TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Betriebssysteme und ***Sicherheit***

Stefan Köpsell, Thorsten Strufe

Modul 2: Sicherheitsanforderungen

*Disclaimer: Inhalte übernommen aus Materialien von **Winfried Kühnhauser**, Günter Schäfer, Mitarbeitern des Lehrstuhls*

Dresden, WS 15/16

Ziele der IT-Sicherheit

Etwas allgemeiner, die Aufgaben der IT-Sicherheit:

Reduktion operationeller Risiken von IT-Systemen

- Modellierung von System und Umwelt
- Erhebung und Spezifikation von Sicherheitsanforderungen
- Bedrohungsanalysen
- Risiko-Einschätzungen
- Design, Konstruktion und Umsetzung von Schutzmechanismen

Übersicht des Ablaufs

Sicherheitsanforderungen

Bedrohungen
Schwachstellen
Risiken

Sicherheitspolitiken
Modellierung und Spezifikation

ZSLs, HRU, RBAC, ABAC, MLS;
Skippy, XACML, SELinux SPSL

Sicherheitsmechanismen

Authentisierung
Zugriffssteuerung
kryptographische Funktionen

Sicherheitsarchitekturen

TCBs, Referenzmonitore,
Nizza, SELinux, Kerberos

Was ist eine Bedrohung?

Abstrakte Definition:

- Bedrohungen sind mögliche *Ereignisse*, oder Reihungen von Ereignissen und Aktionen, die zu einer *Verletzung eines oder mehrerer Sicherheitsziele* führt
- Eine Instanziierung einer Bedrohung ist ein ***Angriff***

Beispiel:

- Ein Hacker bricht in einen Firmencomputer ein
- Mutwillige Manipulation von Bankdaten
- Sabotage und temporäre Abschalten einer Webseite
- Nutzung von Diensten im Namen einer anderen Partei

Aber was sind Sicherheitsziele?

- Sicherheitsziele werden definiert:
  - In Abhängigkeit von Anwendung und Umwelt, oder
  - genereller und technischer…

Sicherheitsziele in Anwendungsdomänen

Public Telecommunication Providers:

- Schutz der Privatsphäre der Kunden
- Einschränkung des Zugriffs zu administrativen Funktionen
- Sicherung gegen Unterbrechungen

Corporate / Private Networks:

- Schutz der Vertraulichkeit von Firmen-Interna / persönlicher Privatsphäre
- Sicherstellung der Autentizität von Nachrichten
- Sicherung gegen Unterbrechungen

Alle Netzwerke:

- Verhinderung des Eindringens durch außenstehende Hacker

Sicherheitsziele werden auch als ***security objectives*** bezeichnet

Etwas formaler: Ziele der IT Sicherheit

Vertraulichkeit (Confidentiality)

- Übertragene und gespeicherte Daten dürfen nur legitimierten Empfängern zugänglich sein
- Vertraulichkeit der Identität wird als Anonymität bezeichnet

Integrität (Integrity)

- Veränderungen an Daten müssen detektiert werden
- (Bedarf der Identifikation des Absenders!)

Verfügbarkeit (Availability)

- Informationen und Dienste sollen berechtigten Nutzern in angemessener Frist zugänglich sein

Zurechenbarkeit (Accountability)

- Die verantwortliche Partei für eine Operation soll identifizierbar sein

Kontrollierter Zugriff (Controlled Access)

- Nur authorisierte Parteien sollen in der Lage sein, auf Dienste oder Informationen zuzugreifen

Terminologie: Bedrohungen

Maskerade

- Instanz gibt vor die Identität einer anderen Instanz zu haben

Informationsverlust (Abhören, Ausspähen)

- Instanz liest Information, die nicht für sie bestimmt ist

Authorisierungsverletzung

- Instanz nutzt Ressourcen ohne dazu autorisiert zu sein

Zerstörung/Modifikation von Information

- Information wird zerstört oder verändert

Fälschung von Information

- Instanz erzeugt Information in der Identität einer anderen Instanz

Abstreiten von Ereignissen

- Instanz leugnet fälschlicherweise, an Ereignis beteiligt gewesen zu sein

Sabotage

- Mutwillige/geplante (Zer-)Störung von Diensten oder Systemen

Terminologie: Sicherheits-Dienste

Authentifizierung (Authentication)

- Nachweis behaupteter Eigenschaft (Identität) einer Instanz

Integritätsschutz (Data Integrity)

- Nachweis der Unversehrtheit von Information

Vertraulichkeitsschutz (Confidentiality)

- Verhinderung unauthorisierten Zugriffs zu Information

Zugangskontrolle (Access Control)

- Überwachung der legitimierten Zugriffsweise auf Ressourcen

Nicht-Abstreitbarkeit (Non repudiation)

- Nachweis der Teilnahme einer Instanz an einem Ereignis

Sicherheitsanforderungen

Ziel

- Methodische
  - Identifikation
  - Spezifikation
- der Sicherheitseigenschaften von IT-Systemen

Globale Einflussfaktoren

Gesetze

- Bundesdatenschutzgesetz (BDSG), US Sarbanes-Oxley Act (SarbOx)

Verträge

- mit Kunden

Zertifizierung

- für Informationssicherheitsmanagementsysteme (ISO 27001)
- nach dem deutschen Signaturgesetz
- nach den *Common Criteria*

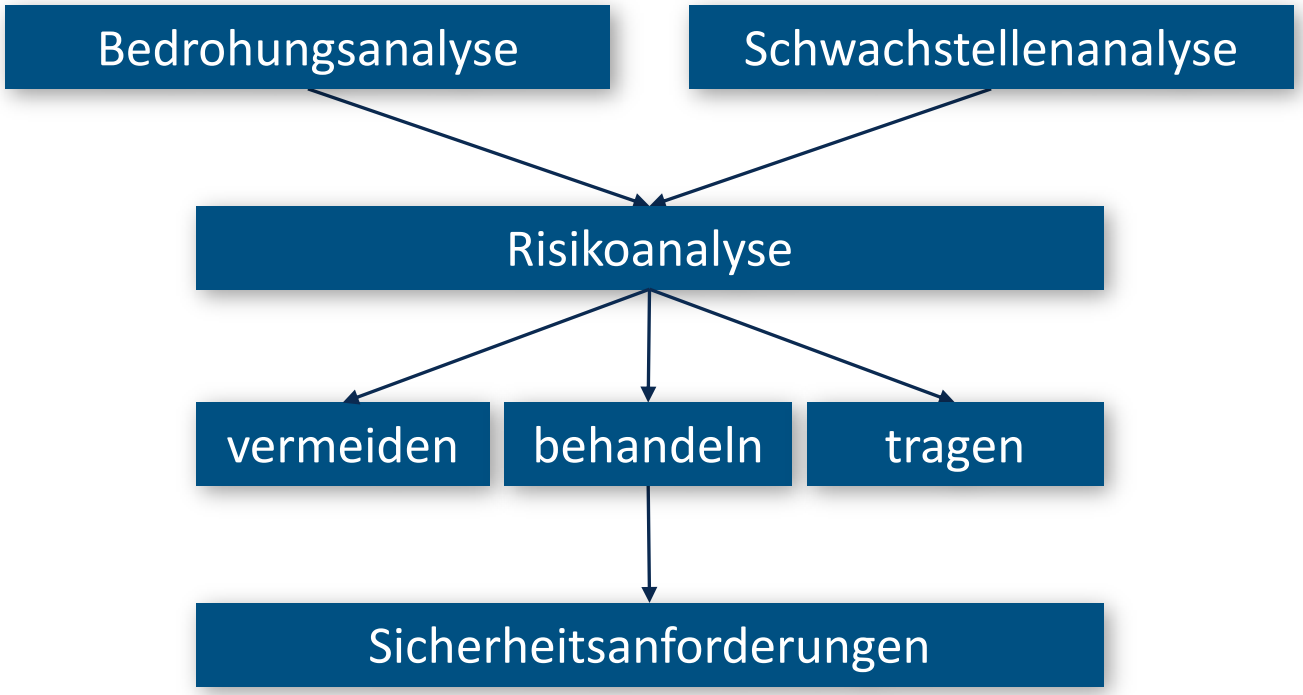
interne Richtlinien und Vorgaben

- Umgang mit externen Geräten, Passwortkonstruktion, Berechtigungsvergabe

informationstechnische Faktoren

- Systemarchitektur, Anwendungssysteme

Methode der Identifikation von Sicherheitsanforderungen



Bedrohungsanalyse

Ziel

- Identifikation der möglichen
- Angriffsziele und Angreifer
- Angriffsmethoden und -techniken

Weg

- Erstellung eines Bedrohungskatalogs; Inhalt:
  - Identifikation der Angriffsziele
  - Identifikation potentieller Angreifer
  - Angriffsmethoden und -techniken
  - Schadenspotential

Bedrohungsanalyse: Gesichtspunkte

Angriffsziele

- Informationsgewinn (Wirtschaftsspionage, Kontrolle kritischen Wissens)
- Modifikation von Daten (Sabotage)

Angreifer

- professionelle Organisationen
- ehemalige und aktive Mitarbeiter
- politische Gegner

Angriffsmethoden und -techniken

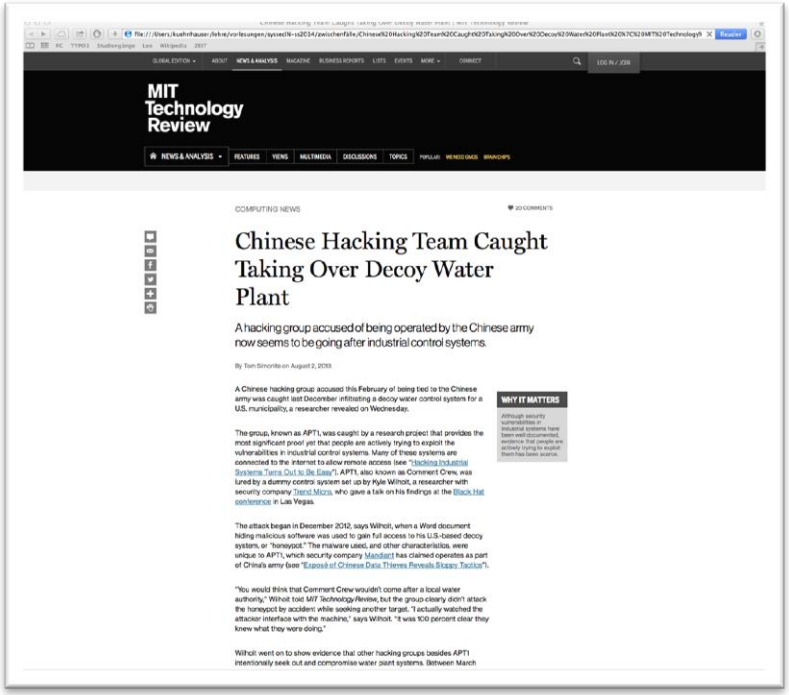
- Ausnutzung technischer und menschlicher Schwachstellen

Schadenspotential

- Verlust der Kontrolle über kritisches Wissen (Risikotechnologien)
- wirtschaftliche Schäden (Vertragsstrafen, Produktkopien)
- Reputationsschäden

Angriffsziele und Angreifer

Die Hitlisten:



Angriffsziele

- wirtschaftliche und politische Macht
- finanzieller Gewinn
- Schaden anrichten
- Herausforderungen meistern

Angreifer

- professionelle Organisationen (bezahlt von Konkurrenzunternehmen, fremden Staaten)
- aktive und ehemalige Mitarbeiter
- Terroristen
- Hacker

Bedrohungspotenziale – Angreifermodell

Potentielle Angreifer – Vor wem ist zu schützen?

Grundlegend zunächst: Betrachtung von Nutzern des Systems und Außenstehenden

- Dienstanbieter
- Berechtigte Nutzer
- Unberechtigte Nutzer
- Wartungsdienst
- …

Ebenfalls zu beachten: Beteiligte am Entwicklungsprozess

- Produzenten des Systems
- Designer des Systems
- Produzenten der Entwurfs- und Produktionshilfsmittel
- …

… ergibt auch eine Betrachtung des Einflusses weiterer IT-Systeme

Bedrohungspotenziale – Angreifermodell

Generell: Kein Schutz vor einem allmächtigen Angreifer!

Ein allmächtiger Angreifer …

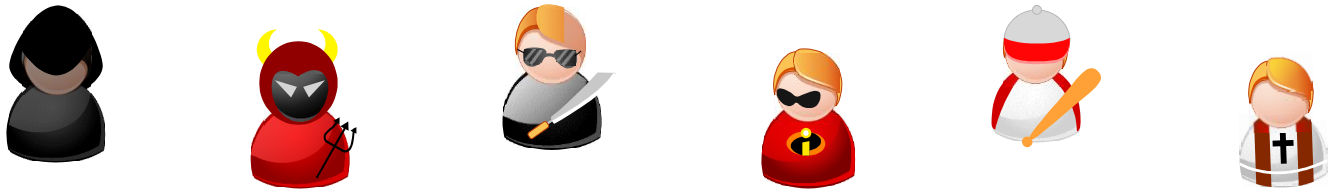
kann alle ihn interessierenden Daten erfassen

kann Daten unbemerkt ändern

kann die Verfügbarkeit des Systems durch physische Zerstörung beeinträchtigen

→ Angreifermodell

Angabe der maximal berücksichtigten Stärke eines Angreifers, d.h., Stärke des Angreifers, gegen die ein bestimmter Schutzmechanismus gerade noch sicher ist



Bedrohungspotenziale – Angreifermodell

Inhalt des Angreifermodells

- Rollen des Angreifers
  (Nutzer, Außenstehender, …)
- Verbreitung des Angreifers
  (kontrollierte Subsysteme, Leitungen, …)
- Verhalten des Angreifers
  (passiv/aktiv, beobachtend/verändernd)
- Rechenkapazität
  (komplexitätstheoretisch (un)beschränkt)
- Verfügbare Mittel
  (Zeit, Geld)

Angriffsmethoden

Ausnutzung technischer Schwachstellen

- Schwachstellen in Systemen
  - konzeptionelle Schwächen
  - Spezifikations- und Implementierungsfehler
- Schwachstellen in Kommunikationsnetzen

  → Netzsicherheit (Vert-4), Kryptographie (Vert-4)

- Ausnutzung nichttechnischer (menschlicher) Schwachstellen
  - *social engineering*
    - der Chef, der Freund, die vergiftete Tochter
  - *phishing*

Realistisches Angreifermodell

Berücksichtigung aller Angreifer

- Momentan erwartete und während der Lebenszeit des Systems zu erwartenden Angreifer berücksichtigen.

Einfach

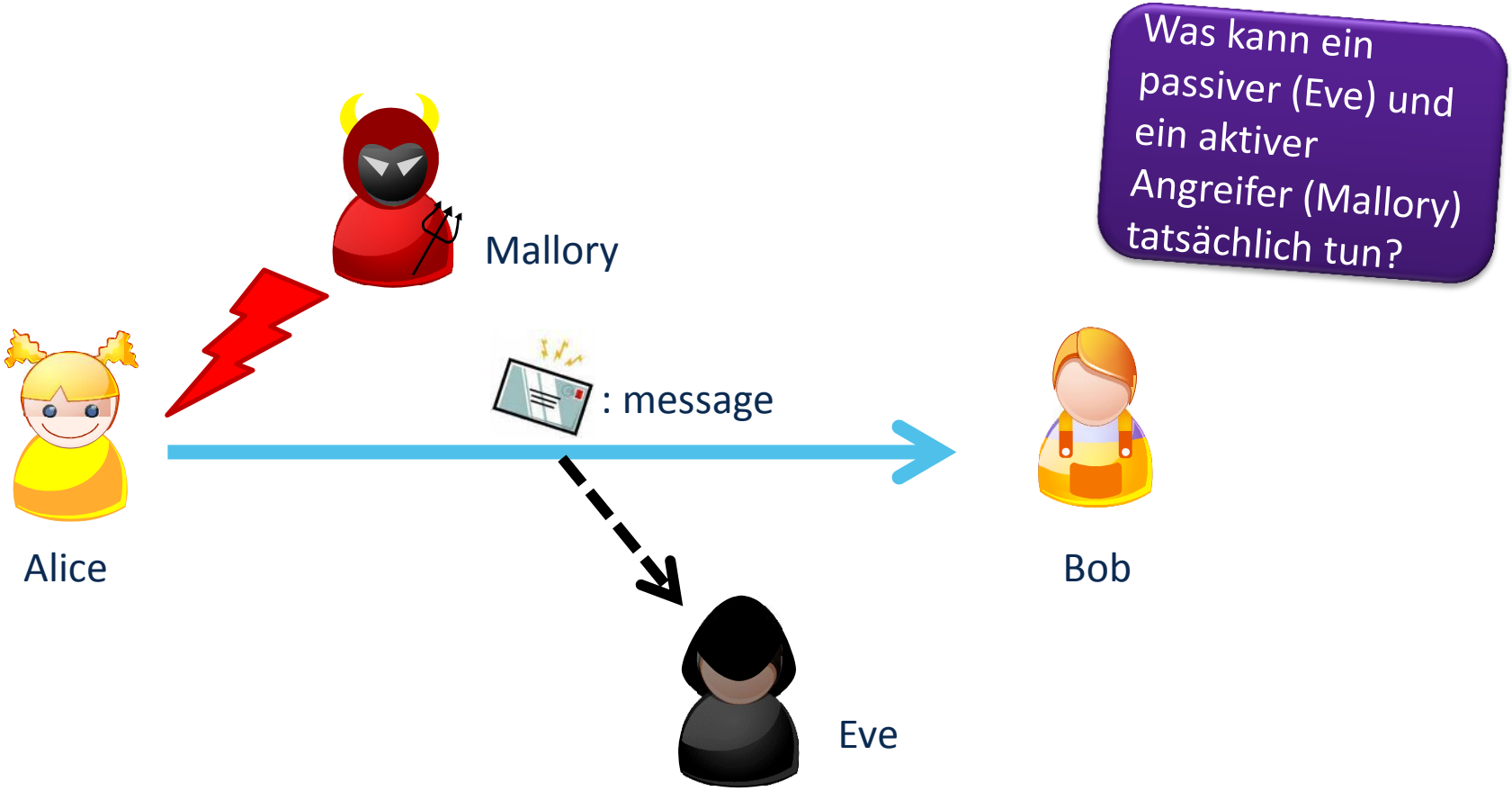
- Restrisiken durch nicht abgedeckte Angreifer müssen verständlich sein.

Realisierbar

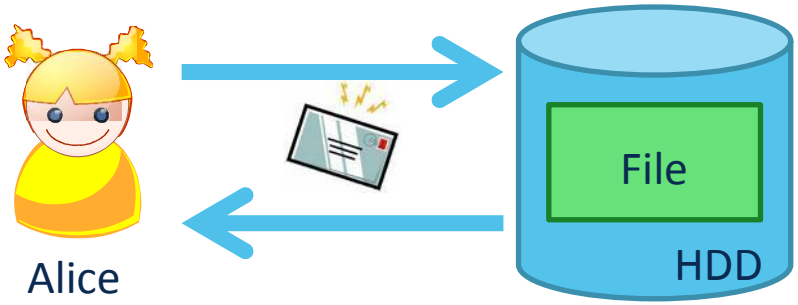
- Gegen aufgestelltes Angreifermodell sicheres System muss mit vertretbarem Aufwand realisiert und betrieben werden können.

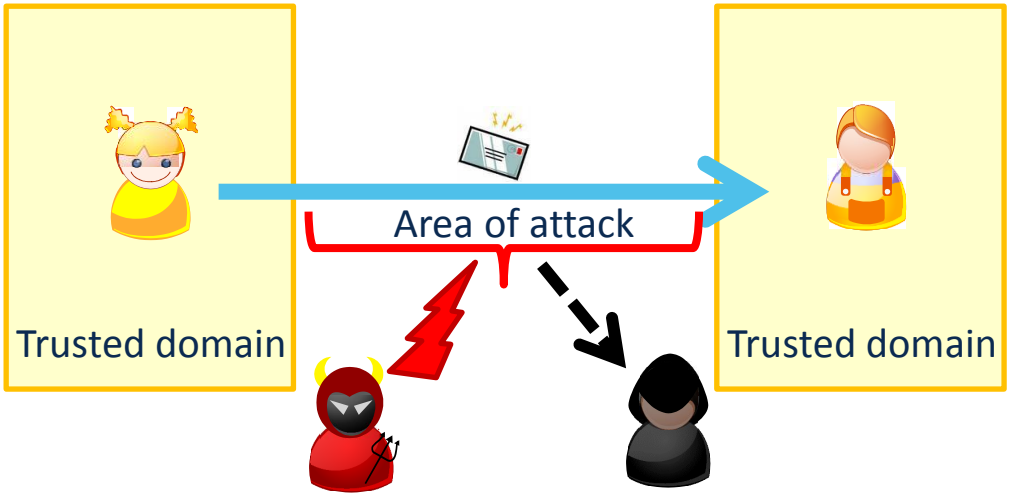
Einige Akteure des Schauspiels

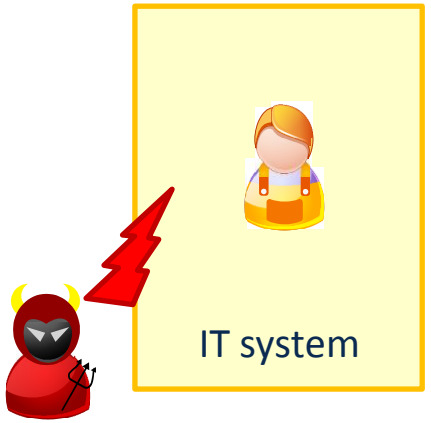
Es ist gut, ein Modell zu haben…



Variationen des Stücks







Szenario: Wirtschaftsspionage

- ***Ziel:*** finanzieller Gewinn, wirtschaftliche und politische Macht
- ***betroffen:*** High-Tech Industrie, KMU***(!)***

- ***Angreifer***
  - Konkurrenzunternehmen, Staaten, Mitarbeiter
  - >40% (direkt) verursacht durch Insider
    - oft indirekt („*Only amateurs attack systems; professionals target people*“)
    - Alter 30-40 Jahre, männlich
    - Abteilungsleiter, Systemadministrator, Programmierer

    → regulärer, oft privilegierter Zugang zu IT-Systemen, Insiderkenntnisse

  - Außenstehende: professionelle Organisationen

Szenario: Bereicherung/Zerstörung

- ***Ziel***: finanzieller Gewinn (kostspieliger Lebensstil, Krankheit)

- ***Angreifer***
  - Konkurrenzunternehmen
  - Mitarbeiter
    - Alter 40-50 Jahre, Karrierehöhepunkt erreicht, *midlife crisis*
    - Männlich

    → regulärer Zugang zu IT-Systemen, Insiderkenntnisse

- ***Szenario***: Zerstörung
- ***Ziel***: Schädigen, Erpressung, Herausforderungen meistern
- ***Angreifer***: Terroristen, Rächer, Psychos, Hacker

  → kein regulärer Zugang zu IT-Systemen, keine Insiderkenntnisse

Beispielszenarien

1. Szenario: Insiderangriff

- Angriffsmethode: Ausnutzung konzeptioneller Schwachstellen

  → Spezifikationsfehler bei der Verwendung von Sicherheitsmechanismen

  → verdeckte Informationsflüsse

Ein alltägliches Szenario

F&E

Anna

Bernd

ProjectX
Bulletin
Board

Bernd‘s
Notes
To Sales

Vertrieb
(„Sales“)

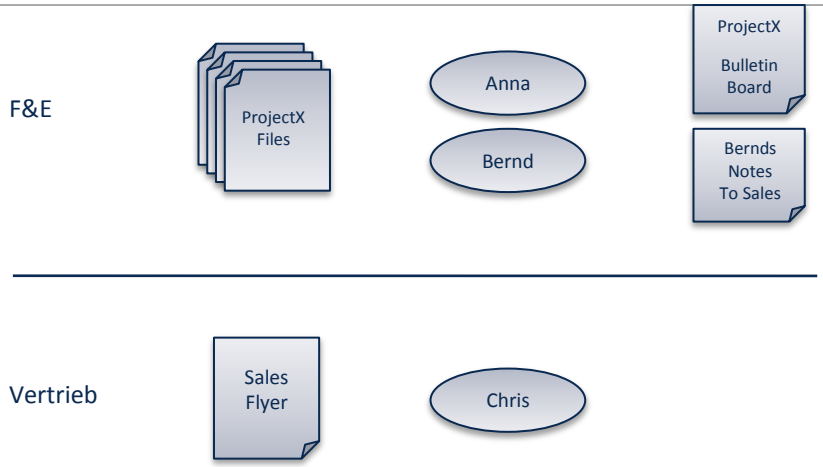
Chris

3 Benutzer: Anna, Bernd, Chris

2 Gruppen:

CrewX: Anna (PL), Bernd

Sales: Bernd, Chris



Kontext: Linux-Zugriffssteuerungssystem

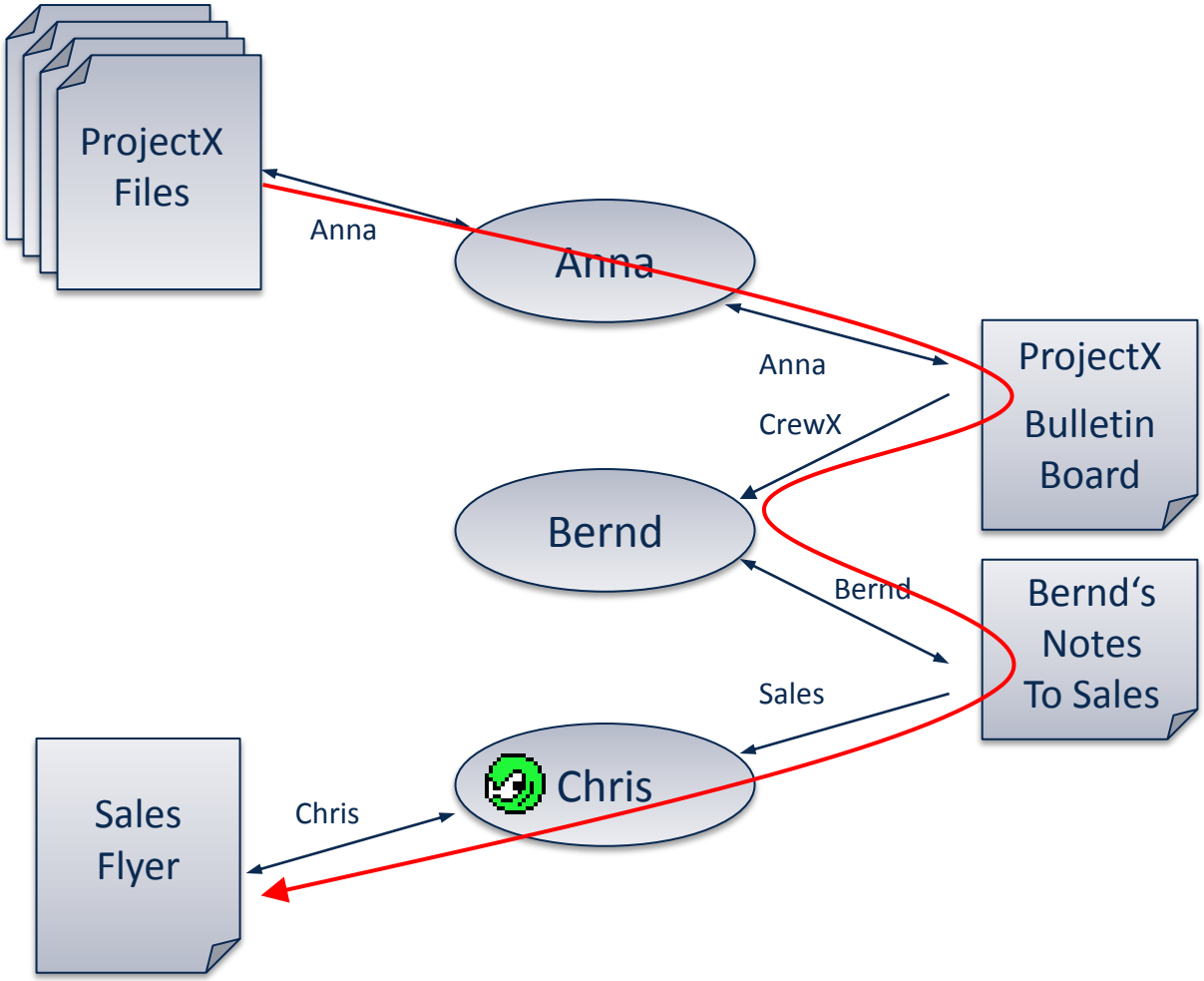
```
rw- --- ---   1 Anna    CrewX   2012-04-23   17:10 ProjectXFiles

rw- r-- ---   1 Anna    CrewX   2012-04-23   17:10 ProjectXBoard

rw- r-- ---   1 Bernd   Sales   2012-04-23   17:10 BerndsNotesToSales
rw- --- ---   1 Chris   Sales   2012-04-23   17:10 SalesFlyer
```

Fazit

- alle 3 Benutzer haben Rechte ihrer Dateien - aus ihrer Sicht - perfekt vergeben
- gemeinsam haben sie eine Zeitbombe gebaut


ProjectX Files
Anna
Anna
Anna
CrewX
ProjectX Bulletin Board
Bernd
Bernd
Bernd's Notes To Sales
Sales
Chris
Sales Flyer
Chris

Fazit: Wirksamkeit wahlfreier Zugriffsschutzsysteme (DAC)?

Problemursachen

- Komplexität des Problems
  - Auswirkungen individueller Rechtevergabe

- mangelndes Wissen der Akteure
  - begrenzter Horizont
  - begrenztes Problembewusstsein
  - begrenzte Fähigkeiten

- begrenzte Möglichkeiten der Akteure
  veraltete und ungeeignete Sicherheitsmechanismen
  - fehlende Isolation nicht vertrauenswürdiger Software
  - fehlende Durchsetzung globaler Sicherheitspolitiken

Konzeptionelle Technikschwächen

- Abstraktionsniveau der Mechanismen
  - → Ausdrückbarkeit von Informationsfluss, Rollen?
- wahlfreie Zugriffssteuerung
  - → globale Verantwortung bei begrenztem Horizont
- Intransparenz der Mechanismen
  - → Erkennbarkeit von Auswirkungen (Gruppenrechte!)

- Konzeptionelle Usability-Schwächen
- Intransparenz von Handlungsfolgen
  - begrenzter Horizont der Akteure
  - begrenzte Kompetenz der Akteure
- globale Auswirkungen von Handlungen

Szenario: In- und Outsiderangriff

- ***Angriffsmethode***: Ausnutzung von Implementierungsfehlern in privilegierter Systemsoftware
  - Betriebssystem, lokale Dämonenprozesse (Insider)
  - *ssh-*, *ftp*-Dämonen, Webserver etc. (Outsider)

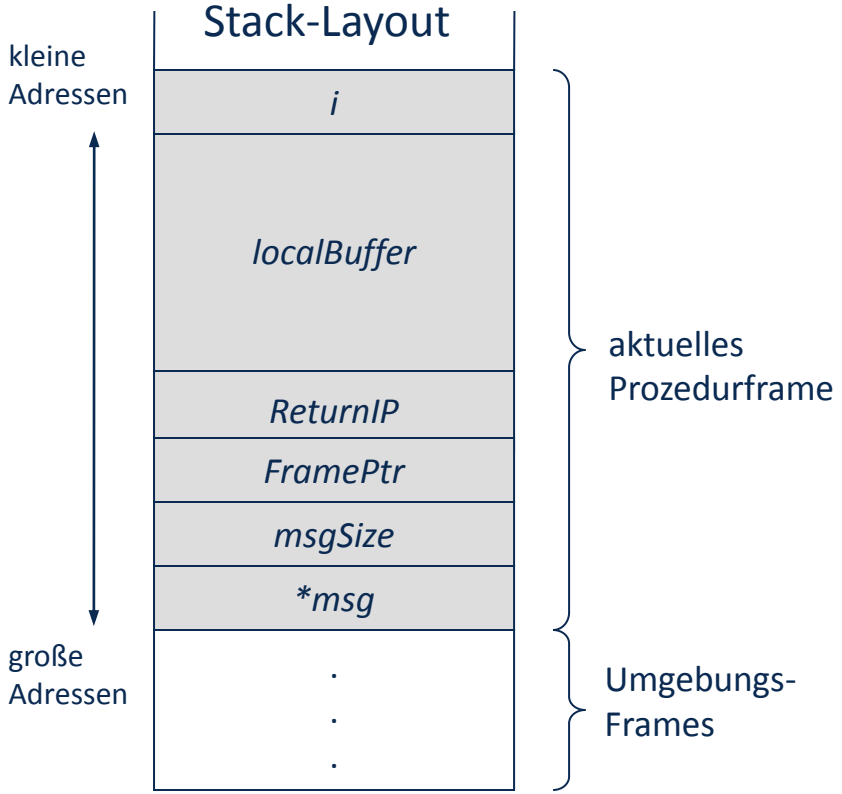
- ***Ziel***: privilegierte Software zur Ausführung eigenen Codes zu bringen

- ***Technik***: durch geschickt geschmiedete Parameter Stackinhalt überschreiben; *„Buffer Overflow“*-Angriff

- notwendiges Wissen:
  - Quellcode des Servers
  - etwas Compilerbautechnik (Prozeduraufrufmanagement)

Prinzip

```
void processSomeMsg(char *msg, int msgSize);
{   char localBuffer[1024];
    int i=0;
    while (i<msgSize) {
        localBuffer[i] = msg[i];
        i++;
    }
    . . .
}
```

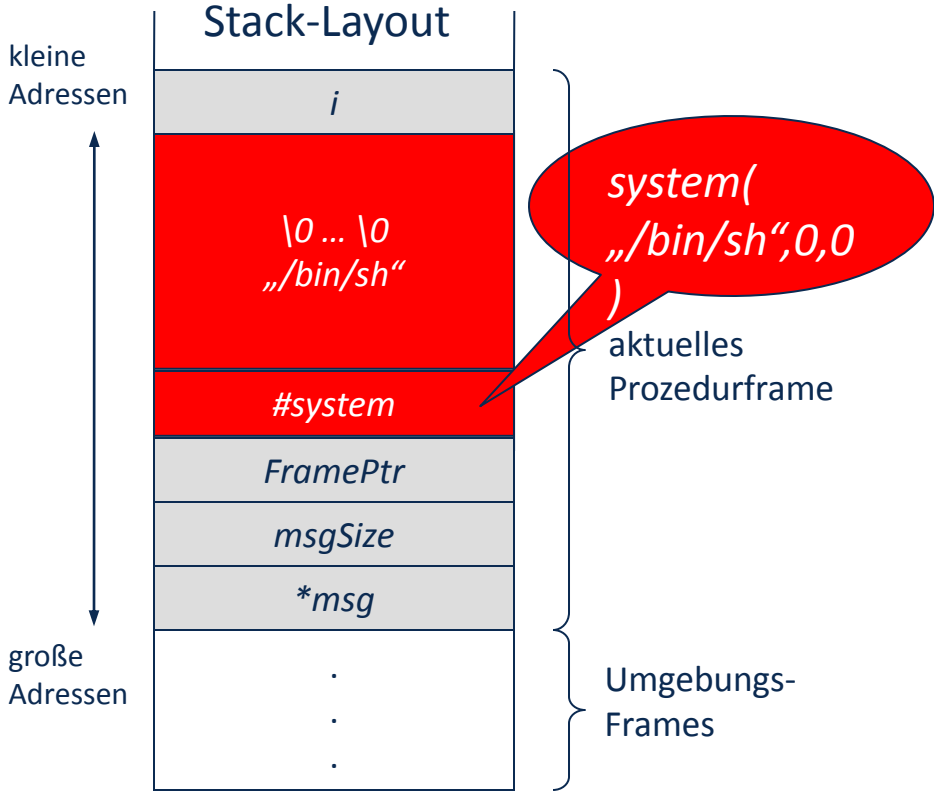


Prinzip

msg: „\0 … \0 / b i n / s h e l l *#system*“

1024 Zeichen

```
void processSomeMsg(char *msg, int msgSize);
{   char localBuffer[1024];
    int i=0;
    while (i<msgSize) {
        localBuffer[i] = msg[i];
        i++;
    }
    . . .
}
```



Schadensprogramme; eine Familienchronik

- Schadsoftware
  - Programme, die neben bekannter Funktion weitere, verborgene besitzen:
  1. ***Viren-/Verbreitungsfunktion***
     - Codesequenzen die eine Modifikations- und Vervielfältigungsfunktion enthalten, oft auch eine Schadensfunktion
     - Anwendung von ***Exploits*** auf Schwachstellen in Software
  2. ***Hintertüren (backdoors)***
     - Codesequenzen deren Aktivierung an den Aufruf einer verborgenen Funktionen gebunden ist (→ login, ssh)
  3. ***Logische Bomben / APT***
     - Codesequenzen deren Aktivierung an konkrete Ereignisse gebunden ist (→ Michelangelo, Chevron)

- Arten von Malware
  - Viren/Würmer
  - Trojanische Pferde
  - Packer
  - Risk-/Scareware

Professionelle Angriffstechniken: *Root Kits*

- ***Ziel***: vollständige, unsichtbare, nachhaltige Kontrolle über System
- ***Mittel***: Werkzeugkasten vollautomatisiert ablaufender Angriffe

Im Werkzeugkasten: Werkzeuge für

- vollautomatische Analyse technischer Schwachstellen
- vollautomatisches Angriffsmanagement
- vollautomatische Installation von Hintertüren
- vollautomatisches Tarnsystem

Angriffsablauf

- Erster Schritt: Schwachstellenanalyse
- Werkzeuge suchen Schwachstellen in
  - aktiven privilegierten Diensten und Dämonen (von außen: *port scans*)
    - Webserver, Remote Zugang (*sshd*), File server (*ftp*), Zeitserver (*ntpd*), *cupsd*, *bluetoothd*, *smbd*, ...
  - Konfigurationsdateien
    - schwache Passworte, offene Kommunikationsports
  - Betriebssystemen
    - bekannte Implementierungsfehler bestimmter Hersteller/BS-Versionen

- Wissensbasis:
  - umfangreiche Schwachstellendatenbank

Resultat

- Schwachstellensammlung

→ Angriffsmethode und zugehörige Werkzeuge → zweiter Schritt

Zweiter Schritt: Angriffsmanagement

- Fabrikation einer Reihe spezialisierter Software-Bausteine; damit
  - Nutzung der Schwachstelle so, dass
    - Server oder Dämonenprozesse
    - BS

  … Code des Angreifers mit Root-Privilegien ausführt.

- Dieser Code
  - installiert Nebelbomben zum Verstecken des Angriffs
  - tauscht Originale gegen fabrizierte Softwarekomponenten aus
    - Server oder Dämonenprozesse
    - Dienstleistungsprogramme und Bibliotheken
    - BS-Module

    mit
    - Hintertüren
    - Nebelbomben für zukünftige Angriffe

Angriffsergebnis

Resultate

- hochprivilegierter Zugang zum System innerhalb von Bruchteilen von Sekunden
- System modifiziert mit Servern, Dämonen, Utilities, BS-Modulen des Angreifers
- Nebelbomben zum Verbergen dieser Modifikationen

Nebelbomben für laufende Angriffe

Reinigung von Logfiles (Einträge über Root Kit Prozesse, Verbindungen)
- *syslog, kern.log, user.log, daemon.log, auth.log, ...*

modifizierte Utilities zum Systemmanagement
- Prozessmanagement (Verbergen laufender Root Kit Prozesse)
  - Unix: z.B. *ps, top, ksysguard*; Windows: *task manager*
- Dateisystem (Verbergen von Root Kit Dateien)
  - *ls, explorer, finder*
- Netzwerk (Verbergen aktiver Root Kit Verbindungen)
  - *netstat, ifconfig, ipconfig, iwconfig*

Austausch von BS-Modulen (Verbergen laufender Root Kit Prozesse, Dateien, Verbindungen)
- Unix: */proc/...*, *stat*, *fstat*, *pstat*

Resultat
- Prozesse, Kommunikation, Software des Root Kits werden unsichtbar

Nachhaltigkeit von Angriffen

Hintertüren für zukünftige Besuche

- in Servern (*ssh*)
- in Utilities (*login*)
- in Bibliotheken (*PAM, pluggable authentication modules*)
- im BS (in von Programmen wie *sudo* benutzten Systemaufrufen)

Modifikationen von Utilities und BS zur Verhinderung des

- Abbrechens von Root Kit Prozessen und Kommunikationsverbindungen (*kill, signal*)
- Entfernens von Root Kit Dateien (*rm, unlink*)

weitere Nebelbomben zur Tarnung von

- Server-, Dämonen-, Bibliotheks-, Utility- und BS-Modifikationen

Resultate aus Angriffen durch Rootkits

Verborgener Zugriff auf angegriffenes System

- jederzeit
- unentdeckbar
- hoch privilegiert
- extrem schnell
- nahezu nicht zu verhindern

Messages: Risiko- und Schadenspotential

Erfolgsaussichten: extrem hoch in heutigen Standardsystemen

- gewaltiges Schwachstellenpotenzial
- Geschwindigkeit
- Methodik
- Automatisierung

Möglichkeiten der Abwehr: extrem gering

- gewaltiges Schwachstellenpotenzial
  (Zahl der „Sicherheitsupdates“ im letzten Jahr ...)
- Ursache der Schwachstellen
  (Spezifikations- und Implementierungsfehler)
- Geschwindigkeit
- Nebelbomben

Möglichkeiten der Rettung eines Systems nach einem Angriff: nahe Null

- Nebelbomben, BS-Modifikationen

Zunehmende Professionalisierung

1. Übernahme Rechner mittels Schad-software
2. Eingliederung in Bot-Betz
3. Vermarktung Bot-Netz
4. Nutzung Bot-Netz für illegale Aktivitäten (Hacking, Betrug, SPAM etc.)

[Tom-b, http://de.wikipedia.org/wiki/Datei:Botnet.svg]

Möglichkeiten der Vorbeugung

reaktiv

- hmm…

präventiv

- dieselben Werkzeuge nutzen: Schwachstellenanalyse und Beseitigung
  (das machen wir seit Jahren)
- korrekte Software schreiben
  (das versuchen wir seit Jahren)

→ Security Engineering

- neue Paradigmen
  → politikgesteuerte Systeme
- Sicherheitsmodelle
  → Reduktion von Spezifikationsfehlern
- Sicherheitsarchitekturen und minimale TCBs
  → Reduktion von Implementierungsfehlern

Ablauf einer Bedrohungsanalyse

Problem

- Vielfalt (Diversität der Einflussfaktoren)
- Komplexität (Anzahl der Einflussfaktoren)
- mangelnde Erfahrung

Praktische Hilfen: Profilekataloge

- national: IT-Grundschutzkataloge des BSI
- international: die Common Criteria

IT-Grundschutzkataloge

- Szenarien spezifische
  Gefährdungskataloge
- Beispielprofile
- Maßnahmenkataloge

Common Criteria for Information Technology Security Evaluation

Internationaler Standard für Sicherheitszertifizierungen von IT-Systemen

πάντα ῥεῖ

Junges Fachgebiet

→ Zahlreiche wissenschaftliche Publikationen (*risk engineering*, *role engineering*, ...)

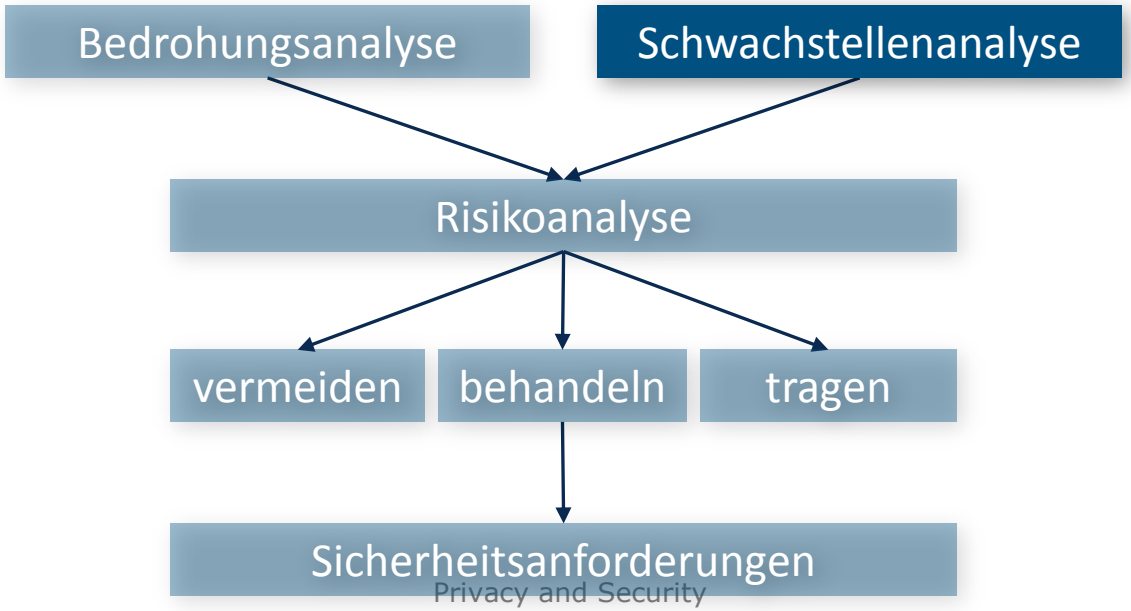
Schwachstellenanalyse

Ziel

- Identifikation
- technischer
- organisatorischer
- menschlicher

Verwundbarkeiten eines IT-Systems

- Produktionsanlagen: Manipulation der Frequenzumrichter (Iran)
- Energie/Wasserversorgung: Stadtwerke Ettlingen (Blackout; Arte Doku „Netwars“)
- intelligente Energienetze (Blackout)
- Cloud Computing: geschäfts-kritische Daten



Technische Schwachstellen

Komplexe IT-Systeme können in absehbarer Zeit nicht

- vollständig, widerspruchsfrei und korrekt spezifiziert sein

  → enthalten Spezifikationsfehler

- korrekt implementiert sein

  → enthalten Implementierungsfehler

- jeden Tag neu gebaut werden (viele Schutzmechanismen heutiger IT-Systeme sind > 40 Jahre alt)

  → enthalten konzeptionelle Schwächen

→ Sammlung: Schwachstellenkataloge des BSI

Organisatorische Schwachstellen

Beispiele

- Vergabe von Berechtigungen
  - → Konfiguration von Zugriffssteuerungssystemen
  - → Definition von Rollen
- Management kryptografischer Schlüssel
  - → Erstellung von Zertifikaten

→ Sammlung: Schwachstellenkataloge des BSI

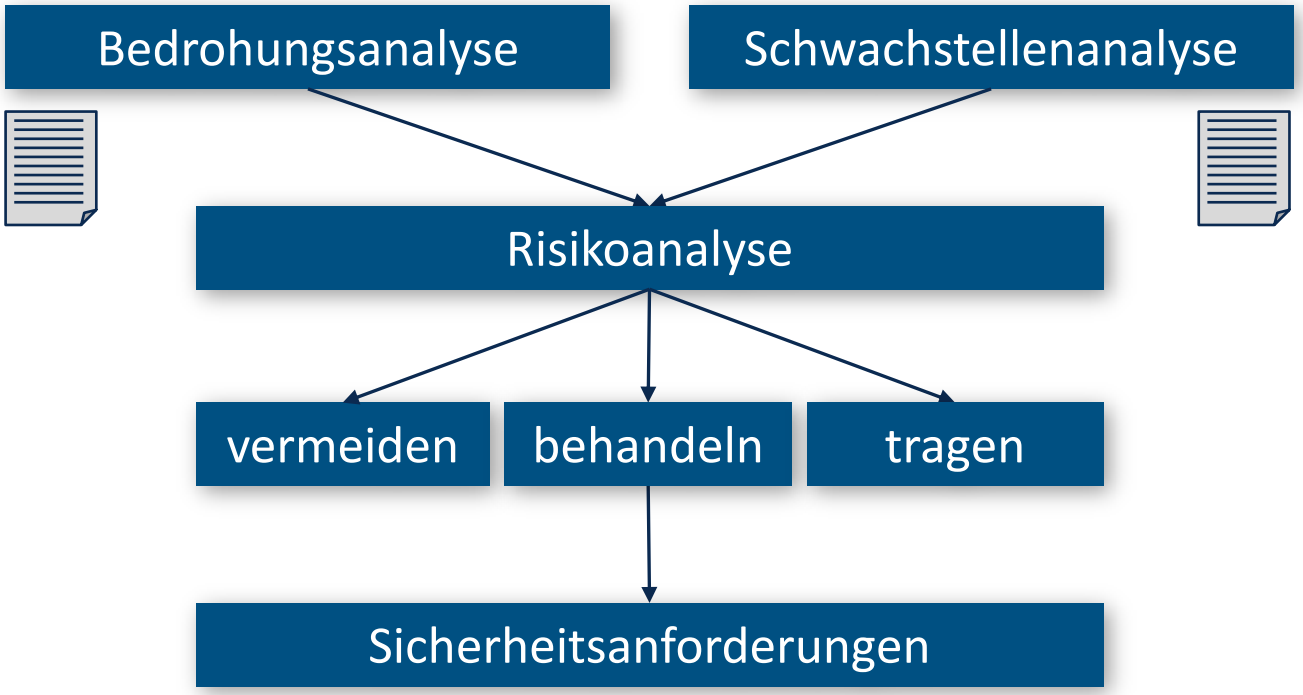
Menschliche Schwachstellen

Beispiele

- Bequemlichkeit (Windows-XP-Boxen)
- mangelndes Problembewusstsein (DAC)
- mangelndes Wissen (DAC)
- Dummheit
- *social engineering*

→ Sammlung: Schwachstellenkataloge des BSI

Zusammenfassung



Risikoanalyse

Ziel

- Identifikation und
- Klassifikation

der tatsächlichen Risiken

Weg

- Korrelation von Bedrohungen und Schwachstellen
  → Risiken
- Klassifikation der Risiken
  → Schadenshöhe und Eintrittswahrscheinlichkeit
  → Risikomatrix

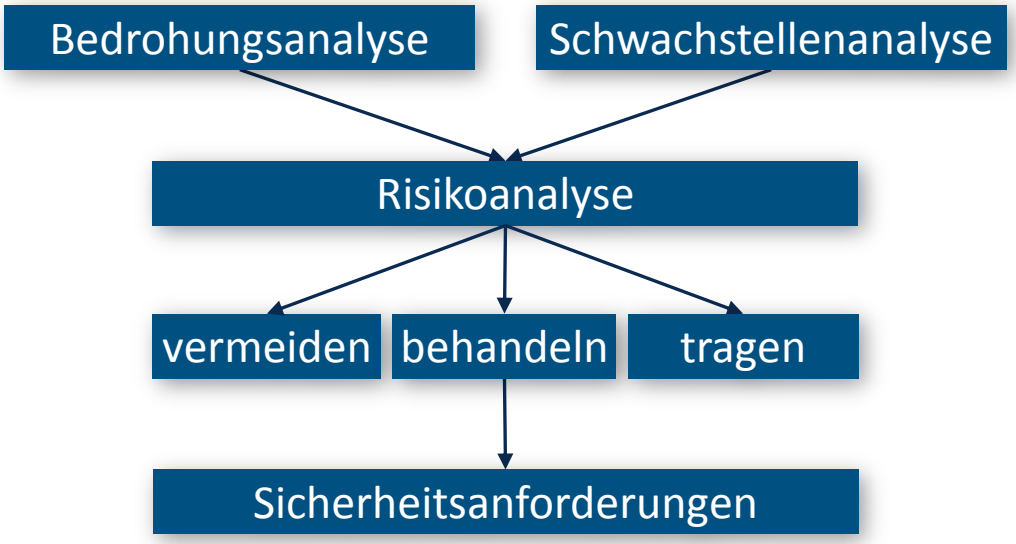
Klassifikation der Risiken

Ziel

∀ Risiken: Bestimmung der Risikoklasse

Risikoklasse bestimmt Reaktion auf Risiko; mindestens:

- vermeiden
- behandeln
- tragen



Methode

Schutzbedarfskategorien abhängig vom Schadenspotenzial

Beispiel (hier mit 3 Klassen)

Schutzbedarfskategorien	
„Normal“	planbare und begrenzte Schadensauswirkungen
„Hoch“	beträchtliche Schadensauswirkungen
„Sehr Hoch“	existentiell bedrohliche Schadensauswirkungen mit katastrophalem Ausmaß

- Anlagensteuerung: Manipulation der Frequenzumrichter?
- Versorgungssicherheit: Energieversorgung?
- Cloud Computing: Isolation der VMs?

nach BSI IT-Grundschutz

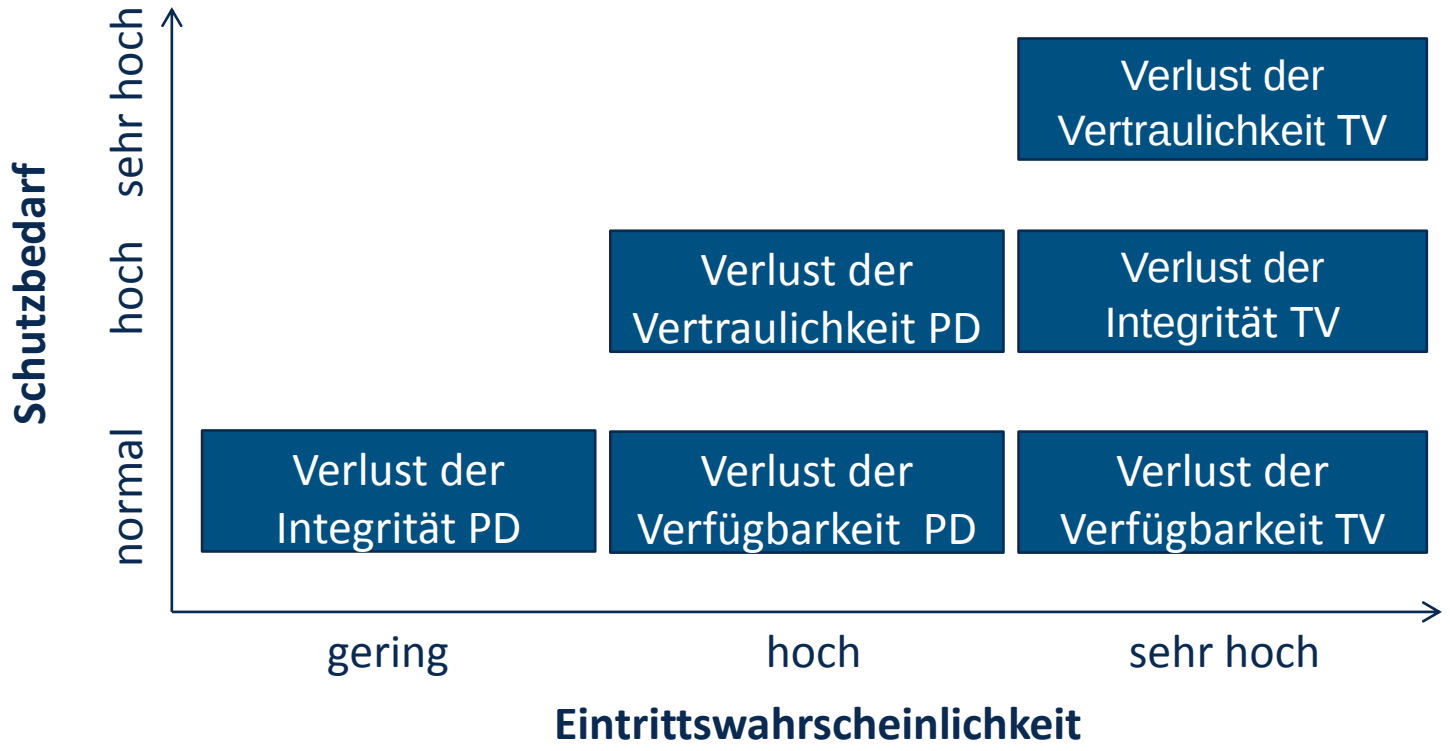
Schutzbedarfsfeststellung

Objekte	Sicherheitsziel	Schutzbedarf	Begründung
Personenbezogene Daten (PD)	Vertraulichkeit	hoch	Bei Bekanntwerden können Daten Betreffenden erheblich beeinträchtigen
	Integrität	normal	Fehler können rasch erkannt und Daten korrigiert werden; jedoch: z.B. Schufa ...
	Verfügbarkeit	normal	Ausfälle bis zu einer Woche können evtl. mit manuellen Verfahren überbrückt werden.
Technische Verfahren (TV)	Vertraulichkeit Integrität Verfügbarkeit	sehr hoch hoch normal	Marktführerschaft weg Stillstand Produktion Backups

Risikomatrix

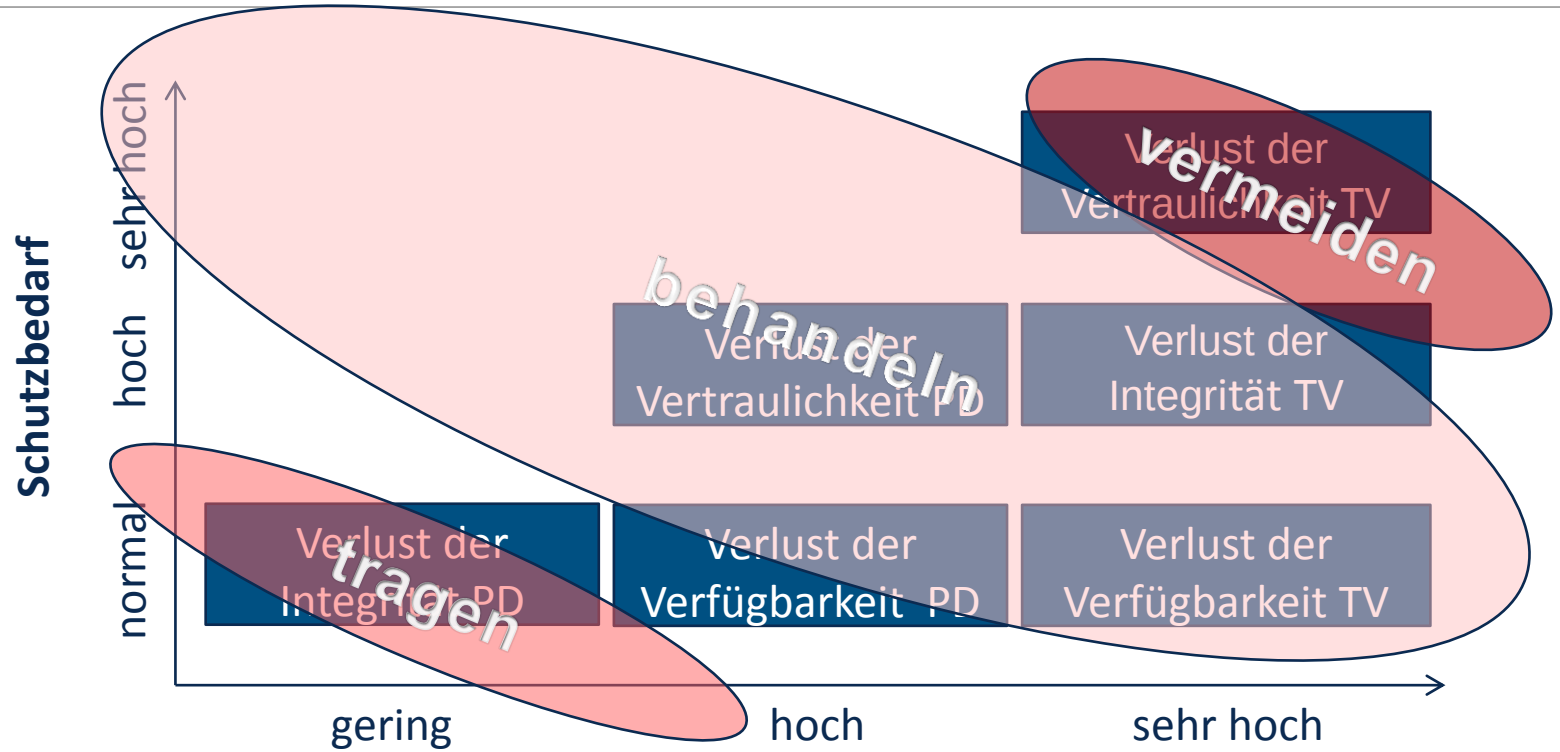
Kombiniert

- Schutzbedarf
- Eintrittswahrscheinlichkeit (geschätzt, statistisch belegt)



nach BSI IT-Grundschutz

Entscheidung des Managements: Risikoklassifikation



Zusätzliche Kriterien

- Schadensausmaß und Konsequenzen
- Personal- und Sachkosten
- organisatorische und technische Machbarkeit

→Kosten/Nutzen-Relation; Management, Geschäftsbereiche involviert

vermeiden

- untragbares Risiko, keine Verhältnismäßigkeit zw. Kosten/Nutzen
- Funktionalität weglassen

tragen

- Akzeptieren der Risiken
- Verringern des Schadenausmaßes z.B. durch Versicherung

behandeln

- Definition von Sicherheitsanforderungen
- Reduzierung der Schadenswahrscheinlichkeit durch systemintegrierte Sicherheitspolitiken

nach BSI IT-Grundschutz

Bedrohungsanalyse, Übersicht