.

(10)-(13) Wird der Zugriff erlaubt, so wird die Anfrage an die Ressource weitergeleitet. Im negativen Fall wird dem Nutzer eine Fehlernachricht angezeigt bzw. der Zugriff verweigert.

Abkürzungen

PDP	policy decision point
PEP	policy enforcement point

LZ KOMPONENTEN- UND ANWENDUNGSSICHERHEIT – ÜA Authentifizierung und Zugriffskontrolle

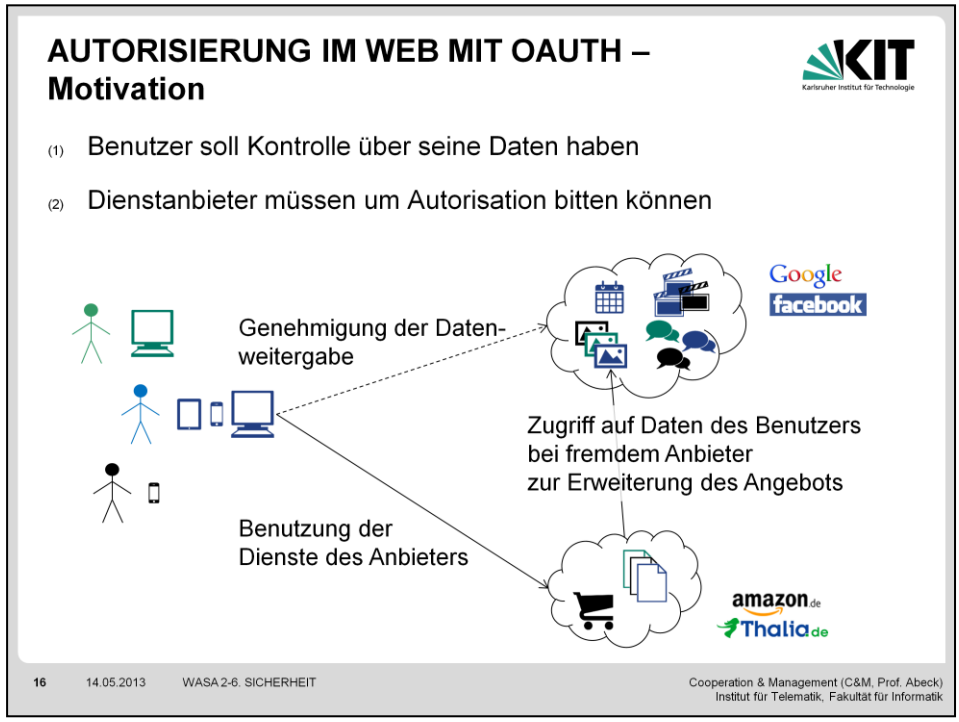
(1) Erläutern Sie den Unterschied zwischen Identifizierung und Authentifizierung.

(2) Wozu dienen zeitbeschränkte Authentifikationstokens?

(3) Erläutern Sie den Unterschied zwischen Autorisierung und Zugriffskontrolle

(4) Eine Grundlage für Zugriffskontrolle ist
   (1) Identifikation
   (2) Authentifikation
   (3) Autorisierung

(5) Erläutern Sie kurz die konzeptionelle Idee hinter rollenbasierter Zugriffskontrolle. Was sind die Probleme bei der Umsetzung dieses Konzeptes?



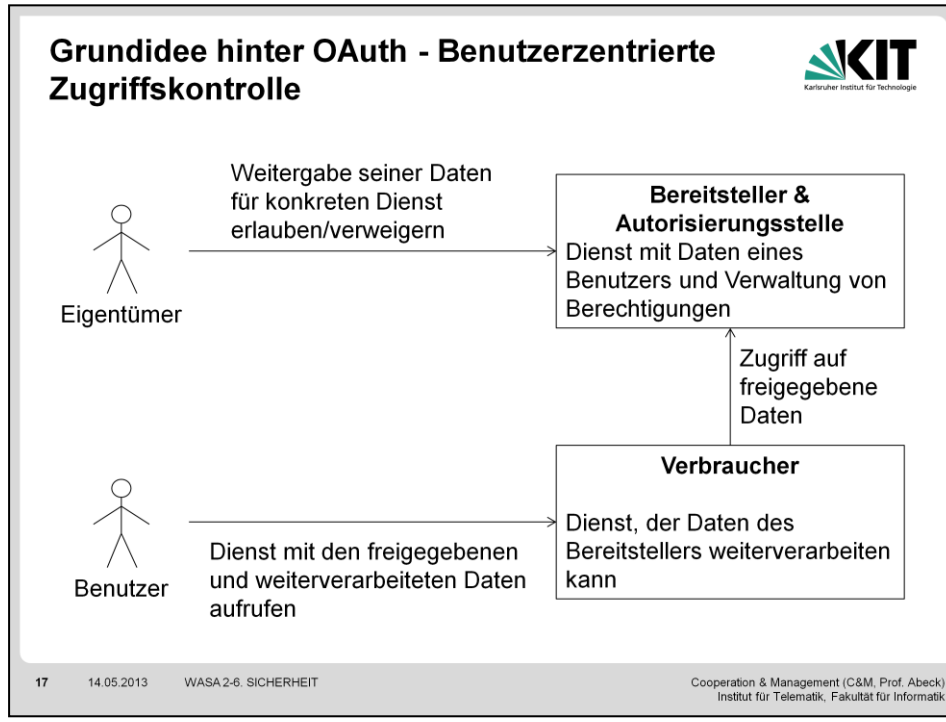
OAuth 2.0 ist ein Framework, dass von der Internet Engineering Task Force (IETF) in [IETF-TOA] beschrieben und zur Autorisierung im Web verwendet wird.

(1) Ein zentrales Ziel bei der Verarbeitung von Daten im Internet muss es sein, dass der Anwender oder der Eigentümer der Daten die Entscheidungshoheit hat. Für ihn muss transparent sein, wer Zugriff auf seine Daten hat und er muss darüber selbst entscheiden können.

(2) Will ein Dienst die Daten eines Anwenders im Zusammenhang mit einem anderen Dienst nutzen, muss einheitlich festgelegt sein, wie dieser die Autorisation erhält.

Die Abbildung zeigt ein typisches Autorisierungsszenario im Web an einem fiktiven Beispiel. Rechts oben ist ein Dienstanbieter mit einer großen Menge Benutzerdaten, wie Google oder Facebook, dargestellt. Als konkrete Daten sind Bilder, Videos, Chatnachrichten und Kalender zu sehen. Zum Zugriff auf die Daten bietet der Anbieter einen passenden Bilder-, Video, Chat und Kalenderdienst an. Die Daten haben jeweils einen konkreten Eigentümer - hier durch farbliche Unterschiede angedeutet. Die Daten des Bilderdienstes kann ein Einkaufsdienstanbieter, rechts unten, verarbeiten und sein Angebot so verbessern. Als Leistung bietet er an, Objekte auf freigegebenen Bildern zu erkennen und das dazugehörige Produkt zu identifizieren. Der Bilderdienst darf die Daten eines Benutzers aber nicht ohne seine Genehmigung an andere Anbieter weitergeben. Der Benutzer braucht somit einen Weg, die Datenweitergabe zu genehmigen, um das erweiterte Angebot beim Einkaufsdienst nutzen zu können. Andererseits muss der Einkaufsdienst eine Möglichkeit haben, den Benutzer um Freigabe von benötigten Daten zu bitten. OAuth bietet eine Lösung für diese Probleme an. Zudem beachtet OAuth eine Reihe von Anwendungsfällen, die sich beispielsweise durch die angedeutete Heterogenität der Endgeräte ergibt.

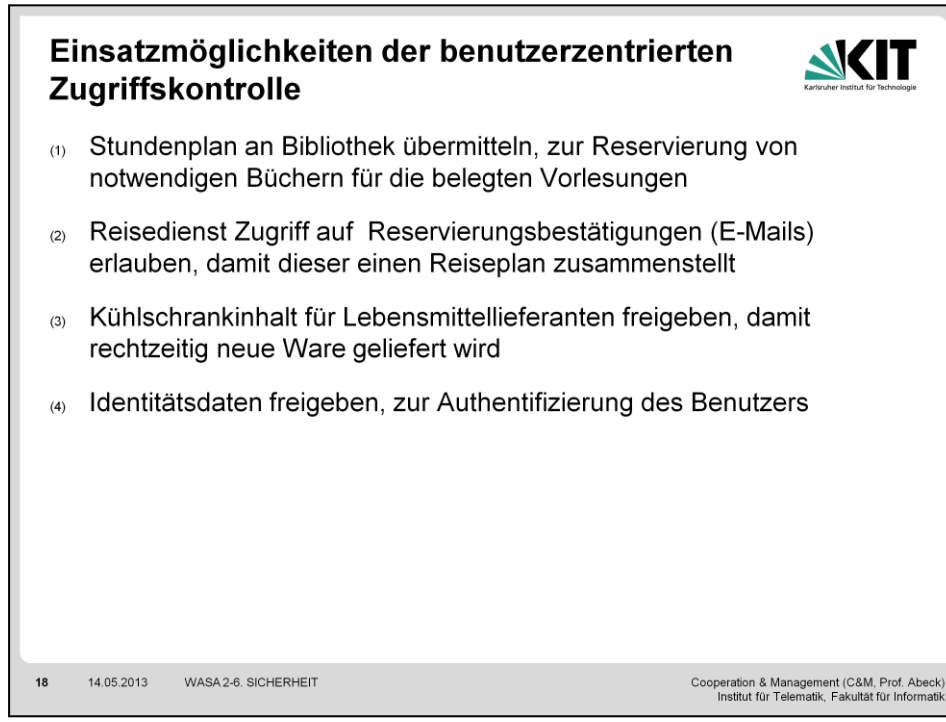
[IETF-TOA] Internet Engineering Task Force: The OAuth 2.0 Authorization Framework, Request for Comments RFC 6749, Oktober 2012.



Die Idee hinter der benutzerzentrierten Zugriffskontrolle ist, dass der Eigentümer von Daten kontrollieren kann, wie diese verwendet werden.
Dabei wird davon ausgegangen, dass es einen Bereitsteller der Daten gibt, der eng mit einer Autorisierungsstelle verbunden ist. Zudem gibt es einen Verbraucher, der die Daten verarbeiten kann. Damit der Verbraucher auf die Daten zugreifen kann, muss er zuvor vom Eigentümer autorisiert werden. Hierzu muss der Eigentümer der Autorisierungsstelle mitteilen, dass der Verbraucher auf seine Daten zugreifen darf. Bei jedem Zugriff des Verbrauchers auf die Daten beim Bereitsteller wird überprüft, ob die Autorisierung des Eigentümers besteht. Der Eigentümer kann jederzeit seine Autorisierung für einen Dienst widerrufen und auf diese Weise seine Daten schützen. Die weiterverarbeiteten Daten werden von einem Benutzer abgerufen. In [MM+10] wird die Verwaltung der Zugriffsrechte durch den Benutzer und die Struktur mit Verbraucher, Bereitsteller und Autorisierungsstelle beschrieben und mit anderen Zugriffskontrollmodellen in Relation gebracht.

(Eigentümer) Die Daten gehören ihm und er soll die Kontrolle über deren Verwendung haben.
(Benutzer) Der Benutzer greift auf den (Daten-) Verbraucher zu und kann dessen Dienste mit den für ihn freigegebenen weiterverarbeiteten Daten nutzen. In den bisherigen Anwendungen von OAuth ist der Benutzer auch der Eigentümer der Daten. Prinzipiell kann der Benutzer aber auch eine weitere Person sein. Beim Beispiel der Produkterkennung auf Bildern, könnten sich Freunde gegenseitig interessante Produktbilder freigeben.
(Bereitsteller) Bei dem Bereitsteller sind die Daten gespeichert. Der Zugriff auf die Daten darf nur gewährt werden, wenn der Benutzer den Zugriff autorisiert hat.
(Autorisierungsstelle) Sie nimmt die zentrale Rolle bei der Zugriffskontrolle ein. Unter anderem ist speichert sie, welche Daten ein Benutzer welchen Verbrauchern freigegeben hat. Im Fall von OAuth muss zudem das Verfallsdatum der Autorisierung gespeichert werden. Gegenüber dem Bereitsteller muss ein ausgeprägtes Vertrauensverhältnis bestehen, da der Bereitsteller mittels der Autorisierungsstelle die Validität von Autorisationen überprüft.
(Verbraucher) Beim Verbraucher wird ein Mehrwert durch die Verarbeitung der Daten des Eigentümers erzeugt. Beispielsweise könnte der Verbraucher die Sortierung von Daten oder das Ausfiltern von uninteressanten Daten anbieten. Um den Dienst an Daten des Eigentümers ausführen zu können, muss er allerdings erst die Autorisation des Eigentümers einholen.

[MM+10] User-Managed Access to Web Resources.



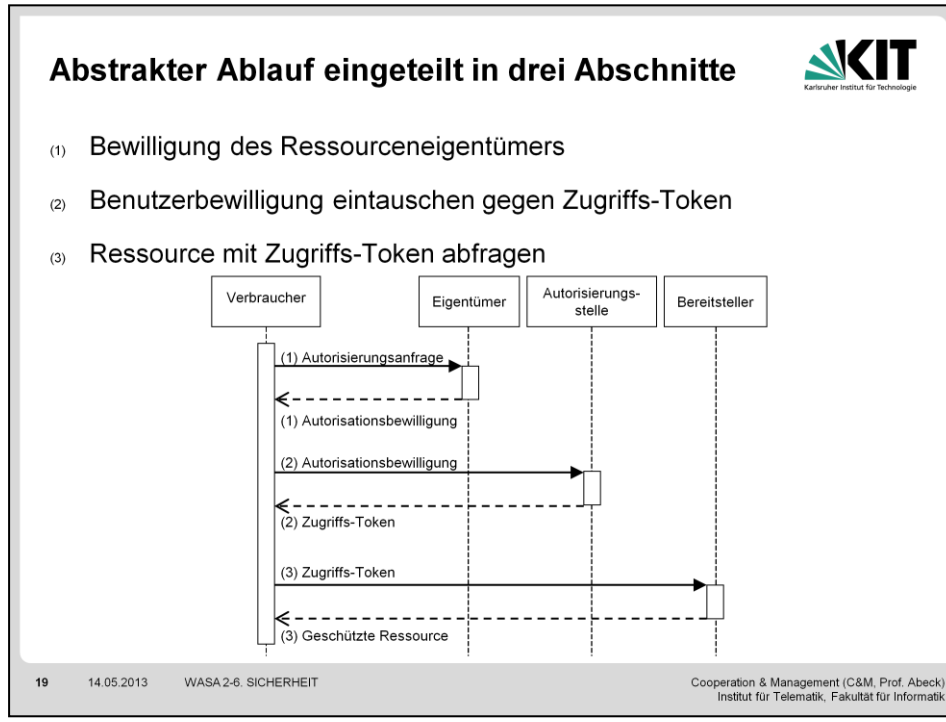
In den folgenden Beispielen ist der Dateneigentümer gleichzeitig Benutzer des Verbrauchers.

(1) Eine interessante Verwendung im universitären Bereich wäre die Reservierung von Büchern, passend zum eigenen Stundenplan. Auf der Seite der Bibliothek könnte eine Funktion, zum Auslesen des Stundenplans aus dem Studierendenportal vorhanden sein. Autorisiert der Student die Bibliothek, den Stundenplan auszulesen, so könnten direkt die Bücher passend zur Literaturliste der Vorlesungen für den Studenten ausgeliehen werden. Der Stundenplan ist die Ressource und diese ist beim Studierendenportal hinterlegt, somit ist das Studierendenportal der Ressourcenbereitsteller. Die Bibliothek ruft den Stundenplan ab und tätigt die Reservierung der passenden Bücher und nimmt somit die Rolle des Ressourcenverbrauchers ein.

(2) Einen Mehrwert könnten auch Reisedienste erreichen. TripIt als Planungsplattform bietet beispielsweise bereits die Möglichkeit, dass Reservierungsbestätigungen (E-Mails) von Hotels direkt ausgewertet und in die Reiseplanung aufgenommen werden. Auswerten meint dabei, dass An- und Abreisedatum sowie Hotelname und Standort im Text der E-Mail gefunden und verstanden werden. Derzeit muss der Benutzer die E-Mails aber an TripIt weiterleiten. Zukünftig könnte TripIt direkt beim E-Mail-Dienst die E-Mails abrufen und der Umweg über eine Weiterleitung würde entfallen.

(3) Im Bereich des Internets der Dinge entwickeln sich ebenfalls viele neue Einsatzmöglichkeiten einer benutzerzentrierten Zugriffskontrolle. Der Zugriff auf alle Objekte in einem Haushalt durch Web-Dienste könnte auf diese Art geschützt werden. Ein konkretes Beispiel ist der Kühlschrankinhalt, der an einen Lebensmittellieferanten freigegeben wird. Der Lieferant könnte in regelmäßigen Abständen den aktuellen Inhalt abfragen und gegebenenfalls neue Ware liefern. Die schützenswerte Ressource ist in diesem Fall der Inhalt des Kühlschranks. Bereitgestellt wird der Inhalt vom Kühlschrank selbst. Der Lieferant tritt in dem Beispiel als Verbraucher auf.

(4) Die derzeit am häufigsten genutzte Anwendung, ist die Freigabe der Identitätsdaten an andere Dienste zur Authentifizierung. Ein Beispiel hierfür ist die Anmeldung mit einem Facebook-Account beim Reiseplanungsdienst TripIt. Des Weiteren bieten viele private Seitenbetreiber die Möglichkeit an, sich bei diesen mit beispielsweise einem Google-, Facebook- oder Twitter-Account anzumelden. Auf diese Weise lagern die Anbieter die Speicherung der Identitätsdaten aus. Die Ressourcen sind in diesem Beispiel die Identitätsdaten des Benutzers. Die Webseite, bei der der Benutzer sich anmelden will, nimmt die Rolle des Ressourcenverbrauchers ein, da dieser die Identitätsdaten abruft. Der Dienst, bei dem die eigenen Identitätsdaten hinterlegt sind, ist der Ressourcenbereitsteller.



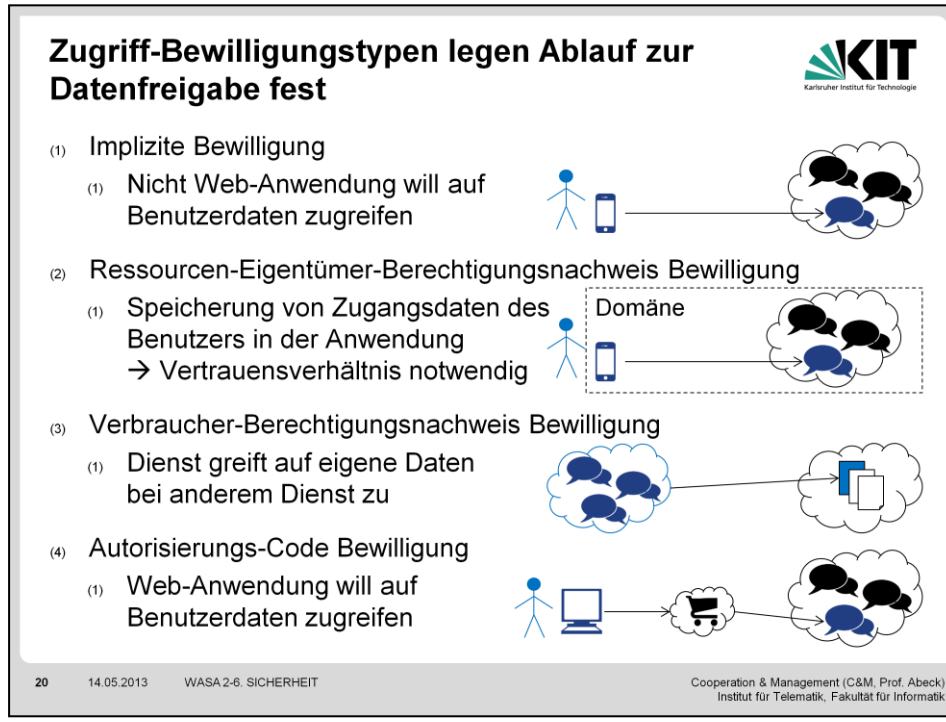
In [IETF-TOA:1.1] wird der abstrakte Ablauf des Protokolls wie dargestellt beschrieben. Zudem werden eine Reihe von Sicherheitshinweisen, die bei der Implementierung beachtet werden sollten, gegeben. Grob kann der Ablauf in die folgenden drei Abschnitte eingeteilt werden.

(1) Zuerst erhält der Verbraucher eine Zugriffsautorisierung des Eigentümers für eine Ressource. Der Verbraucher fragt hierfür beim Eigentümer an und erhält im positiven Fall eine Autorisationsbewilligung. Diese Bewilligung vergibt der Eigentümer für einen konkreten Verbraucher, an den diese gebunden ist.

(2) Der Verbraucher muss die Bewilligung gegen einen Zugriffs-Token bei der Autorisierungsstelle eintauschen. Dieser Schritt ist notwendig, da der Eigentümer die Bewilligung für einen bestimmten Verbraucher gewährt hat. Deshalb muss sich der Verbraucher in diesem Schritt authentifizieren. Die Autorisierungsstelle kann so prüfen, ob die Bewilligung für diesen Verbraucher ausgestellt wurde. Zudem muss die Autorisierungsstelle die Echtheit prüfen. Dies setzt voraus, dass die Autorisierungsstelle mit dem Eigentümer in Kontakt steht. Auf diesen Aspekt wird in den konkreten Bewilligungstypen eingegangen.

(3) Im letzten Schritt kann der Verbraucher durch seinen Zugriffs-Token auf die Ressource beim Bereitsteller zugreifen. Hierzu muss der Bereitsteller die Echtheit des Tokens überprüfen können und beispielsweise mit der Autorisierungsstelle kommunizieren. Denkbar ist auch, dass es einen eigenen Dienst für Token gibt, auf den die Autorisierungsstelle und der Bereitsteller zugreifen. Wie genau die Echtheit überprüft wird, legt OAuth nicht fest.

[IETF-TOA] Internet Engineering Task Force: The OAuth 2.0 Authorization Framework, Request for Comments RFC 6749, Oktober 2012.



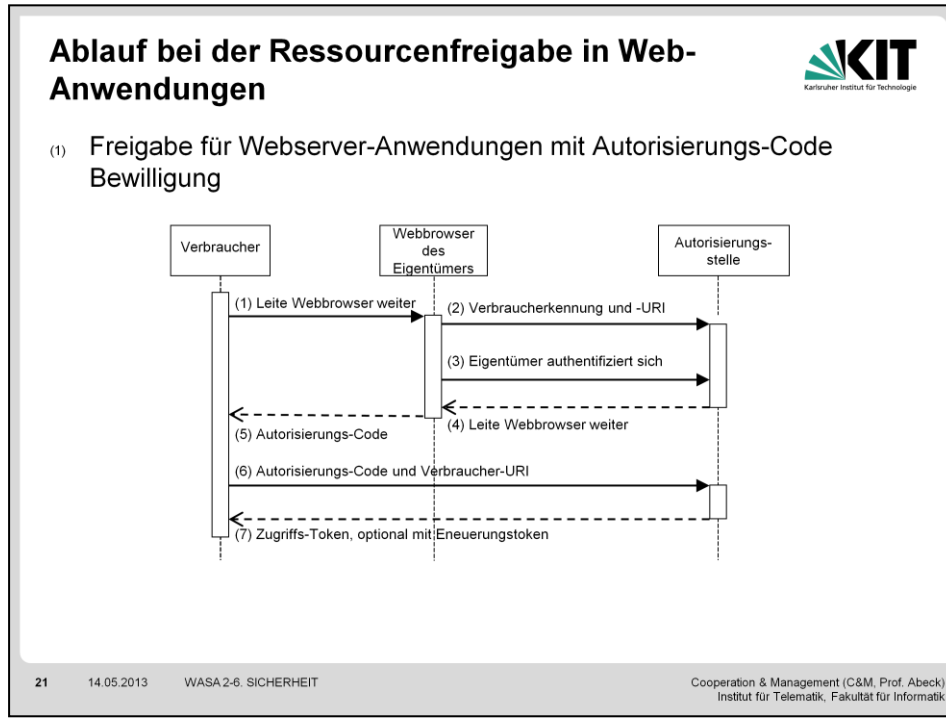
Diese Folie befasst sich mit den ersten beiden Schritten des abstrakten Ablaufs. Die Bewilligungstypen wurden aus vier Anwendungsfällen entwickelt, welche im folgenden eingeführt werden.
(1) In diesem Anwendungsfall hat der Anwender eine native Applikation auf seinem mobilen Endgerät installiert und diese möchte auf Daten des Anwenders zugreifen, die dieser bei einem externen Dienst hinterlegt hat. Neben einer nativen Applikation könnte der Anwender auch eine Flash-oder Javascript-Anwendung mit seinem Browser geöffnet haben und diese erbittet seine Autorisierung. Der zweite Fall betrifft eine sogenannte Thick-Client-Architektur. Für diesen Anwendungsfall sieht die OAuth-Spezifikation den Einsatz der impliziten Bewilligung vor [IETF-TOA:4.2].

(2) In diesem Anwendungsfall wird angenommen, dass ein starkes Vertrauensverhältnis zwischen der Anwendung und dem Ressourcenbereitsteller besteht und es keine Sicherheitslücke darstellt, wenn die Anwendung die Zugangsdaten des Benutzers kennt. Der Benutzer gibt die Zugangsberechtigung zu seinen Daten in der Anwendung an und gewährt so den Zugriff. Im Folgenden kann die Anwendung mit der Zugangsberechtigung auf die Daten zugreifen. Angedacht für diesen Einsatz ist der Ressourcen-Eigentümer-Berechtigungsnachweis Bewilligung. Neben Desktop-Anwendungen kann dieser Bewilligungstyp auch bei vertrauenswürdigen mobilen oder browserbasierten Anwendungen eingesetzt werden. Allerdings sollte stets die Geheimhaltung des Berechtigungsnachweises im Mittelpunkt stehen[IETF-TOA:4.3].

(3) Der letzte Anwendungsfall betrifft den Zugang der Anwendung auf die eigenen Daten. Beispielsweise könnte es ein Ressourcenbereitsteller ermöglichen, dass die Client-Anwendungen ein Anwendungslogo hinterlegen. Mit der Client-Berechtigungsnachweis Bewilligung kann der Client Zugriffs-Token zum Ändern des Bildes abrufen. OAuth beschreibt nicht, wie ein Berechtigungsnachweis auszusehen hat. Dieser kann beispielsweise als Passwort oder Zertifikat vorliegen und muss zwischen den Akteuren ausgehandelt werden [IETF-TOA:4.4].

(4) Der Anwender hat eine Anwendung mit seinem Browser geöffnet. Die Anwendung möchte nun Daten des Benutzers, die dieser bei einem anderen Dienst hinterlegt hat abrufen und bittet ihn deshalb um eine Zugriffs-Bewilligung. Die Anwendung agiert in diesem Szenario auf dem Webserver und nicht im Browser des Clients. Der Browser dient somit lediglich der Darstellung, eine sogenannte Thin-Client-Architektur. Für dieses Beispiel ist der Einsatz der Autorisierungs-Code Bewilligung angedacht [IETF-TOA:4.1].

[IETF-TOA] Internet Engineering Task Force: The OAuth 2.0 Authorization Framework, Request for Comments RFC 6749, Oktober 2012.



Das Beispiel der E-Mail Freigabe für die Reiseplanungsanwendung kann auf den dargestellten generalisierten Ablauf abgebildet werden.
Dargestellt ist der Ablauf zur Freigabe einer Ressource für einen Thin Client, somit läuft die Anwendung auf dem Client und der Webbrowser wird lediglich zur Darstellung der Inhalte genutzt.
(1) Konkret ist der Ablauf, den die Autorisierungs-Code Bewilligung vorsieht, dargestellt. Die Interaktion des Benutzers mit dem Browser ist wegen der Übersichtlichkeit nicht explizit dargestellt und steckt hinter dem Ablaufpunkt (3) "Eigentümer authentifiziert sich". Das vollständige Diagramm kann in in [IETF-TOA:4.1] gefunden werden. Wichtige Sicherheitshinweise zur Implementierung werden dort ebenfalls gegeben.

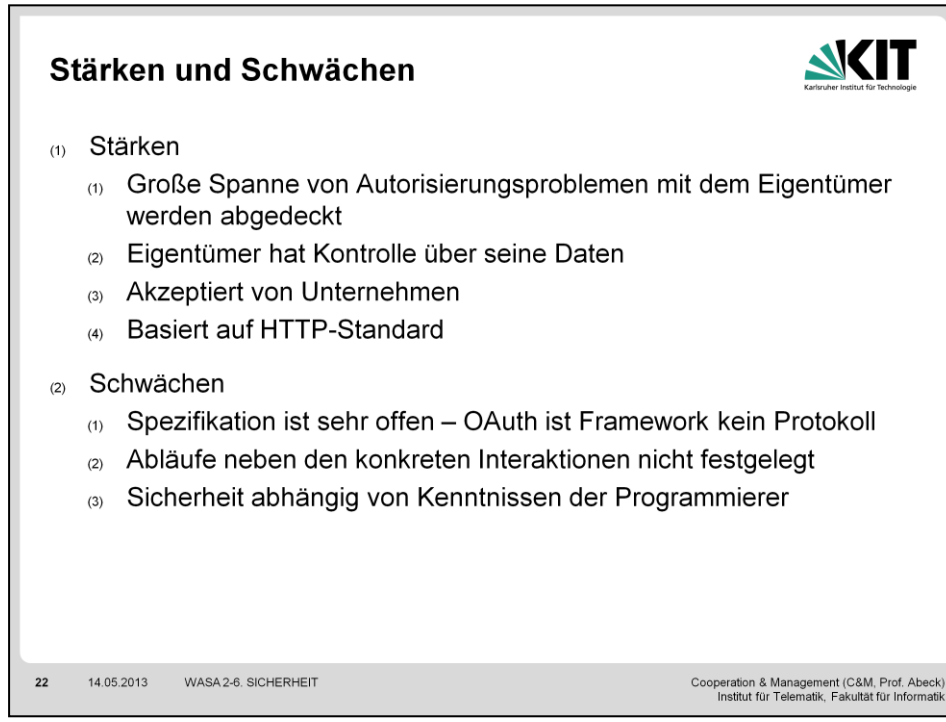
Ablauf
(1,2) Der Bewilligungsprozess beginnt mit der Weiterleitung des Benutzers an die jeweilige Autorisierungsstelle. Bei der Weiterleitung wird eine Kennung des Verbrauchers und die URI, an die nach der Interaktion des Anwenders mit der Autorisierungsstelle zurückgeleitet werden soll, mit übertragen. Die Weiterleitungs-URI kann bereits im Vorhinein festgelegt sein, damit der Autorisierungs-Code immer den richtigen Verbraucher erreicht.

(3) Sofern der Anwender noch nicht authentifiziert ist, muss er sich nun bei der Autorisierungsstelle anmelden. Wie genau die Anmeldung durchgeführt wird, ist nicht Teil der OAuth-Spezifikation. Das heißt es können sowohl Benutzername und Passwort, ein Kartenlesegerät oder sonstige Authentifizierungsverfahren eingesetzt werden.

(4,5) Nach der Einwilligung des Benutzers bei der Autorisierungsstelle, wird dieser mit einem Autorisierungs-Code an die festgelegte Weiterleitungs-URI des Verbrauchers zurückgeschickt.

(6,7) Abschließend kann der Verbraucher den Autorisierungs-Code gegen einen Zugriffs-Token eintauschen. Dies geschieht durch eine Anfrage an die Autorisierungsstelle mit dem Autorisierungs-Code und wiederum einer Weiterleitungs-URI. Als Antwort erhält der Verbraucher den Zugriffs-Token und optional einen Refresh-Token, um abgelaufene Zugriffs-Token zu erneuern.

[IETF-TOA] Internet Engineering Task Force: The OAuth 2.0 Authorization Framework, Request for Comments RFC 6749, Oktober 2012.



(1.1) Bei der Entwicklung von OAuth wurden verschiedene Anwendungsfälle betrachtet, die bei der Interaktion eines Benutzers mit Web-Anwendungen auftreten können. Aus diesem Grund wurde es geschafft, eine Vielzahl von Autorisierungsproblemen in diesem Umfeld abzudecken.
(1.2) Durch die Benutzerzentrierung wurde erreicht, dass der Eigentümer selbst über die Verarbeitung von seinen Daten entscheiden kann. Bevor ein Verbraucher auf seine Daten zugreifen kann, muss der Eigentümer erst seine Einverständnis geben.
(1.3) OAuth ist mittlerweile weit verbreitet und wird von verschiedenen Unternehmen eingesetzt, da diese aktiv an der Entwicklung mitgewirkt haben. Beispiele für diese Verbreitung ist der Zugriff per OAuth auf Daten von Google, Facebook, Twitter oder Microsoft-Anwendungen. Dies liegt auch daran, dass der Standard relativ offen gehalten ist. Beispielsweise ist der Zusammenhang zwischen Bereitsteller und Autorisierungsstelle offen gehalten, wodurch diese sowohl durch ein einziges System oder mehrere stark verteilte Systeme in der Cloud repräsentiert werden können.
(1.4) Als Basistechnologie zur Übertragung setzt OAuth auf dem HTTP-Standard auf und passt somit in die Web-Landschaft.
(2.1) Die Beteiligung vieler Unternehmen bei der Entwicklung hat auch seine Nachteile. OAuth wurde sehr offen spezifiziert und kann somit in sehr vielen Gebieten eingesetzt werden. Allerdings führt dies auch dazu, dass einige Aspekte nicht konkret beschrieben wurden. Beispielsweise wird nur vorausgesetzt, dass sich der Benutzer authentifizieren können soll, aber nicht wie dies geschieht. Des Weiteren ist unklar, wie der Bereitsteller und die Autorisierungsstelle miteinander in Verbindung stehen und wie diese miteinander kommunizieren (müssen). Ein schnelles Implementieren des Bereitstellers und der Autorisierungsstelle ist somit nicht möglich. Diese Offenheit wird selbst von Mitschreibern der Spezifikation als großes Problem angesehen, wie folgender Blogbeitrag zeigt [Ham12].
(2.2) Neben den definierten Interaktionen bleiben einige Fragen ungeklärt. Beispielsweise soll der Benutzer nach der Authentifizierung beim Bereitsteller angemeldet bleiben? Dies wäre für den Benutzer nicht ersichtlich und stellt beispielsweise bei öffentlichen Rechnern in Internetcafés ein Problem dar. Der nächste Benutzer des gleichen PCs hätte Zugang zum Benutzerkonto seines Vorgängers beim Bereitsteller.
(2.3) In der Spezifikation von OAuth wird auf Grund der offenen Spezifikation ein komplettes Kapitel den Schwierigkeiten bei der Programmierung gewidmet. Wie sollte implementiert werden, damit keine Sicherheitslücken entstehen. Dies betrifft beispielsweise das zuvor genannte Problem der dauerhaften Anmeldung.

[Ham12] OAuth 2.0 and the Road to Hell, http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/

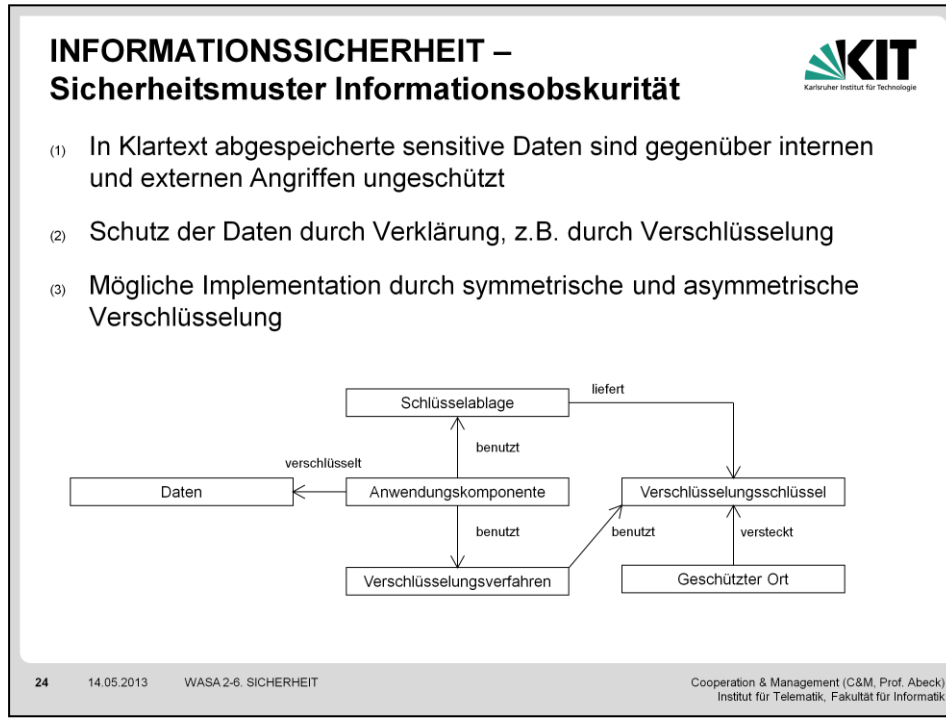
LZ AUTORISIERUNG IM WEB – ÜA
Benutzerzentrierte Zugriffskontrolle und OAuth

(1) Welches sind die beteiligten Rollen in OAuth und wie spielen diese zusammen?

(2) Wie sieht ein abstrakter Ablauf in OAuth aus und welche Informationen werden zwischen den Beteiligten ausgetauscht?

(3) Nennen Sie zwei mögliche Einsatzszenarien von OAuth, die nicht in der Vorlesung genannt wurden. Wer nimmt welche Rolle in den Szenarien ein?

(4) Welche Möglichkeiten gibt es als Verbraucher, Zugriff auf eine Ressource zu bekommen?



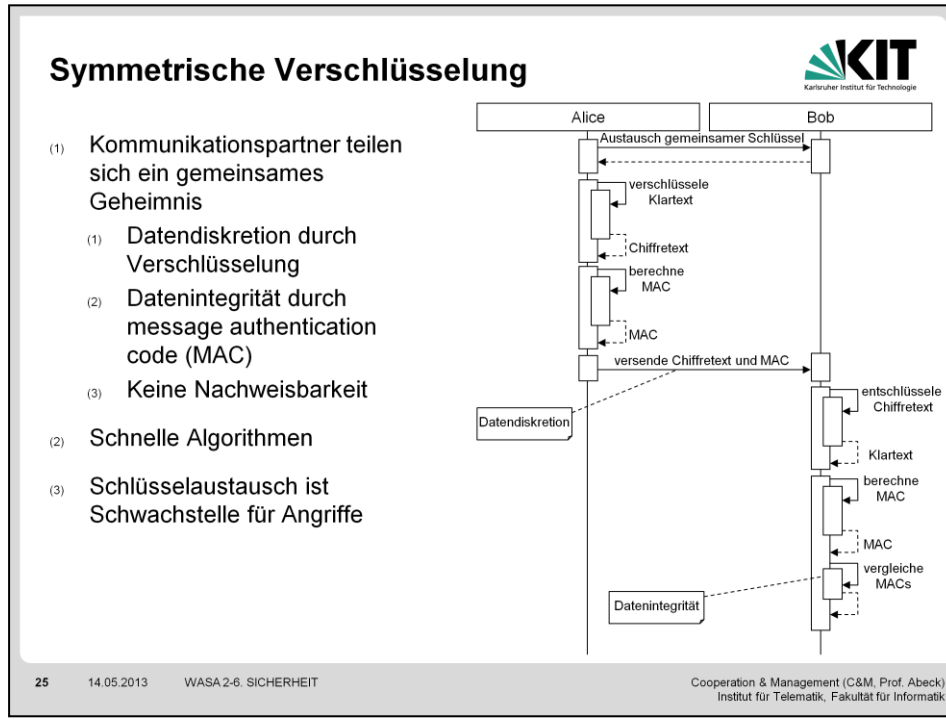
(1) Der Zugriff zu geschäftskritischen oder anderweitig sensiblen Daten ist in der Regel von außen durch zahlreiche Sicherheitsmechanismen eingeschränkt. Hierdurch sollen unautorisierte Zugriffe von außen verhindert werden. Erfolgt ein unautorisierter Zugriff von innerhalb einer Organization oder überwinden mögliche Angreifer die vorgeschalteten Sicherheitsmechanismen, sind die Daten meist ungeschützt gespeichert.

(2) Ein Großteil der Informationen in einem Anwendungssystem ist nicht von sensitiver Natur und muss daher nicht verschlüsselt werden. Je nach Schutzbedarf der Informationen sollten sie in bestimmten Fällen in verschlüsselter Form, d.h. nicht in Klartext, abgespeichert werden. Kryptografische Verfahren arbeiten mit sogenannten Schlüsseln. Sind diese in einer Schlüsselablage jedoch selbst ungeschützt abgelegt, so ist auch eine Verschlüsselung der Daten ineffektiv. Auch sie Schlüsselablage sollte ebenfalls an einem gut geschützten Ort aufbewahrt werden. Der Schutz von Daten in einem persistentem Speicher durch verschiedene Formen von kryptographischen Verfahren ist der letzte Schutzwall vor unberechtigten Zugriff.
Eine weitere Methoden zur Verklärung, welche aber aufgrund ihrer Komplexität und Benutzerunfreundlichkeit eher selten eingesetzt wird, ist etwa die Steganographie, die versucht, die Existenz von Informationen zu verdecken, statt ihren Inhalt zu verbergen [SF+05: 426ff].
Zu beachten ist allerdings, dass kryptografische Verschlüsselungsverfahren sehr aufwendig und langsam sind. Dieser erhöhte Resourcen- und Performanceverbrauch ist, wenn nicht unbedingt nötig, zu vermeiden. [SF+05: 426ff].

(3) Bei der Verschlüsselung wird zwischen symmetrischen und asymmetrischen Verschlüsselungsverfahren unterschieden, welche im Folgenden vorgestellt werden.

Referenzen:

[SF+05] M. Schumacher, E. B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns. Chichester, England: John Wiley & Sons Ltd, 2005, p. 565.



(1) Symmetrische Verschlüsselung weist die Eigenschaft auf, dass alle am Kommunikationsaustausch beteiligten Partner einen identischen Schlüssel zur Ver- und Entschlüsselung verwendeten.
(1.1)-(1.3) Durch die Verschlüsselung sind sowohl die Diskretions- als auch die Integritätseigenschaft der Daten erfüllt. Die Verschlüsselung verhindert die Einsicht der Daten durch Personen, welche den Schlüssel nicht besitzen. Dies sind üblicherweise unberechtigte Personen. Mittels des Schlüssels lässt sich ebenfalls ein message authentication code (MAC) erzeugen, welcher die Integrität der Daten nachweist. Die MACs vom einem Originaltext und der veränderten Kopie würden sich mit dem gleichen Schlüssel unterscheiden.

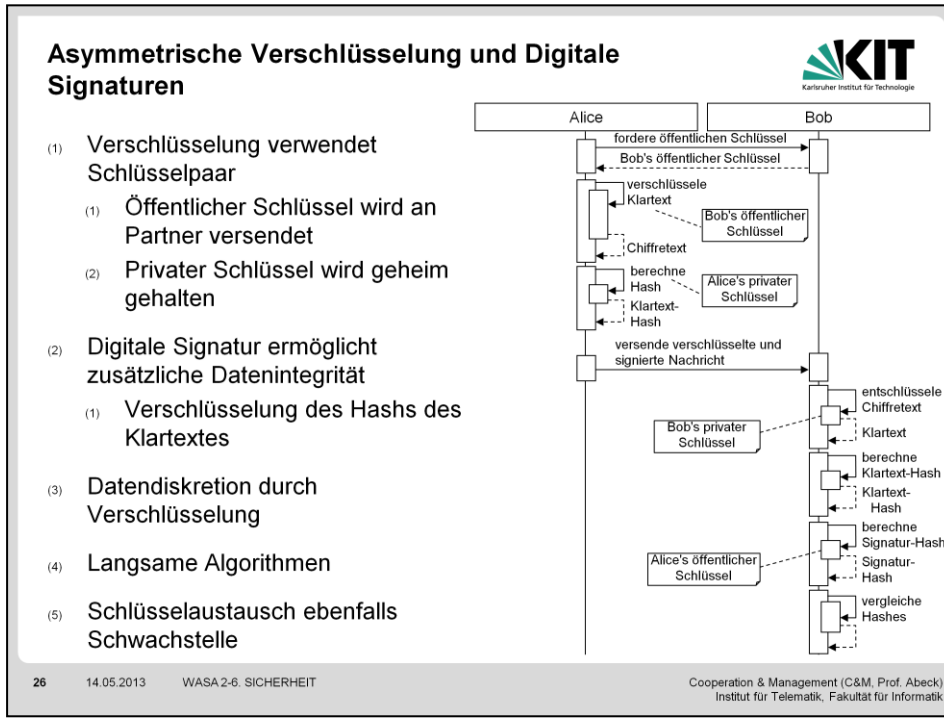
(2) Der Grund für den weiten Einsatz von symmetrischen Verschlüsselungsverfahren liegt in der Performanz der verwendeten Algorithmen.

(3) Die Verschlüsselung ist hinfällig, sobald eine unberechtigte Personen in den Besitz des gemeinsamen Schlüssels kommt. Hierbei ist neben der geschützten Ablage des Schlüssels der Schlüsselaustausch die größte Gefahrenquelle, da dieser oft über ein ungesichertes Medium, z.B. Internet, geschieht. Entsprechend sind die Verfahren zum Schlüsselaustausch oft sehr aufwendig und verwenden z.B. asymmetrische Verschlüsselung, die in diesem Bereich etwas sicherer ist.

Die Verschlüsselung von Daten mittels symmetrischer Verschlüsselung gestaltet sich wie folgt:
(Alice, Bob) Dies sind zwei generische Partner, die Daten sicher austauschen wollen. Dies können entfernte Kommunikationspartner sein aber auch ein lokaler Benutzer in Interaktion mit einem Massenspeicher.
(Austausch) Zunächst muss das gemeinsame Geheimnis, bzw. der Schlüssel, zwischen den beiden Partnern ausgetauscht werden, so dass beide über die gleich Kopie verfügen.
(Verschlüssele Klartext, Chiffretext) Alice sendet Daten an Bob und muss diese zunächst mittels des Schlüssels und eines Algorithmuses verschlüsseln. Das Ergebnis wird Chiffretext genannt.
(MAC) Vom Klartext wird ein MAC berechnet, um spätere Modifikationen zu erkennen.
(Versende ...) Beide Artefakte werden an Bob gesendet.
(Entschlüsselung) Bob entschlüsselt den Chiffretext und erhält einen Klartext.
(vergleich MAC) Von dem erhaltenen Klartext berechnet er auf dieselbe Art und Weise wie Alice einen MAC. Er vergleicht diesen mit dem von Alice gesendeten MAC. Sind die beiden identisch, kann sich Bob sicher sein, dass der Klartext nicht von anderen modifziert wurde.

Referenzen:

[Wi05] P. J. Windley, Digital Identity – Unmasking Identity Management Architecture (IMA), Beijing, Köln: O'Reilly, 2005.



(1) Im Gegensatz zur symmetrischen Verschlüsselung, wird der Schlüssel in einen öffentlichen und einen privaten Teil aufgespalten. Der öffentliche Schlüssel kann ohne Bedenken an Partner versendet werden, da er ohne den privaten Schlüssel nicht zur Entschlüsselung genutzt werden kann. Der private Schlüssel sollte dagegen an einem geschützten Ort geheimgehalten werden.

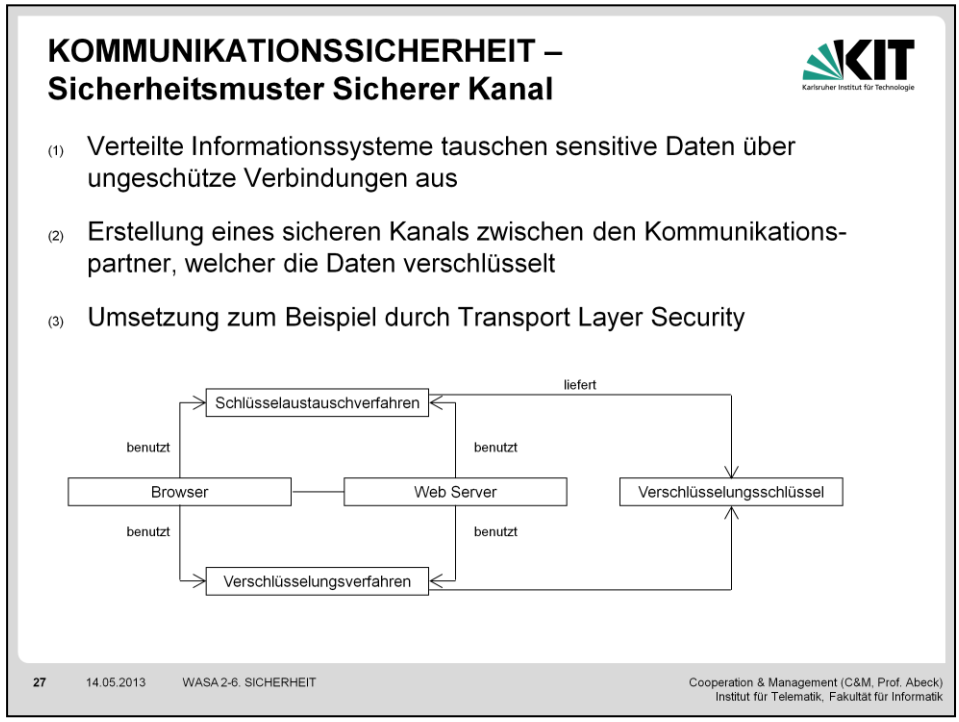
(2) Ähnlich wie der MAC bei symmetrischer Verschlüsselung kann bei asymmetrischer Verschlüsselung durch eine digitale Signatur die Integritätseigenschaft der Daten garantiert werden. Die Signatur wird durch die zusätzliche Verschlüsselung des Hash des Klartextes erhalten.

(4)-(5) Die Ver- und Entschlüsselung wird durch die Verwendung von zwei Schlüsselteilen erschwierigt, da mit großen Primärzahlen gearbeitet werden muss. Dadurch sind asymmetrische Verfahren häufig langsamer als symmetrische. Auch der Schlüsselaustausch ist nicht ohne Gefahren.

(öffentlicher Schlüssel) Um Bob eine Nachricht zu senden, fordert Alice den öffentlichen Teil von Bob's Schlüssel an.
(Verschlüsselung) Mit Bob's öffentlichen Schlüssel kann Alice den von ihr zu versendenden Klartext verschlüssel. Sie hat nun keine Möglichkeit mehr den Klartext aus dem Chiffretext zu erhalten. Nur Bob ist dazu in der Lage (s.u.).
(Hash) Um die Integrität zu gewährleisten, berechnet Alice einen Hash des Klartextes und verschlüsselt ihn mit ihrem eigenen privaten Schlüssel um eine digitale Signatur zu erhalten.
(Versendung) Alice schickt Bob den Chiffretext und die digitale Signatur.
(Entschlüsselung) Dieser entschlüsselt den Chiffretext mit Hilfe seines privaten Schlüssels und erhält so den Klartext.
(Klartext-Hash) Aus dem Klartext berechnet er mit dem gleichen Verfahren wie Alice einen Hash.
(Signatur-Hash) Mit Hilfe von Alice's öffentlichen Schlüssel entschlüsselt Bob ebenfalls den von Alice in der digitalen Signatur mitgeschickten Hash-Wert des ursprünglichen Klartextes.
(Hash-Vergleich) Ein Vergleich der Hash-Werte zeigt, ob der ursprüngliche Klartext bei der Vermittlung verändert wurde oder nicht.

Referenzen:

[Wi05] P. J. Windley, Digital Identity – Unmasking Identity Management Architecture (IMA), Beijing, Köln: O'Reilly, 2005.



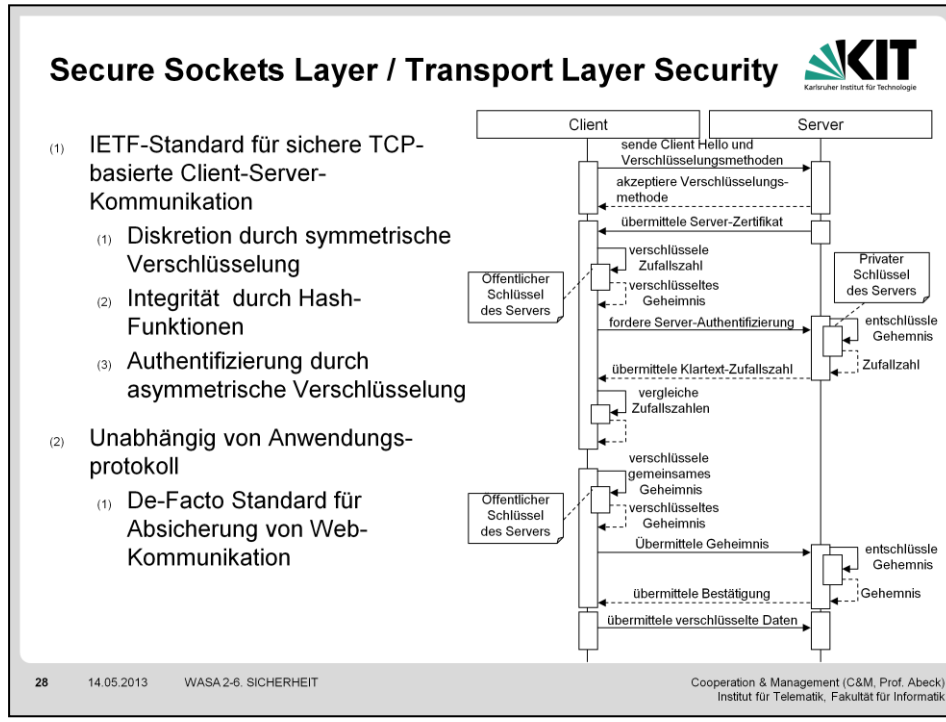
(1) Verlässt die Kommunikation zwischen Softwaresystemen das interne Netz einer Organisation, welches ggf. abgesichert ist, muss auf die Vertraulichkeit und Integrität der übermittelten Daten geachtet werden. Insbesondere Web-Anwendungen tauschen Daten über das Internet aus, in welchem die übermittelten Nachrichten von Dritten abgefangen und manipuliert werden können. Verschlüsselungsverfahren sind relativ einfach umzusetzen, trotz des Performanzverlustes. Jedoch müssen diese in den Prozess der Nachrichtenübermittlung eingebaut werden.

(2) Der sichere Kanal ermöglicht das Versenden und Empfangen von verschlüsselten Informationen zwischen Client, z.B. ein Browser, und einem (Web-) Server. Wie bei den zuvor beschriebenen Verschlüsselungsverfahren wird hierzu ein Schlüssel zur Ver- und Entschlüsselung der Daten eingesetzt. Um einen Performanzverlust vorzubeugen, sollte der sichere Kanal lediglich zum Austausch von schützenswerte Informationen genutzt werden. Nicht sensitive Informationen sollten weitesgehend über einen ungeschützen Kanal gesendet werden. Die Schwierigkeit beim sichern Kanal liegt im Schlüsselaustauschverfahren, welches selbst sicher sein sollte, um mögliche Angriffe durch gestohlene Schlüssel zu vermeiden.

(3) Weitere Beispiel für sichere Kanäle auf unterschiedlichen Kommunikationsschichten sind z.B. IPSec, Virtual Private Networks und WS-Security.

Referenzen:

[SF+05] M. Schumacher, E. B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns. Chichester, England: John Wiley & Sons Ltd, 2005, p. 565.



(1) Transport Layer Security (TLS), früher auch Secure Sockets Layer (SSL) genannt, ist eine Erweiterung für Protokolle der Anwendungsschicht, welche auf TCP basieren. Somit können nicht nur HTTP-Verbindungen abgesichert werden, wobei dies jedoch das bekannteste Einsatzszenario ist [IETF-RFC-2246]. Es ermöglicht die verschlüsselte Verbindung zwischen zwei Kommunikationspartnern, eine sogennante Punkt-zu-Punkt-Verbindung, Wird die Nachricht über mehrere Zwischenstellen gesendet, so muss zwischen jeden zwei Verbindungsknoten eine separate TLS-Verbindung erstellt werden. Eine TLS-Verbindung verschlüsselt die gesamte Nachricht.

(1.1)-(1.3) TLS nutzt sowohl symmetrische als auch asymmetrische Verschlüsselungsverfahren um eine gesicherte Verbindung herzustellen. Hierbei wird ein asymmetrisches Verfahren eingesetzt, um einmalig zu Beginn des Nachrichtenaustausches die Authentizität von Server und Client festzustellen und ein gemeinsamen symmetrischen Schlüssel auszustauschen. Aufgrund der Performanzvorteils von symmetrischen Verfahren, werden diese bei der Ver- und Entschlüsselung der übermittelten Daten eingesetzt.

(Client Hello) Möchte ein Client eine gesicherte Verbindung zu einem Server aufbauen, so sendet er diesen eine entsprechende Anfragenachricht. Danach werden die Verschlüsselungsverfahren für die Authentifizierung und Datenverschlüsselung ausgehandelt.
(Serverzertifikat) Der Server sendet dem Client sein Server-Zertifikat, eine durch eine vertrauenswürdige dritte Instanz unterschriebene Kopie des öffentlichen Schlüssels des Servers.
(Server-Authentifizierung) Um die Authentizität des Servers festzustellen, sendet der Client eine mit dem öffentlichen Schlüssel des Servers verschlüsselte Zufallszahl an den Server und erwartet die entschlüsselte Kopie als Antwort.
(Geheimnisaustausch) Anschließend wird, bei erfolgreicher Serverauthentifizierung, gemäß der ausgehandelten Verschlüsselungsverfahren ein neuer symmentrischer Schlüssel generiert. Dieser wird mittels des öffentlichen Schlüssels des Servers verschlüsselt und an de Server gesendet. Nun sind beide Parteien im Besitz eines identischen symmetrischen Schlüssels und können diesen zur Versendung von chiffrierten Nachrichten nutzen.

Abkürzungen

TLS	Transport Layer Security
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol

Referenzen

[Wi05] P. J. Windley, Digital Identity – Unmasking Identity Management Architecture (IMA), Beijing, Köln: O'Reilly, 2005.

LZ KOMMUNIKATION- UND INFORMATIONSICHERHEIT – ÜA Informations- und Kommunikationssicherheit

(1) Welche kryptographischen Verfahren kennen Sie?
  (1) Worin besteht der wesentliche Unterschied?
  (2) Was sind deren Vor- und Nachteile und wann sollten Sie eingesetzt werden?

(2) Erläutern Sie grob den Ablauf einer symmetrischen / asymmetrischen Verschlüsselung.

(3) Welche bekannte Lösung für das Problem des sicheren Nachrichtenaustauschen gibt es?

(4) Transport Layer Security ...
  (1) verschlüsselt nur HTTP-Verbindungen
  (2) verschlüsselt Punkt-zu-Punkt-Verbindungen
  (3) verschlüsselt nur die sensitiven Teile einer Nachricht