LEHRSTUHL 14
SOFTWARE ENGINEERING

Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2013

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

3.1: Modellbasierte Sicherheit

v. 26.06.2013

3.1 Modellbasierte Sicherheit

Literatur:
[Jür05] Jan Jürjens: **Secure systems development with UML**, Springer-Verlag 2005.
Unibibliothek (e-Book): http://www.ub.tu-dortmund.de/katalog/titel/1361890
Papier-Version: http://www.ub.tu-dortmund.de/katalog/titel/1091324

IT-Sicherheit

- Wirtschaft, Unternehmen und Gesellschaft hängen zunehmend ab von Computernetzwerken für Kommunikation, Finanzen, Energieversorgung, Transport...
- Angriffe können großen finanziellen Schaden verursachen.
- Vernetzte Systeme können anonym und aus der Entfernung angegriffen werden.
- Computersysteme müssen also sicher sein.

Problem: Sicherheit

- Sicherheit (Security): Schutz von Daten oder Systemen gegen mutwillige Angriffe. Inhärent schwierig (zielorientierter Angreifer). Beispiel (1997):
- NSA Hacker Team bricht in U.S. Department of Defense Computer und U.S. Strom-versorgungssystem ein. Demonstriert Strom- und Notrufausfälle in Washington, D.C..

Beispiele für Sicherheitsvorfälle

- Einbruch in die Website SalesGate.com, Diebstahl von 3.000 Kundendateien (z.B. Kreditkartennummern). Z.T. im Internet veröffentlicht.
- Unkontrollierte Weiterleitung persönlicher Informationen aus Hersteller-Sites (z.B. Finanzrechenprogramme auf Intuit) zu Anzeigen-Sites (wie DoubleClick) ohne Wissen der Benutzer und oder von Intuit.
- Februar 2000: massive Denial-of-Service Angriffe.

[Schneier: Secrets & Lies]

Softwareschwächen (09.11. - 01.12.2004)

- Microsoft schließt das IFrame-Loch (1.12.2004, ju). Mit einem überraschenden Update beseitigt Microsoft das IFrame-Problem des Internet Explorer 6.0.
- Buffer Overflow in Suns Ping-Befehl (1.12.2004, dab). Sun weist in einem Advisory auf einen Buffer Overflow im Ping-Befehl hin, mit dem angemeldete Nutzer unter Umständen ihre Zugriffsrechte erhöhen können.
- Cross-Site-Scripting-Schwachstelle in Linux-Firewall IPCop (1.12.2004, dab). Durch eine Cross-Site-Scripting-Schwachstelle in IPCop kann ein Angreifer das Authentifizierungscookie des Administrators stehlen und sich damit später ohne Kenntnis des Passwortes an der Firewall anmelden.
- Server des CCC gehackt (29.11.2004, pab). Spanische Hacker sind in die Server des Chaos Computer Clubs eingedrungen und haben unter anderem die Registrierungsdaten vom CCC-Camp 2003 veröffentlicht.
- Windows-Namensdienst verwundbar (29.11.2004, ju). Im Windows-Namensdienst WINS gibt es einem Advisory von Nicolas Waisman zufolge eine Schwachstelle, über die ein Angreifer beliebigen Code einschleusen und ausführen kann.
- SQL-Injection-Lücke in PHPNews beseitigt (25.11.2004, dab). In Version 1.2.4 der Board-Software PHPNews wurde eine SQL-Injection-Schwachstelle beseitigt.
- Lücke in Suns Java Plug-ins gewährt Zugriff auf das System (23.11.2004, dab). Durch einen Fehler in Suns Java Plug-ins für Browser können Angreifer mit präparierten Java Applets aus der Sandbox ausbrechen und die Kontrolle über den Rechner erlangen. Betroffen sind alle Browser, die Suns Plug-in einsetzen.
- Vergiftete Websites [Update] (22.11.2004, ju). Langsam lichtet sich der Nebel um die IFrame-Attacken vom Wochenende. Falk eSolutions leitete offenbar Zugriffe auf seine Ad-Server auf einen kompromittierten Server um, der Trojaner auf den Systemen der Anwender installierte.
- Zone Labs beseitigt DoS-Schwäche in Firewall-Produkten (19.11.2004, dab). Der Hersteller Zone Labs weist auf seinen Seiten auf eine Schwachstelle in seinen Firewall-Produkten hin, durch die das System zum Stillstand kommen kann.
- Samba-Entwickler schließen kritische Lücke -- ohne darauf hinzuweisen [Update] (15.11.2004, dab). Weil die Entwickler von der Ausnutzbarkeit eines Fehlers nicht überzeugt waren, beseitigten sie die Lücke, ohne darauf in einem Advisory hinzuweisen. Ein Sicherheitsexperte will aber dafür einen Exploit entwickelt haben, der Code auf dem Server ausführt.
- Angeblich zehn Sicherheitslücken in Service Pack 2 für Windows XP (12.11.2004, dab). Der Hersteller Finjan hat nach eigenen Angaben zehn gravierende Sicherheitslücken in Windows XP Service Pack 2 festgestellt. Damit sollen Hacker relativ einfach in Netzwerke eindringen und die Kontrolle über Clients gewinnen können.
- DHCP-Pakete blockieren Netzwerkschnittstellen auf Cisco-Routern (11.11.2004, dab). Cisco hat eine Schwachstelle in Geräten mit IOS-Version 12.2s gemeldet. Fehlerhafte DHCP-Pakete können die Eingangsqueue einer Netzwerkschnittstelle verstopfen, sodass der Router keine an ihn direkt gerichteten Pakete mehr annimmt.
- Update behebt Schwachstelle in Microsoft ISA und Proxy Server (9.11.2004, dab). Ein Angreifer kann einen Fehler im DNS-Cache des ISA und Proxy Server ausnutzen, um Anwender auf falsche Web-Seiten umzuleiten.
- Suns Messaging Server gewährt unautorisierten Zugriff auf Webmail-Konten (9.11.2004, dab). Die Webmail-Funktion in Suns iPlanet Messaging Server und Sun ONE Messaging Server gewährt unter besonderen Umständen Angreifern Zugriff auf Mail-Konten.
- Fehler in Ruby CGI-Modul bringt System zum Stillstand (9.11.2004, dab). Im CGI-Modul cgi.rb der objekt-orientierten Skriptsprache Ruby wurde eine Schwachstelle entdeckt, mit der Angreifer über das Netzwerk das komplette System zum Stillstand bringen können.
- Denial-of-Service-Schwachstelle in Samba-Server (9.11.2004, dab). Dateinamen mit zu vielen Wildcard-Zeichen erhöhen die Prozessorlast so stark, dass der Server nicht mehr antwortet.

[Heise security, 1.12.2004]

Gehackte Web-Seiten, 06.12.2004 bis Mittag

Todays reported and verified attacks: 1204 of which 352 are single IP and 852 mass defacements

Time	Attacker					Domain	OS	View		
•14:58	PHTeam					....com/cgi-bin/index.cgi		FreeBSD		view | mirror
•14:56	hackbsd crew		H	M		automondial.ro		Linux	view | mirror	
•14:53	BI0S	H				hosting2b.com		Linux	view | mirror	
•14:53	BI0S	H	M			statebase.com		Linux	view | mirror	
•14:52	Next Time		H			hightechtoys.it		Linux	view | mirror	
•14:47	Antrax	H				healthlawtoday.com		Linux	view | mirror	
•14:46	Antrax	H				mosessinger.com		Linux	view | mirror	
•14:45	DeF4x0rz Group					miawolf.net/guestbook		Linux	view | mirror	
•14:39	DeF4x0rz Group					...flats.com.br/guestbook		Linux	view | mirror	
•14:37	BI0S	H	M			psynix.com		Linux	view | mirror	
•14:37	Q8Crackers		H			groovetx.com		Linux	view | mirror	
•14:36	BI0S	H				tamingfire.com		Linux	view | mirror	
•14:35	BI0S	H				carpet24.com		Linux	view | mirror	
•14:35	NaOnaK	H				forums.deeko.com		Linux	view | mirror	
•13:51	Logicb0x		M			pcxp.piusx.net/index.htm		Win 2000	view | mirror	
•13:51	Logicb0x					...s.wnyric.org/index.htm		Win 2000	view | mirror	
•13:51	Logicb0x					...rn.k12.or.us/index.htm		Win 2000	view | mirror	
•13:50	KERANGKA|LANGIT				M	...udi.gov.cn/igenus/temp		NetBSDOpenBSD	view | mirror	
•13:50	Logicb0x					pcxp.usd262.net/index.htm		Win 2000	view | mirror	
•13:49	Next Time			M		...o.ch.it/media/next.htm		Linux	view | mirror	
•13:49	Logicb0x					pcxp.usd437.net/index.htm		Win 2000	view | mirror	
•13:48	DeF4x0rz Group				R	bodamagica.com/visitas		Linux	view | mirror	
•13:48	DeF4x0rz Group					emcorner.it/book		Linux	view | mirror	
•13:47	Logicb0x					...aschools.org/index.htm		Win 2000	view | mirror	
•13:46	SyRiaN_HacKerZ		H			syria4you.com		Linux	view | mirror	
•13:41	NaOnaK	H				phantom-legion.net		Linux	view | mirror	
•13:04	Simiens	H				nexusthegame.com		FreeBSD	view | mirror	
•13:03	Logicb0x					...ll.k12.pa.us/index.htm		Win 2000	view | mirror	
•13:02	BI0S	H				uralmebel.ru		FreeBSD	view | mirror	
•12:59	batistuta		M			moe.go.th/moego		Linux	view | mirror	
•12:59	Logicb0x					pcxp.lex2.org/index.htm		Win 2000	view | mirror	
•12:57	BI0S	H	M			bayareamarine.com		Linux	view | mirror	
•12:55	BI0S	H	M			erwinsautosales.com		Linux	view | mirror	
•12:55	BI0S	H	M			boxlaser.com		Linux	view | mirror	
•12:55	BI0S	H	M			locallaw1.com		Linux	view | mirror	

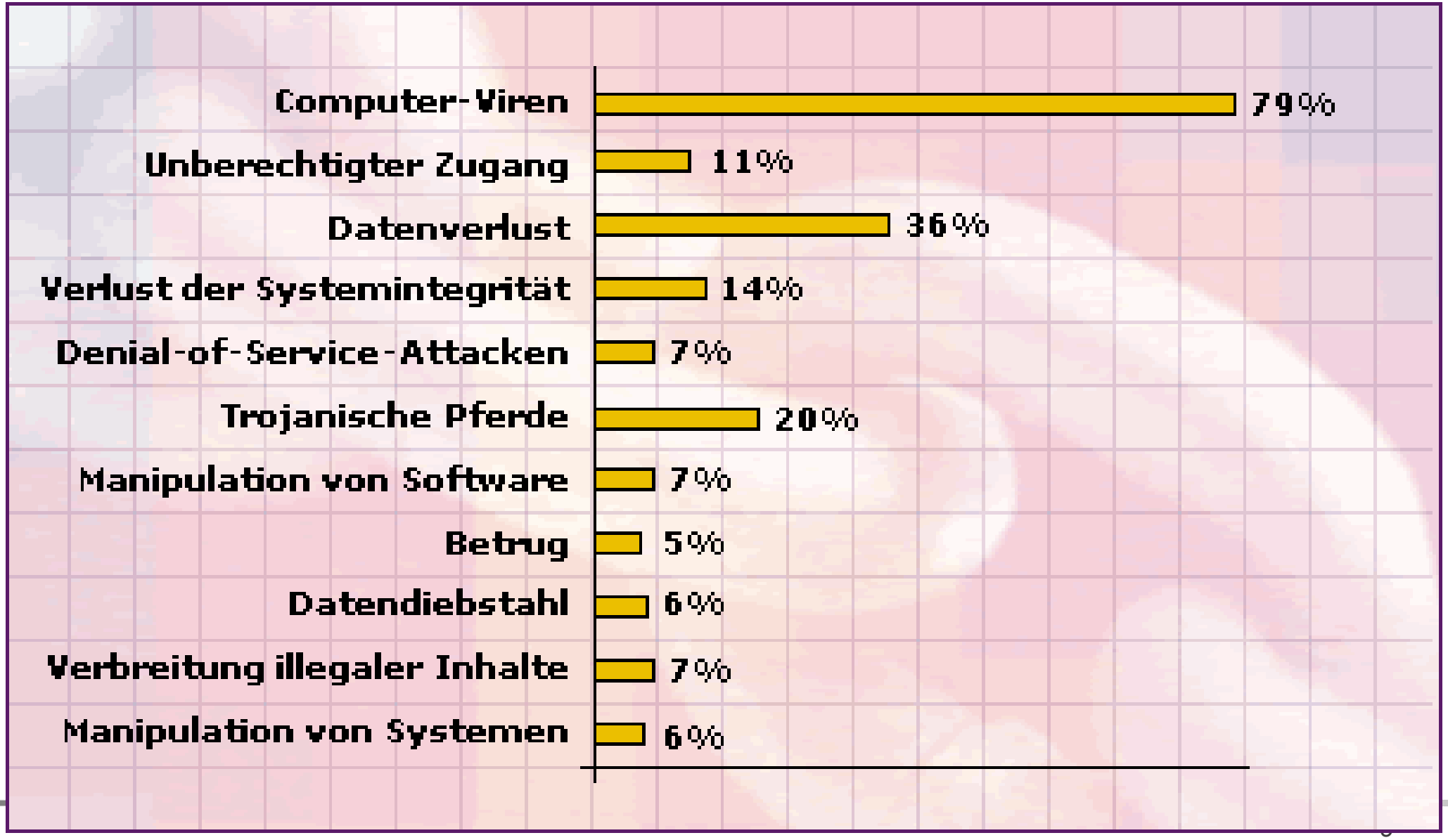
[www.zone-h.org]

Aktuelleres Beispiel (12.06.11): Hacker attackieren den Währungsfond

http://www.zeit.de/digital/datenschutz/2011-06/hacker-iwf-wirtschaftsdaten/komplettansicht?print=true

Hacker haben den Internationalen Währungsfonds angegriffen und offenbar Daten gestohlen. Der IWF geht von Spionage aus und macht eine "bestimmte Regierung" verantwortlich.
Der Internationale Währungsfonds (IWF) ist Opfer einer Attacke auf seine Computer geworden. Nach einem Bericht von Bloomberg News wurden bei dem Angriff E-Mails und weitere Dokumente gestohlen. Der Fonds habe Ermittlungen eingeleitet, wie es zu dem Hacker-Angriff kommen konnte, erklärte ein IWF-Sprecher. Die Arbeit der Organisation sei durch den Angriff aber nicht beeinträchtigt. Über das Ausmaß des Schadens machte der Sprecher keine Angaben.
Ein Sprecher der Weltbank sagte, man habe alle Netzwerk-Verbindungen zur Schwester-Organisation gekappt. Es handle sich dabei aber um eine reine Vorsichtsmaßnahme, bis man die Attacke und ihre Auswirkungen genauer verstanden habe. Das Netzwerk ermöglicht den Austausch von Informationen zwischen beiden Organisationen, wird aber nicht für die Kommunikation geheimer Informationen oder sensibler Finanzdaten genutzt.
Auch FBI leitete inzwischen eine Untersuchung ein. Die Bundespolizei arbeite eng mit dem IWF zusammen, erklärte eine Sprecherin des US-Verteidigungsministeriums. Das FBI selbst lehnte eine Stellungnahme ab.
Nach Angaben des Internet-Sicherheitsexperten Tom Kellermann, der in dieser Funktion auch für den IWF und die Weltbank gearbeitet hat, zielte der Hackerangriff darauf, heimlich eine Software zu installieren, um einer bestimmten Regierung Zugang zu Insider-Informationen des IWF über andere Länder zu verschaffen. Um welche Regierung es sich handle sei noch unklar.
Der Angriff habe sich über mehrere Monate ereignet, zitierte die New York Times einen namentlich nicht genannten IWF-Mitarbeiter. "Das war ein sehr großer Einbruch." Die Organisation wollte offiziell jedoch nichts dazu sagen, wie umfangreich der Angriff war und welches Ziel er hatte.
Die Zeitung spekuliert über die Art des Eindringens, es habe sich dabei um "spear phishing" gehandelt. Beim Phishing werden unbedarfte Nutzer meist in einer Mail aufgefordert, Passworte und Zugangscodes auf Seiten einzugeben, die aussehen wie echte – beispielsweise für das Onlinebanking – jedoch echten nur nachempfunden sind, um diese Codes abzufangen. Spear phishing ist eine gezieltere Form dieses Verfahrens. Die Mail kommt dabei von jemandem, den der Nutzer kennt und dem er vertraut, der IT-Abteilung des Hauses beispielsweise.

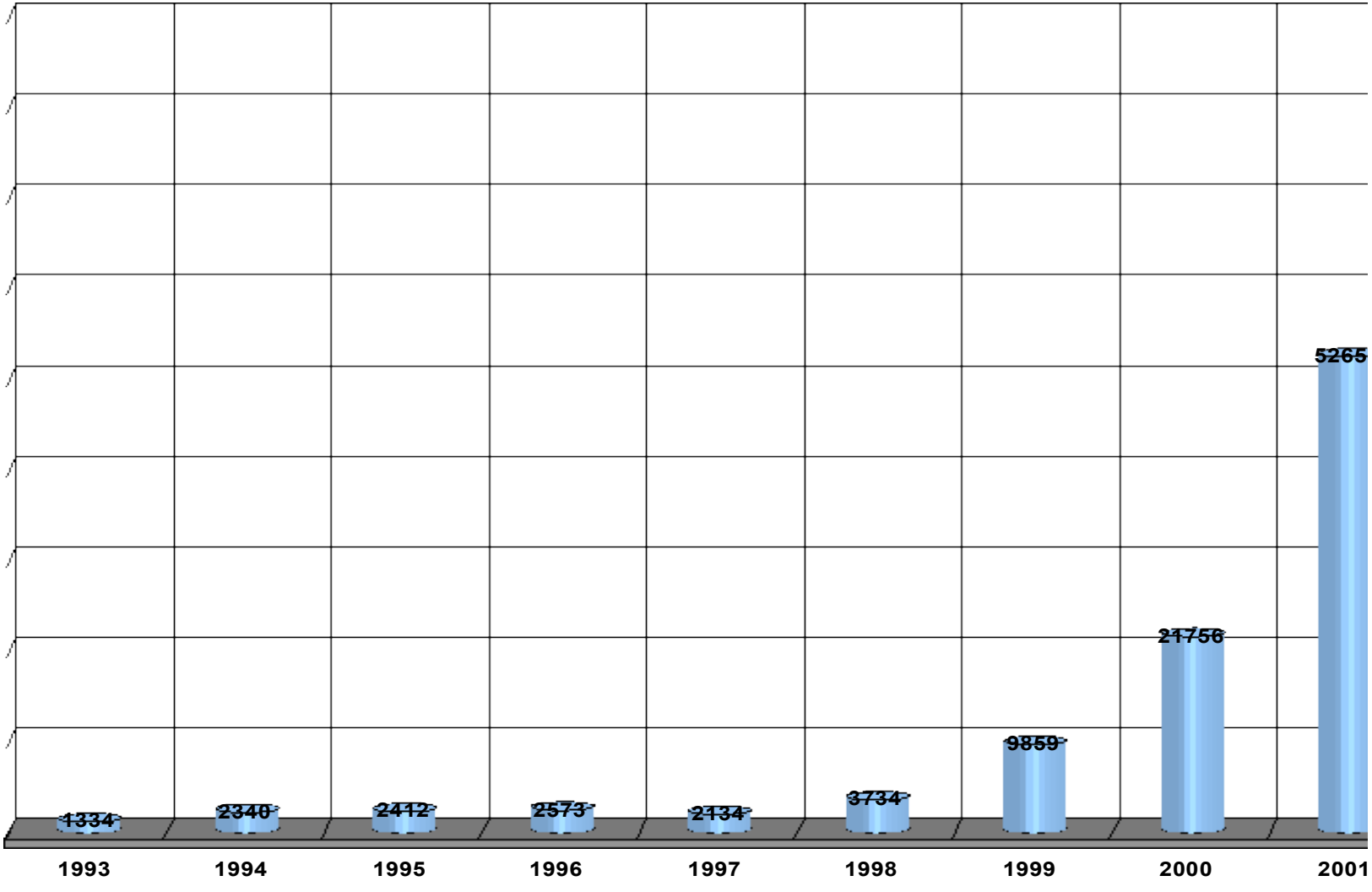
Arten von Sicherheitsproblemen



[silicon.de, 2001]

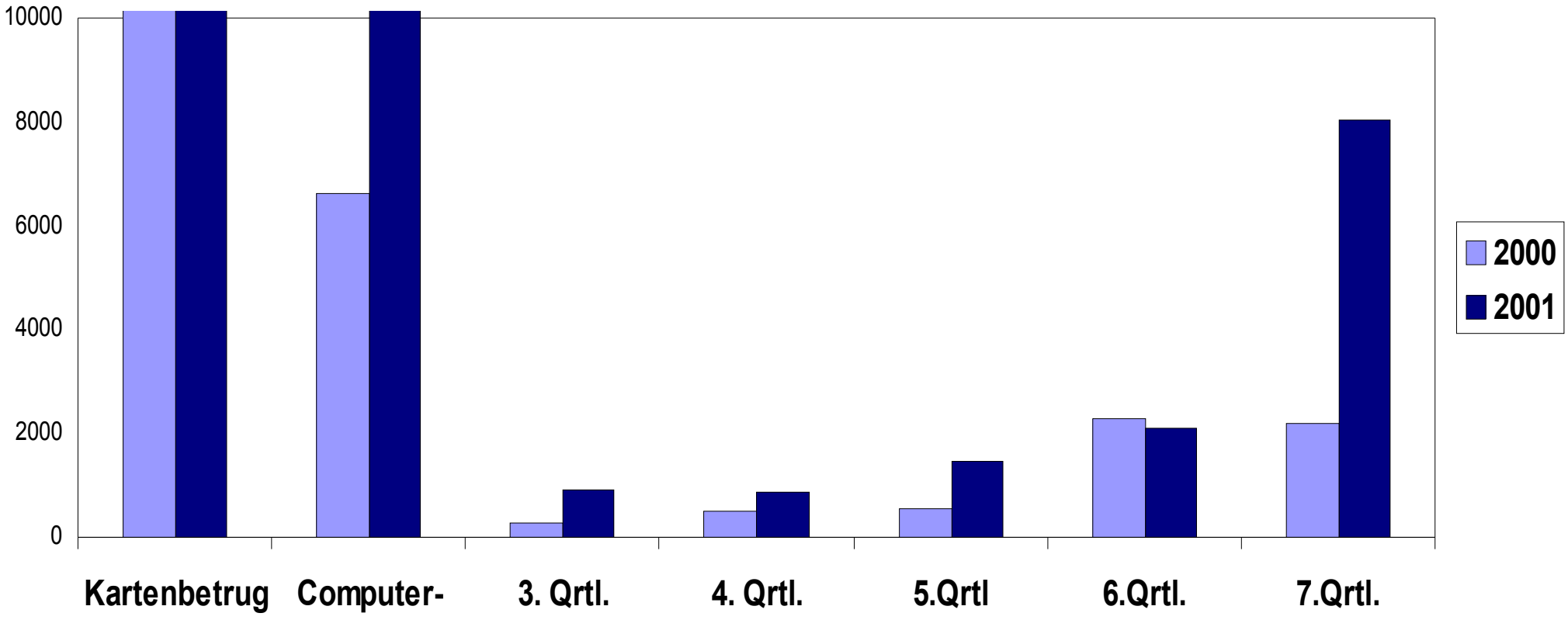
Gemeldete IT-Sicherheitsvorfälle

[www.cert.org 2002]



Kriminalstatistik

[Bundesinnenmin.]



Karten-betrug

Computer-betrug

Daten-fälschung

Computer-sabotage

Auspähen von Daten

Software-piraterie

Betrug mit Kommunika-tionsdiensten

IT-Risiken vs. KontraG/Basel II

- Basel II (bis 2006): risikogerechtere Regelung der Eigenkapitalanforderungen
- Genauere Analysemethoden (Kreditrisiko, operationelles Risiko, internal-ratings-based). Offenzulegen.
- Insbesondere IT Risiken (unexpected loss, z.B. Virenbefall, Hackerangriff, …)

Ursachen I

- Korrektes Entwerfen sicherer Systeme ist schwierig.
- Selbst Experten können sich irren:
  - Needham-Schroeder-Protokoll (1978)
  - Gefundene Angriffe 1981 (Denning, Sacco), 1995 (Lowe)
- Designern fehlt es oft an Erfahrung in Sicherheit.
- Sicherheit im Nachhinein.
- Angriffs-Informationen verbreiten sich schnell.
- Keine Rückmeldung über die Sicherheit aus Kundensicht.

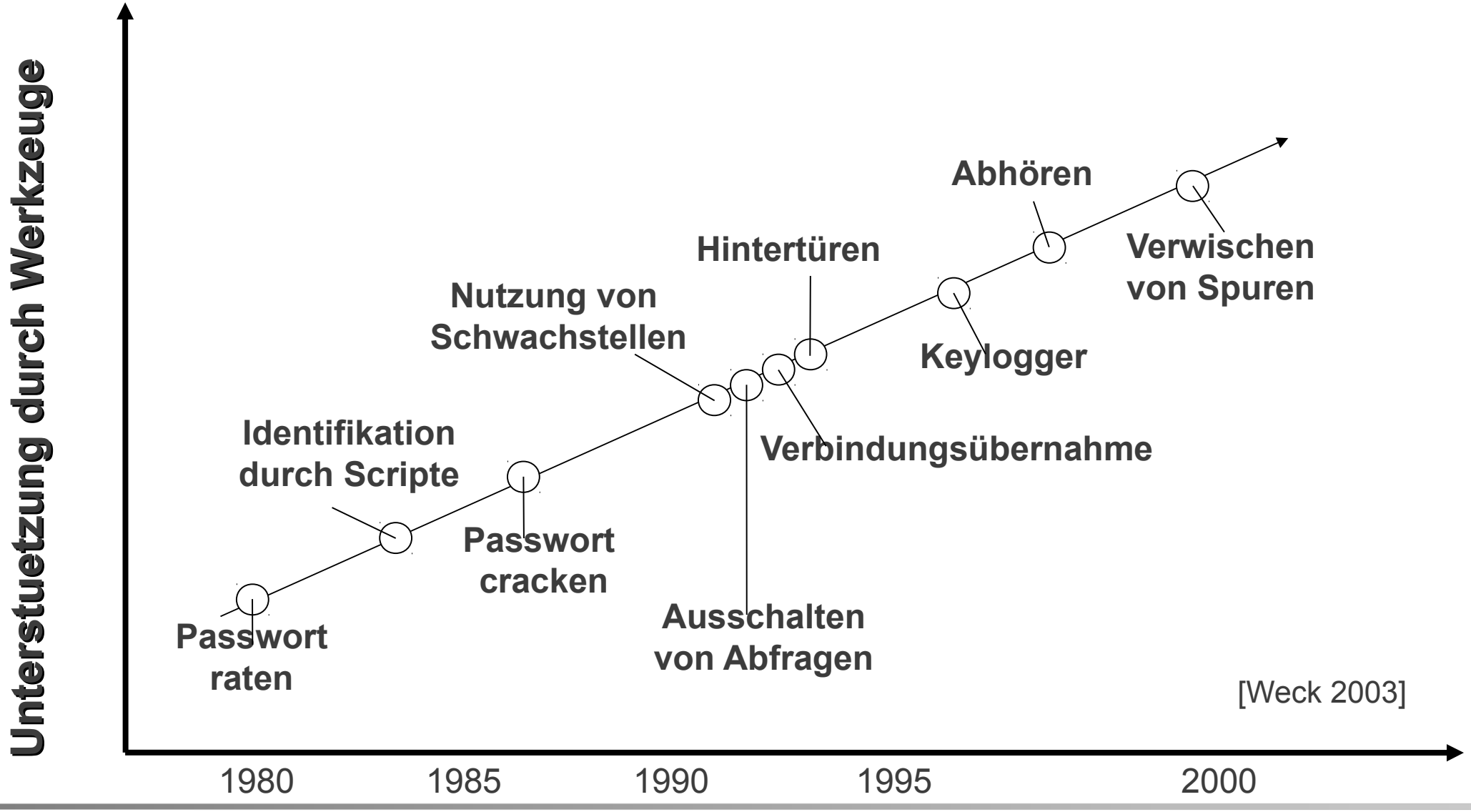
Ursachen II

„Blindes“ Vertrauen in Mechanismen:

- Sicherheitstechnik ist gefährdet, da sie meist umgangen werden kann (ohne dass sie gebrochen werden muss)

- Annahmen in Bezug auf den System-Kontext und die physische Umwelt.
  - "Diejenigen, die glauben ihre Probleme durch den einfachen Einsatz von Kryptographie lösen zu können, verstehen diese nicht, und verstehen ihr Problem nicht" (R. Needham)

Immer bessere Hackerwerkzeuge



[Weck 2003]

Sichere Systeme

- Analyse von sicherheitskritischen Systemen schwierig (motivierter Angreifer).
- Viele entwickelte und eingesetzte Systeme genügen nicht den Sicherheitsanforderungen.
- Sichere Produkte oft auf unsichere Weise eingesetzt.
- Viele z.T. spektakuläre Angriffe.
- Problem: Qualität vs. Kosten.

Softwaretechnik & Sicherheit

- "Penetrate-and-patch"

  (auch genannt "Bananen-Strategie“):
  - unsicher
  - störend
- Traditionelle formale Methoden: teuer.
  - Ausbildung von Entwicklern
  - Erstellung formaler Spezifikationen.

Problem: Sicherheit ist schwierig

- Klassische Schwäche in alten Unix-Systemen: "Falsches Passwort"-Meldung nach Eingabe des ersten falschen Zeichens.

  Was ist damit das Problem ?

Problem: Sicherheit ist schwierig

- Klassische Schwäche in alten Unix-Systemen: "Falsches Passwort"-Meldung nach Eingabe des ersten falschen Zeichens.
  Durch Zeitmessungs-Angriff reduziert sich der Passwort-Raum von 26^n auf 26*n (für n = Passwortlänge).
- Neuere Schwäche von Smart-Cards: Rekonstruktion des geheimen Schlüssels durch Zeit-Messung des Stromverbrauchs während der Crypto-Operationen

➔ Frage: Wie findet man diese Schwächen mit klassischen Tests?

Problem: Nicht vertrauenswürdige Programmierer

- Bei hoch-sicherheitskritischen Systemen kann nicht einmal den Programmierern vertraut werden.
- Vielleicht haben diese absichtlich eine Hintertür in den Code eingebaut.
- Es ist beinah unmöglich, diese durch Zufalls- oder Black-Box-Tests zu finden.
- Noch schlimmer, wenn versteckte Schwächen eingebaut und verwendet werden (vorherige Folie).

➔ Frage: Wie kann man dies verhindern ?

Besonderes Problem: Kryptographie

- Kryptographie spielt in vielen sicherheitskritischen Anwendungen eine wichtige Rolle.
- Per Definition muss sie sicher gegen Brute-Force-Angriffe sein.

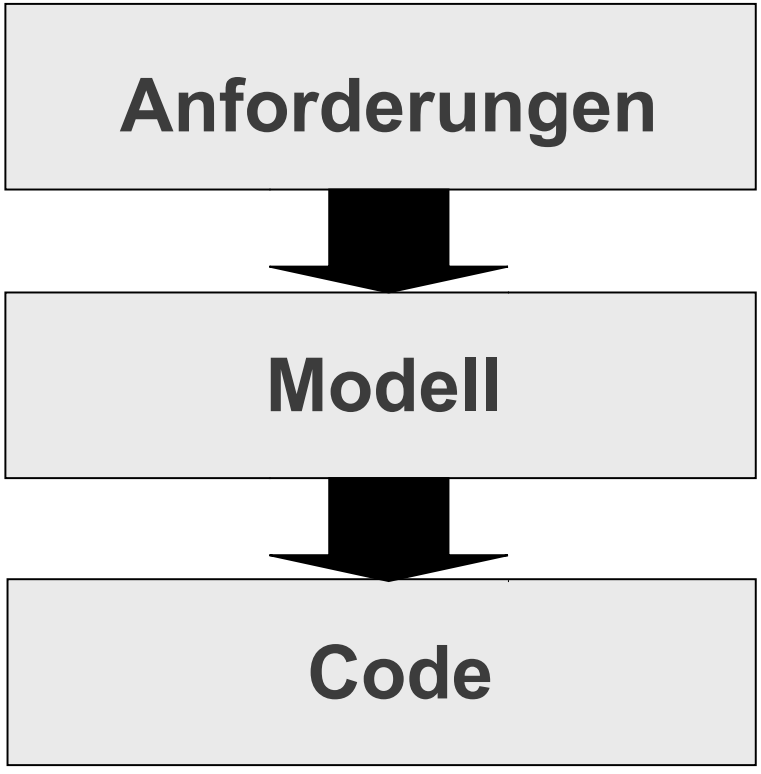
➔ Paradox: Wie bekommt man ausreichend Testabdeckung (für die aus dem Internet erreichbaren Eingabeschnittstellen für einen bestimmten Angreifer) eines Systems, um sicher gegen Brute-Force-Angriffe zu sein?

Modellbasierte Entwicklung

- Ziel: Erleichtere den Übergang von menschlichen Ideen zu ausführenden Systemen
- Qualitätssteigerung

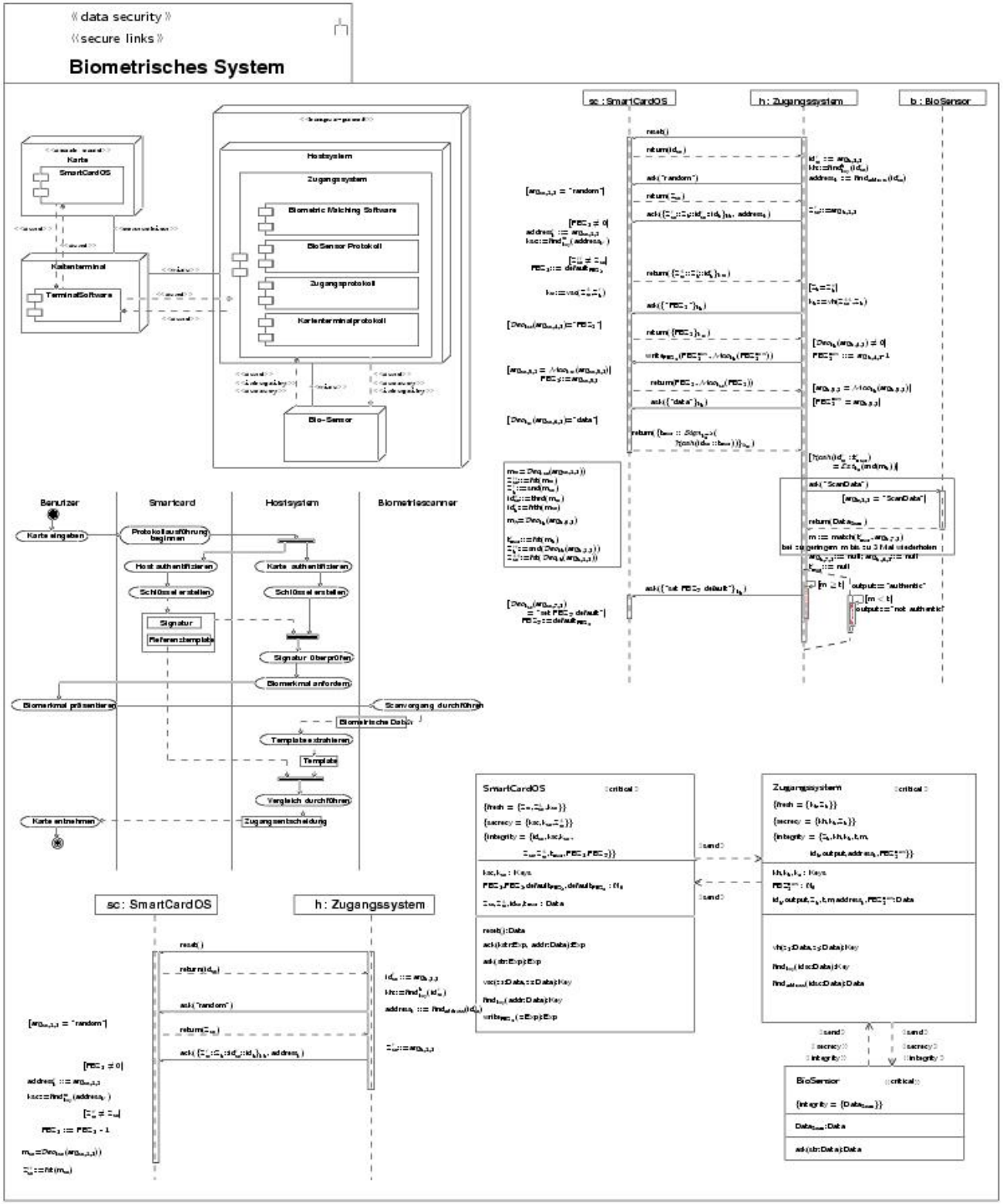
  bei beschränkten

  Zeit-zum-Markt und Kosten.

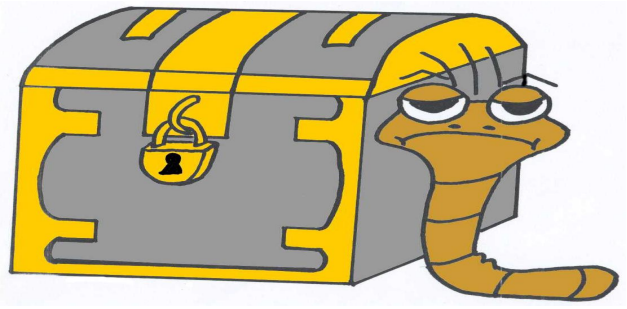


Beispiel: Biometrisches Authentifizierung-System in industrieller Entwicklung.

Sicher?



Modell-basierte Entwicklung sicherer Systeme



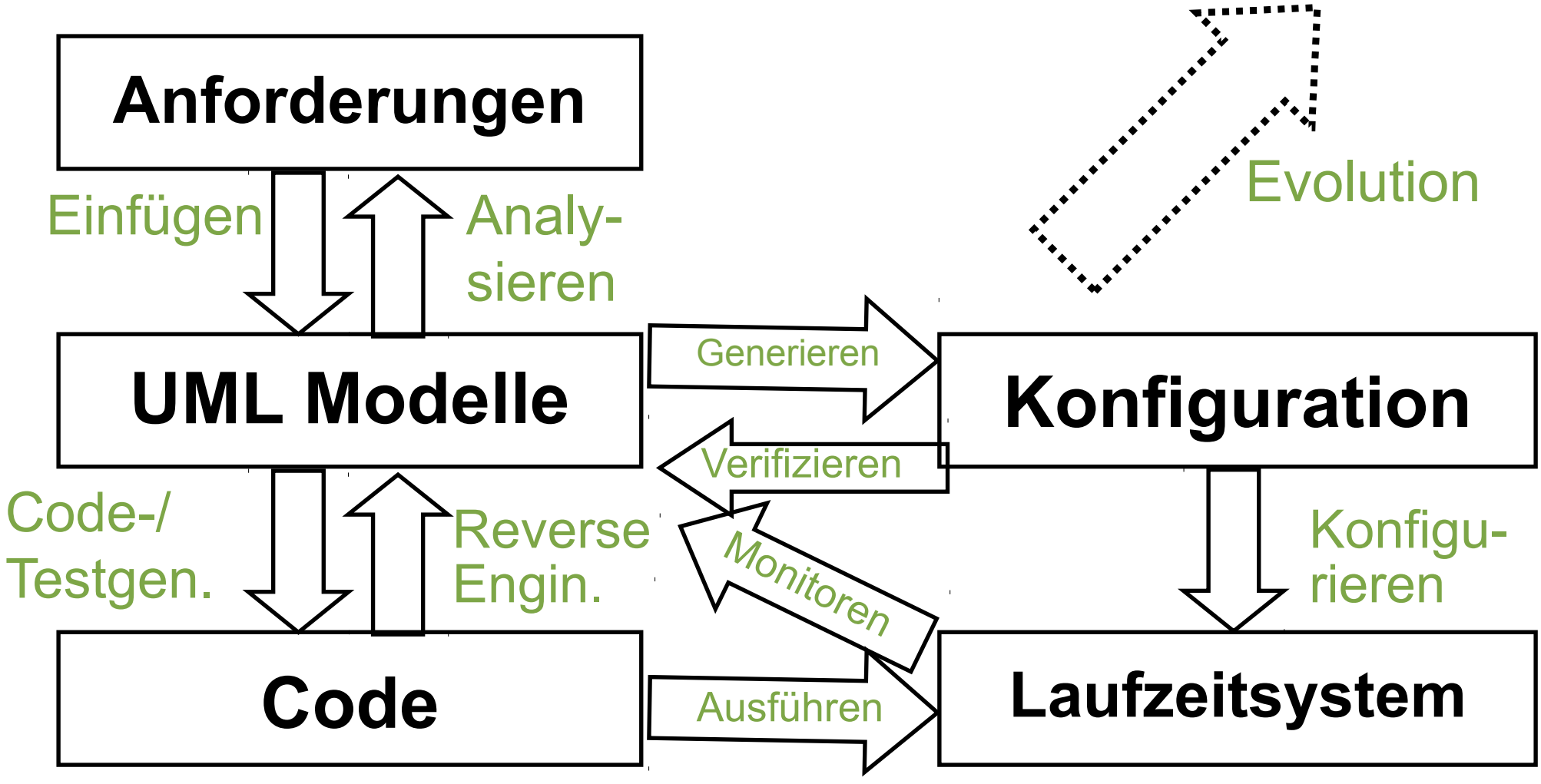
Ziel: Sicherheit erhöhen, bei begrenzter Investition an Zeit und Kosten.

Lösungsansatz:

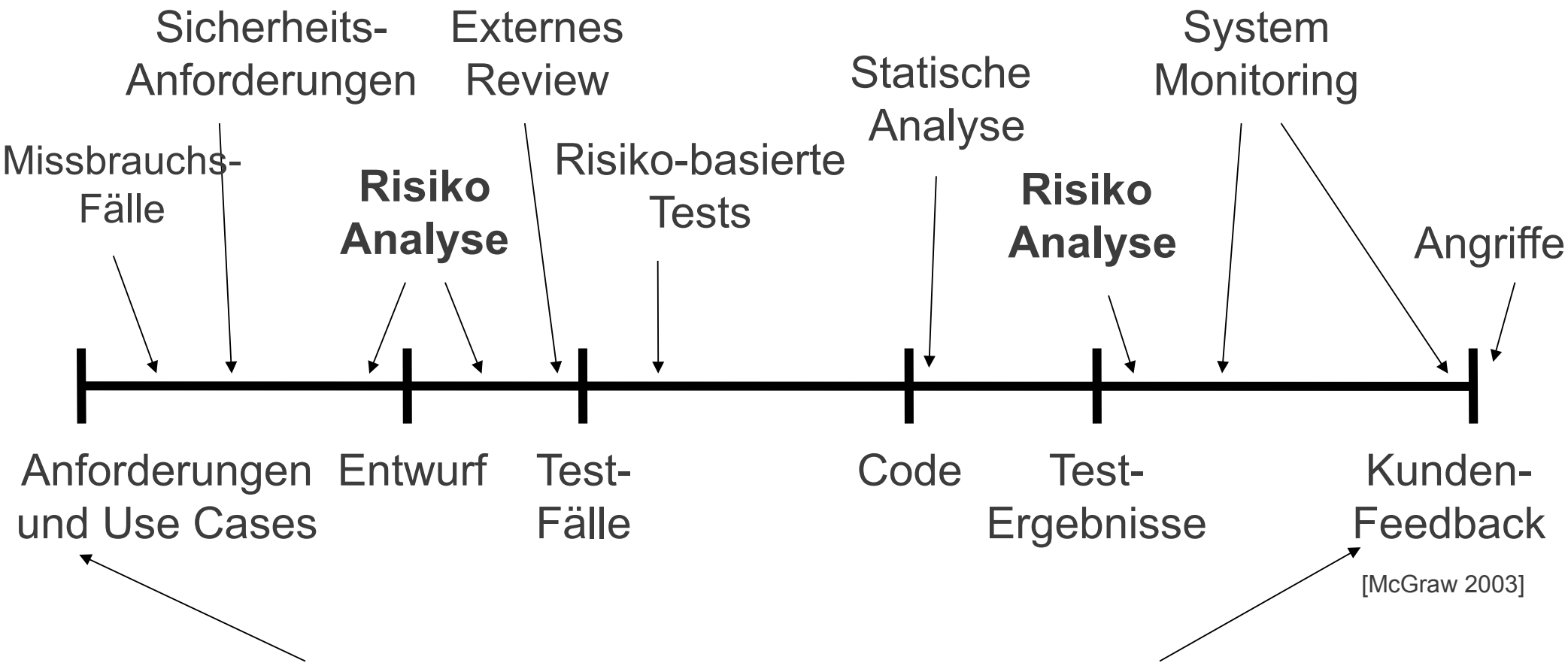
- Aus Artefakten in industrieller Entwicklung und Betrieb sicherheitskritischer Software: Modelle extrahieren (UML, Quellcode, Konfigurationen).
- Werkzeugunterstützung für theoretisch fundierte, effiziente (automatische) Sicherheitsanalyse.

➔ Modell-basierte Entwicklung sicherer Systeme

Modellbasierte Entwicklung sicherer Systeme



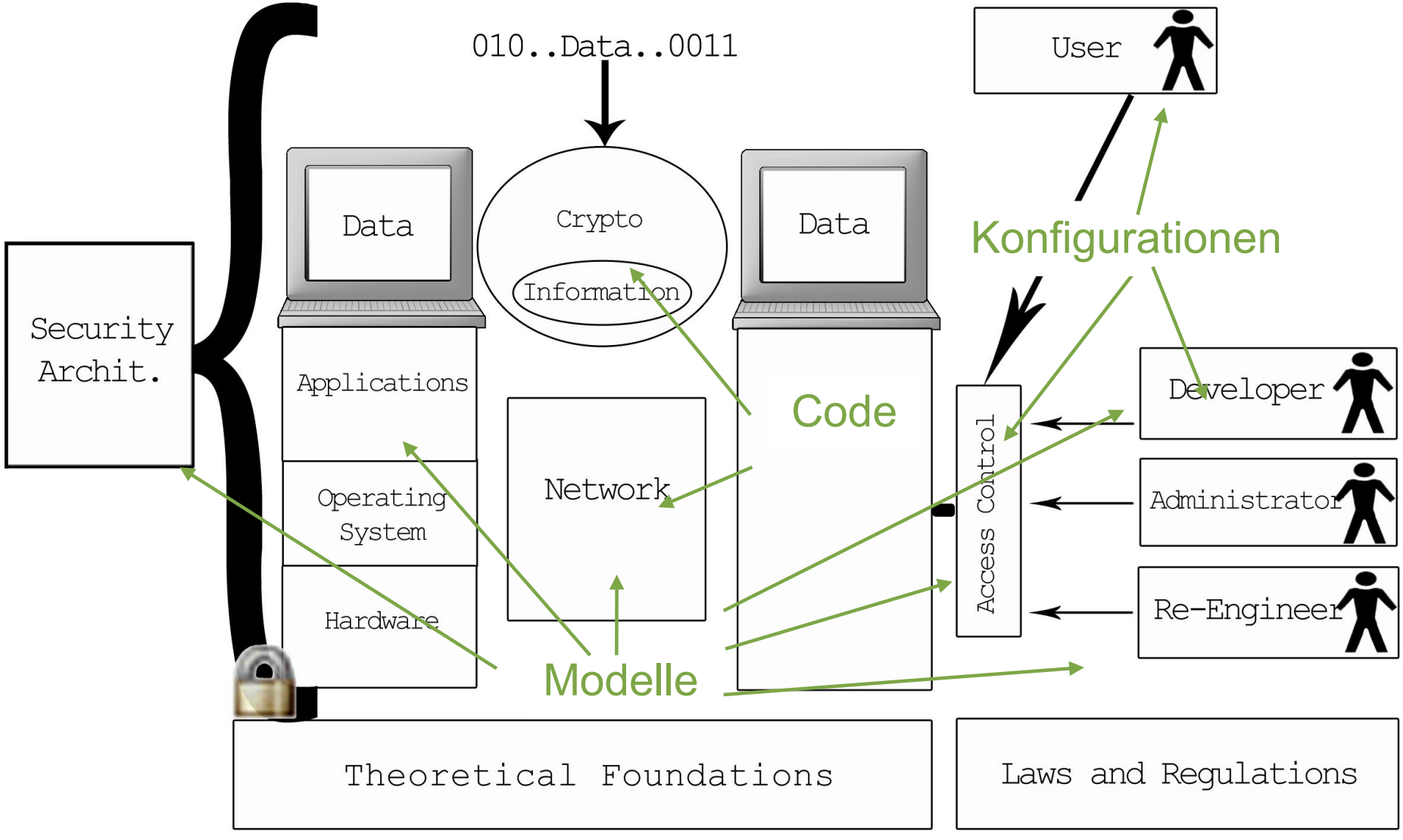
Lebenszyklus eines sicherheitskritischen Systems



[McGraw 2003]

Modell-basierte Entwicklung sicherer Systeme

Architekturebenen



Einsatzfelder

- Systementwurf
  - z.B. Architekturbewertung, Plattformenwahl, Altsystemeinbindung.
  - Automatische Sicherheits- und Risikoanalyse der modellierten Geschäftsprozesse und Softwarearchitekturen unter Einbezugnahme des Systemkontexts.
- Implementierung
  - Quellcodeanalyse, Testfolgengenerierung.
- Laufender Betrieb
  - Konfigurationsmanagement, Überprüfung von Berechtigungen, Einrichtungen von Firewalls…
  - Automatische Checks von Systemkonfigurationen (z.B. SAP-Berechtigungen, …).
- Sichere Geschäftsprozesse / Workflows: Berücksichtigung von Sicherheit ab Geschäftsprozessentwurf.
- Einsatz in Sicherheitsaudits: Erhöht Vertrauen in Korrektheit und Vollständigkeit.

Warum UML ?

Modellbasierte Sicherheitsanalyse von Geschäftsprozessen und Softwarearchitekturen mit der Unified Modeling Language (UML):

- De-facto Standard in der industriellen Modellierung. Eine große Anzahl von Entwicklern sind in UML ausgebildet.
- Einfache, intuitive Notation, relativ genau definiert.
- Weitgehende Werkzeugunterstützung (auch für Code-Generierung, Analyse, Reverse Engineering, Simulation, Transformation).

Warum UML ?

- Ziel: Übertragen von Resultaten aus der Forschung in formalen Methoden in die praktische Softwareentwicklung
- Verwendung durch Entwickler (die nicht in formalen Methoden ausgebildet sind), um die
  - Sicherheit von erstellten Komponenten zu prüfen
  - vorhandene Sicherheits-Komponenten korrekt im Systemkontext einzusetzen
  - die Systemumgebung der betrachteten Komponente zu analysieren

UML-Erweiterung UMLsec

- Erweiterung für die Entwicklung sicherer Systeme.
  - Auswertung von UML-Spezifikationen für Sicherheitsanalyse während der Entwicklung
  - beinhaltet etablierte Sicherheits-Regeln als Checkliste
  - stellt sie Entwicklern, die nicht auf sichere Systeme spezialisiert sind, zur Verfügung
  - berücksichtigt Sicherheitsanforderungen von Beginn der System-Entwicklung und im System-Kontext.
  - Sicherer Entwurf durch Modellanalyse.
  - Sichere Implementierung durch modellbasierte Testgenerierung.
  - Macht Zertifizierung kostengünstiger.

Literatur: J. Jürjens, Secure Systems Development with UML, Springer 2005 (s. Bibliothek).

UMLsec: Wie?

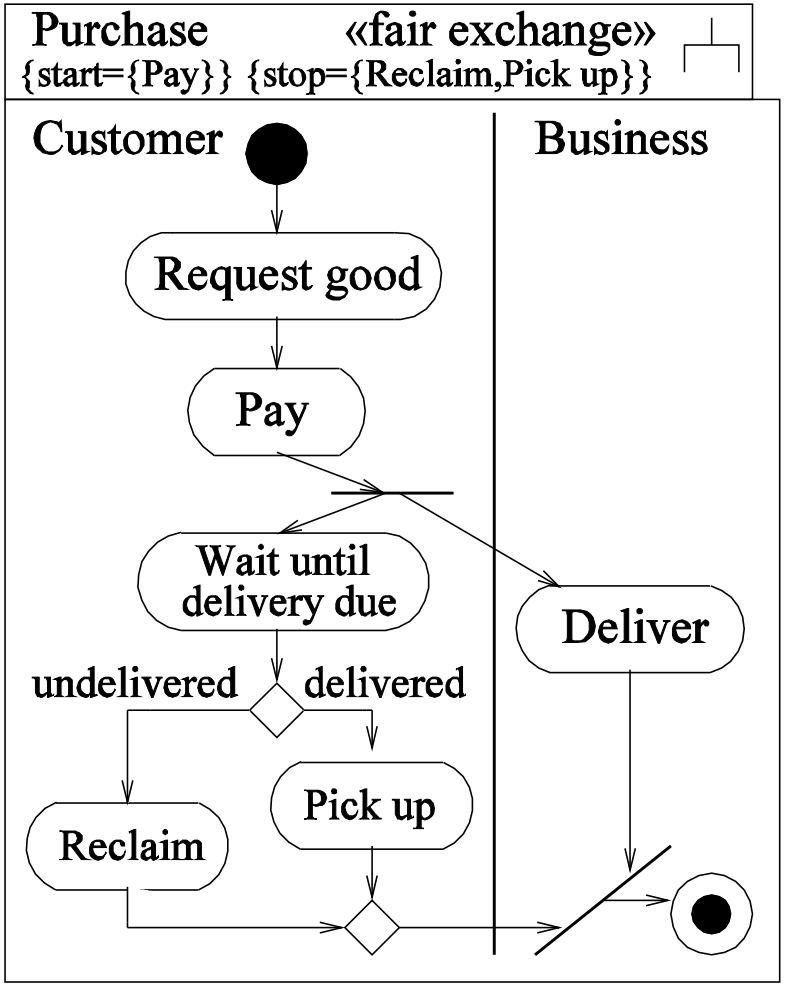
- Wiederkehrende Sicherheitsanforderungen, Angriffs-Szenarien, und Sicherheits-Konzepte werden als Stereotypen und Tagged Values angeboten.
  [Stereotypen: Markierungen an UML-Modellelementen, z.B. <<call>>. Tagged Values: Annotationen der Form (tag=value), mit denen weitere Informationen übergeben werden. => vgl. V18]
- Verwenden Constraints, um Spezifikationen unter Verwendung automatischer Werkzeuge zu verifizieren und mögliche Schwächen anzuzeigen.
  [Stereotypen können mit vordefinierten Constraints verbunden werden.]
- Garantiert, dass UML Spezifikation gewünschtes Niveau der Sicherheitsanforderungen liefert.
- Verbindung zum Code über Round-Trip.

Herausforderungen

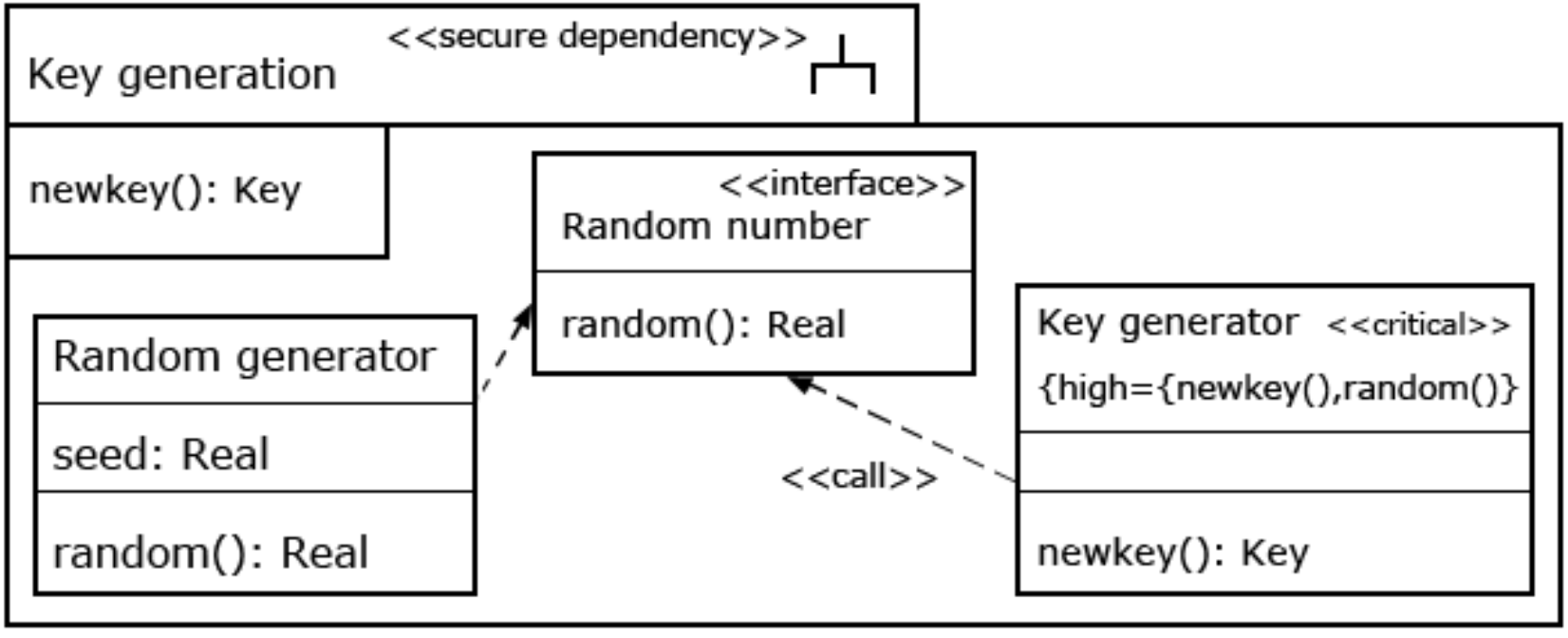
- Sichere Systeme aus (un)sicheren Komponenten ?
- Sicherheit als pervasive Eigenschaft: vs Zuverlässigkeit, Programm-Analyse, formale Methoden, Software Engineering, Programmiersprachen, Compiler, Computer-Architekturen, Betriebssysteme, reaktive Systeme, ..., ...
- Problem: keine Integration / Kohärenz.
- Wie bekommt man all diese Aspekte auf eine wasserdichte Weise im Kontext der Entwicklung sicherer Systeme zusammen? Notwendig für die Sicherheit (Angriffe auf die Grenzen zwischen Ansichten / Aspekten / Ebenen ...).

Beispiel: Sichere e-Transaktionen

- Sicherheit von Geschäftsprozessen z.B. bei e-Transaktionen.
- Hier: Kunde kauft Ware beim Händler.
- Nach Bezahlung bekommt Kunde Ware ausgeliefert oder kann Bezahlung zurückfordern.

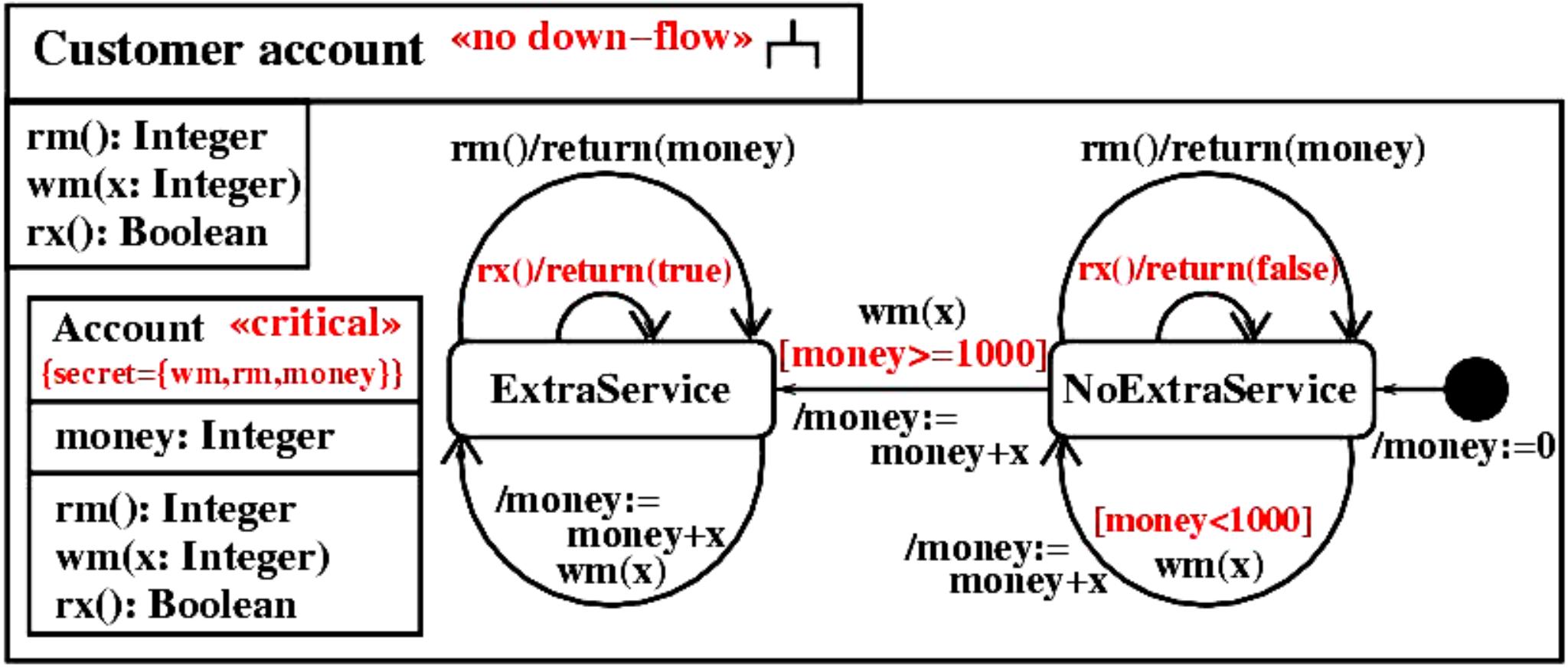


Beispiel: Durchgängige Datensicherheit



Sicherheitslevel definieren. Konsistenzanalyse.

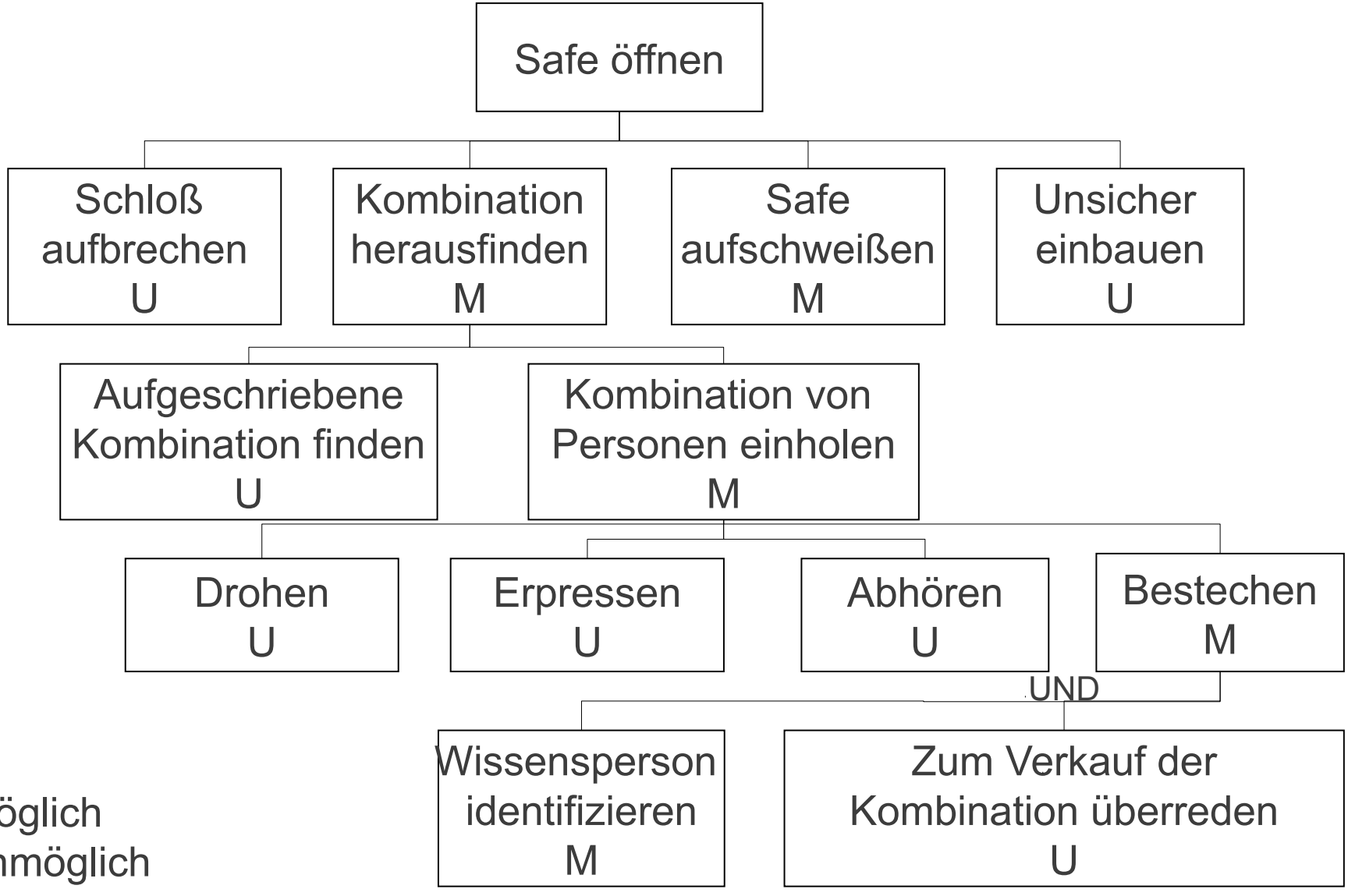
Beispiel: Versteckte Informationsflüsse



Können vertrauliche Daten heraussickern ? Oft ohne Werkzeugunterstützung nicht ersichtlich.

Technik für Sicherheitsanalysen: Angriffsbäume

[Weck 2003]



M = Möglich
U = Unmöglich

Zusammenfassung

- Sicherheit als Problem
- Ursachen
- Modellbasierter Ansatz
- Angriffsbäume