Embedded & Software Engineering

MIXED MODE

technik.mensch.leidenschaft

IoT – Security Check

Michael Schnelle, 21.04.2018

DIN EN ISO 9001

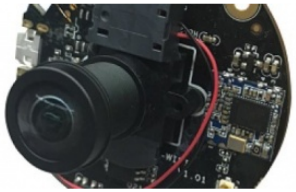
QZV

Wozu IT-Sicherheit?

DDOS FÜR 7.500 US-DOLLAR

Hacker verkaufen Zugang zu IoT-Botnetz im Darknet

Der Zugang zum IoT-Botnetz Mirai setzt neuerdings keine technischen Kenntnisse mehr voraus, sondern nur genügend Finanzmittel - 7.500 US-Dollar. Ein chinesischer Hersteller sagt *"Mirai ist ein Desaster für das IoT"* und reagiert mit einer Rückrufaktion.



Ein chinesischer Hersteller räumt ein, dass seine Produkte für Mirai genutzt werden. (Bild: Hangzhou Xiongmai Technology)

ANDY GREENBERG SECURITY 07.21.15 06:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

98 Sekunden bis zur Infektion: IoT-Botnetz im Selbstversuch

UPDATE

22.11.2016 13:16 Uhr – Dennis Schirrmacher vorlesen

MQTT-Protokoll: IoT-Kommunikation von Reaktoren und Gefängnissen öffentlich einsehbar

17.02.2017 09:44 Uhr – Uli Ries vorlesen

(Bild: Purple Slog, CC BY 2.0)

Über das Telemetrie-Protokoll MQTT spricht eine unüberschaubare Zahl an IoT-Sensoren in etwa Autos und Flugzeugen mit ihren Servern – unverschlüsselt, ohne Frage nach Passwörtern. Hacker könnten nicht nur mitlesen, sondern Daten auch manipulieren.

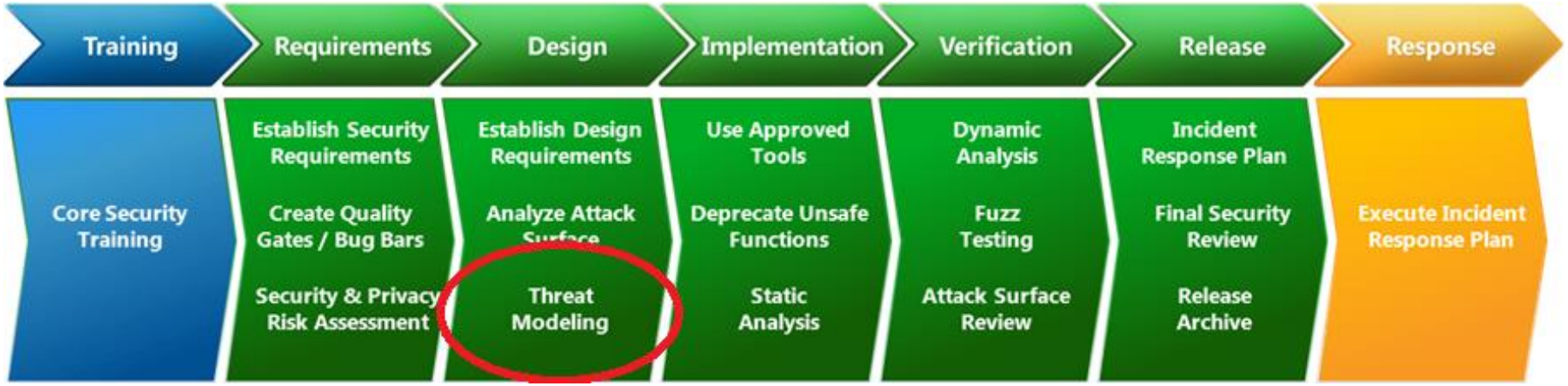
Quelle: heise online / golem.de / wired.com

Agenda

1. **Wie sichere ich mein System ab?**
2. **Bedrohungsmodellierung allgemein**
3. **Bedrohungs- und Risikoanalyse an einem Beispiel**
4. **Verbreitete Schwachstellen und Gegenmaßnahmen**

Wie sichere ich mein System ab?

Exkurs: Microsoft Security Development Lifecycle



Quelle: https://www.microsoft.com/en-us/sdl/

IoT – Beispielarchitektur – Heimnetzwerk

Angreifer

Heimnetzwerk

NAS

Router/
Firewall

Internet / externe Dienste

Bewohner

Besucher

IoT – Beispielarchitektur – Heimnetzwerk

Angreifer

Heimnetzwerk

NAS

Smart Home HUB

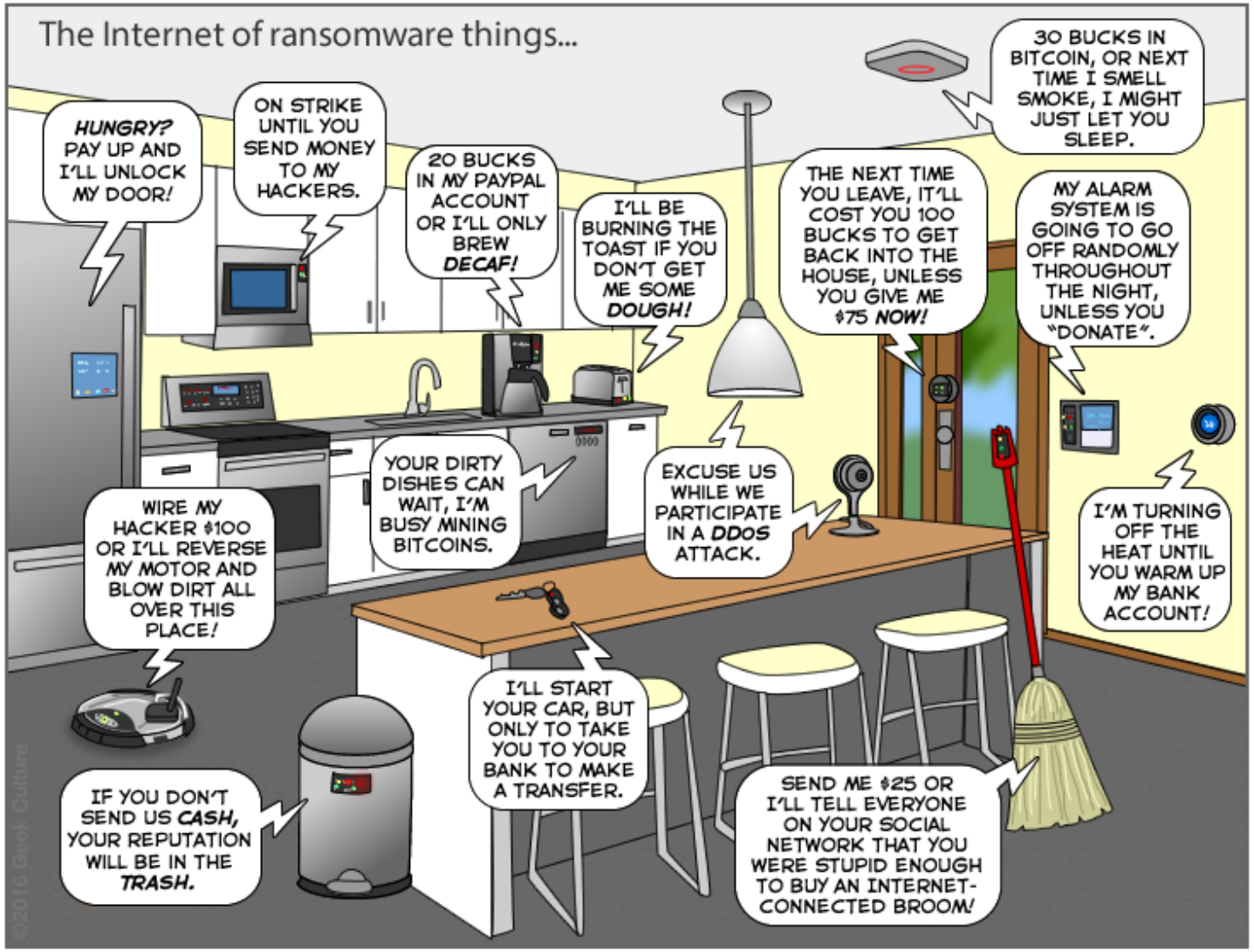
Router/
Firewall

Internet / externe Dienste

Bewohner

Besucher

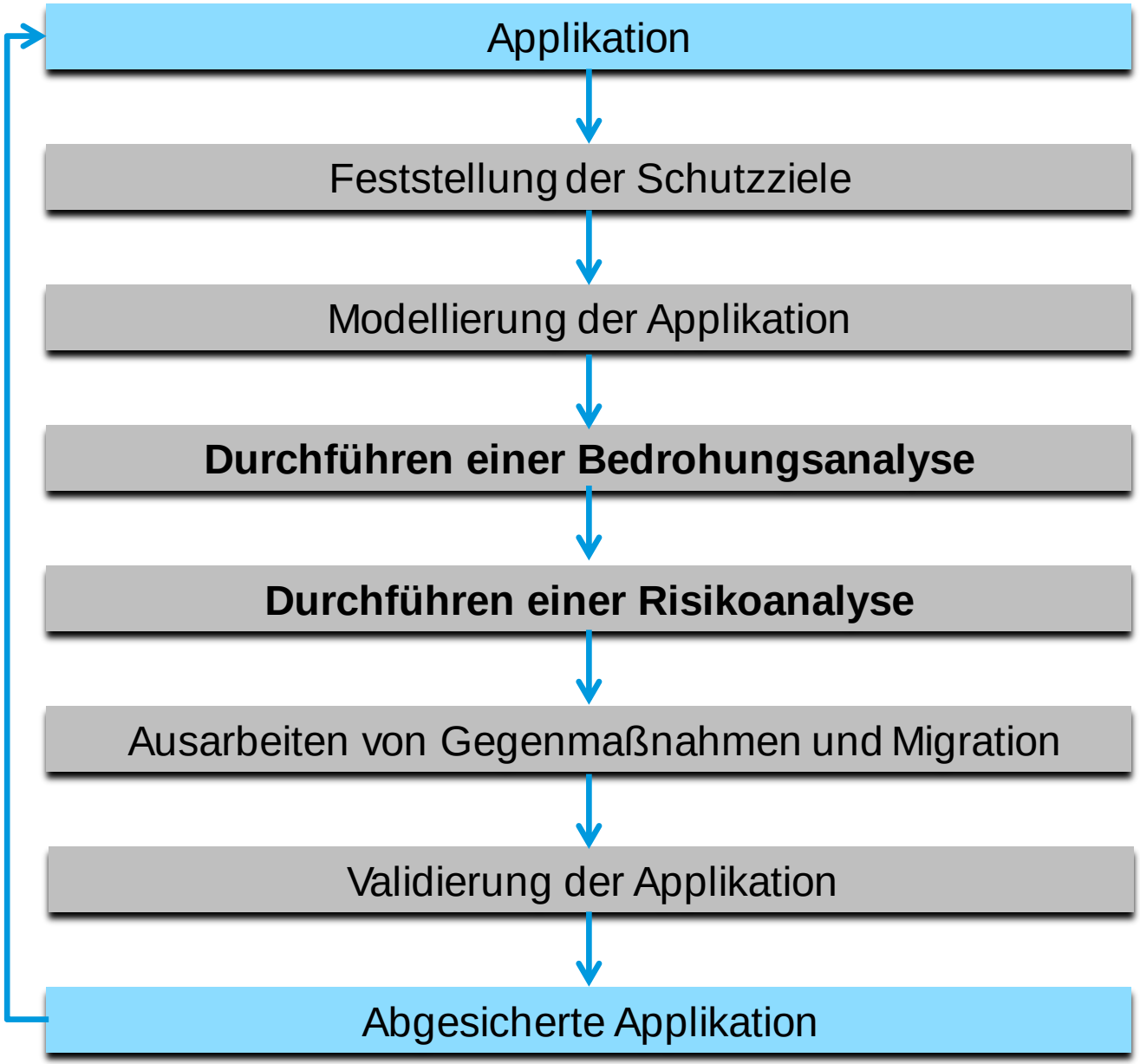
The Joy of Tech™ by Nitrozac & Snaggy



You can help us keep the comics coming by becoming a patron!
www.patreon/joyoftech

joyoftech.com

Vorgehen bei der Bedrohungsmodellierung



Modellierung der Applikation

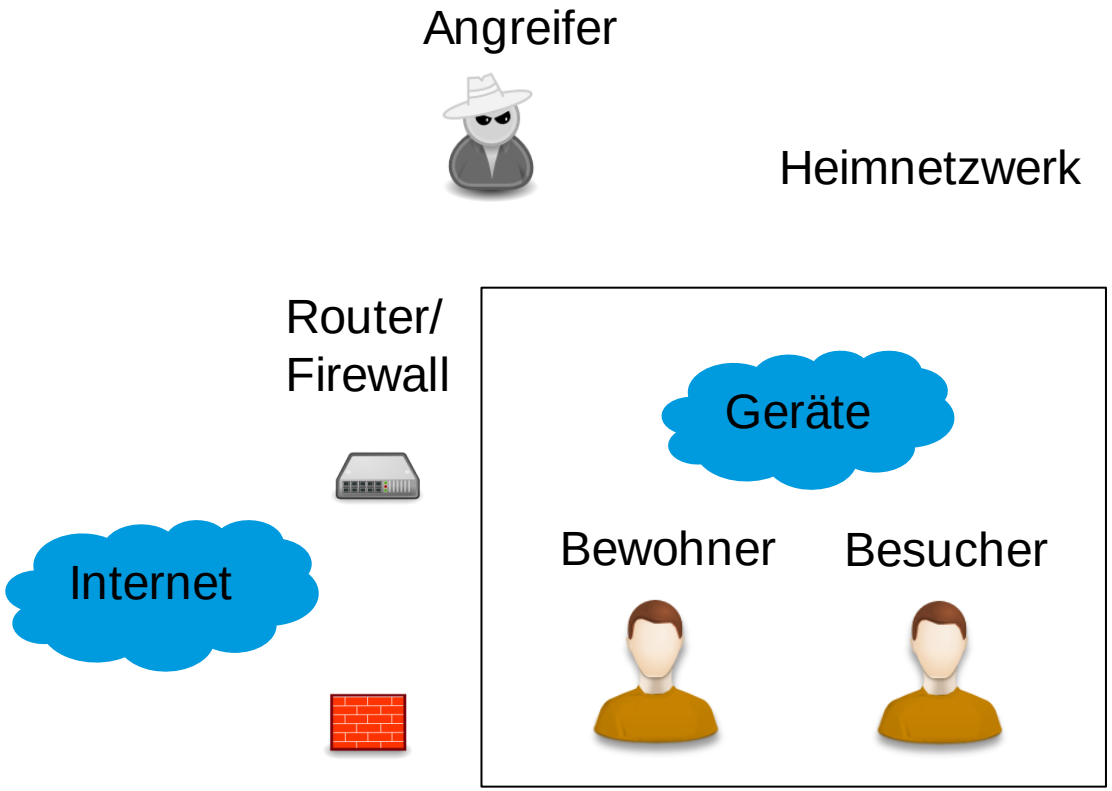
- Zusammentragen verfügbarer Informationen
- Dokumentation in einheitlicher Form
- Abgrenzung von irrelevanter Umgebung
- Besonders Relevant: Schnittstellen
- Typische Techniken:
  - Interviews
  - Sichten von Dokumenten / Standards
  - Modellierung mittels abstrakter Diagramme (Komponentendiagram/DFD)
- Schwierigkeiten:
  - Wahl einer passenden Notation
  - Wahl einer passenden Abstraktionsebene

Was sind die konkreten Bedrohungen für meine Applikation?

- Technologische Aspekte
  - Verwendete Bibliotheken
  - Hardwareeigenschaften
  - Übertragungstechnologien
- Angreifermodelle
  - Skript-Kiddie
  - Insider
  - Organisiertes Verbrechen
- Verschiedenste Vorgehensmodelle:
  - Gefährdungskataloge des BSI
  - Top n Listen (z.B. OWASP IoT - Top 10 Bedrohungen)
  - Bedrohungsmodellierung nach STRIDE

Kriterien:

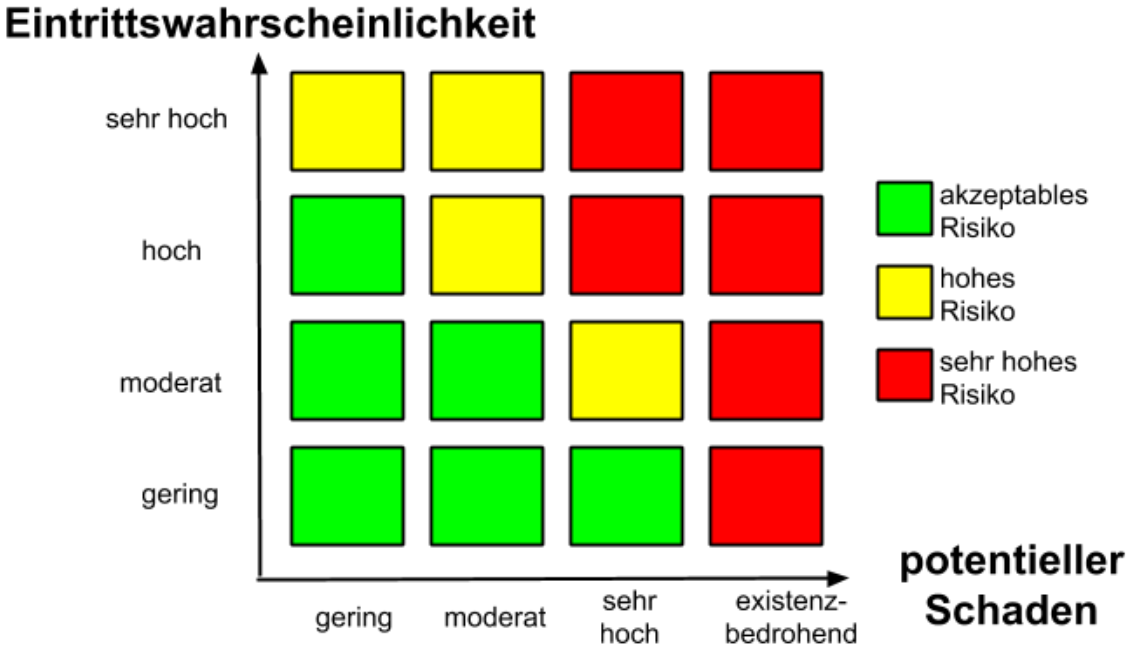
- **S**poofing Identity
- **T**ampering with data
- **R**epudiation
- **I**nformation disclosure
- **D**enial of service
- **E**levation of privileges



Ziel: Priorisierung der Bedrohungen

- Welcher Schaden kommt auf mich zu, wenn die Bedrohung eintritt?
- Wie hoch ist die Wahrscheinlichkeit, dass die Bedrohung eintritt?

Risikomatrix:



Ziel: Priorisierung der Bedrohungen

Bedrohung: Mein IoT Gerät macht bei einer DDOS Attacke mit!

Skala: z.B. 0 – 10

Damage	**8** (hoher wirtschaftlicher Schaden, mein Gerät begeht eine Straftat)
Reproducibility	**8** (hoch, viele unbekannte Sicherheitslücken, Geräte hängen meist offen im Netz)
Exploitability	**6** (nicht zu einfach, aber doch sehr verbreitet)
Affected users	**9** (theoretisch kann jeder betroffen sein)
Discoverabilty	**6** (mittlerer technischer Aufwand nötig)

37 / 5 = 7,4

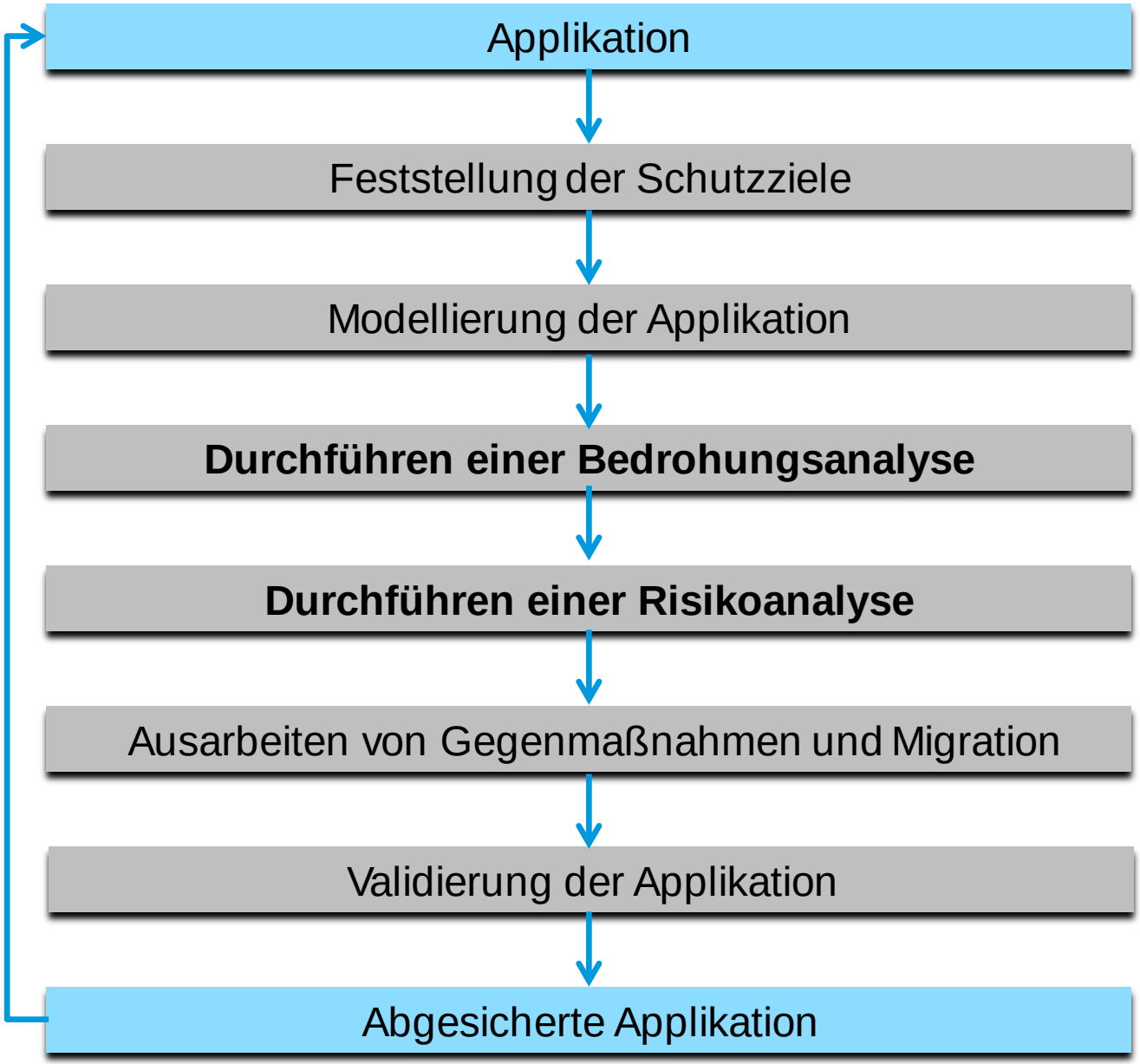
Risiko = (D + R + E + A + D) / 5

10 – 7 hohes Risiko

6 – 4 mittleres Risiko

3 – 0 geringes Risiko

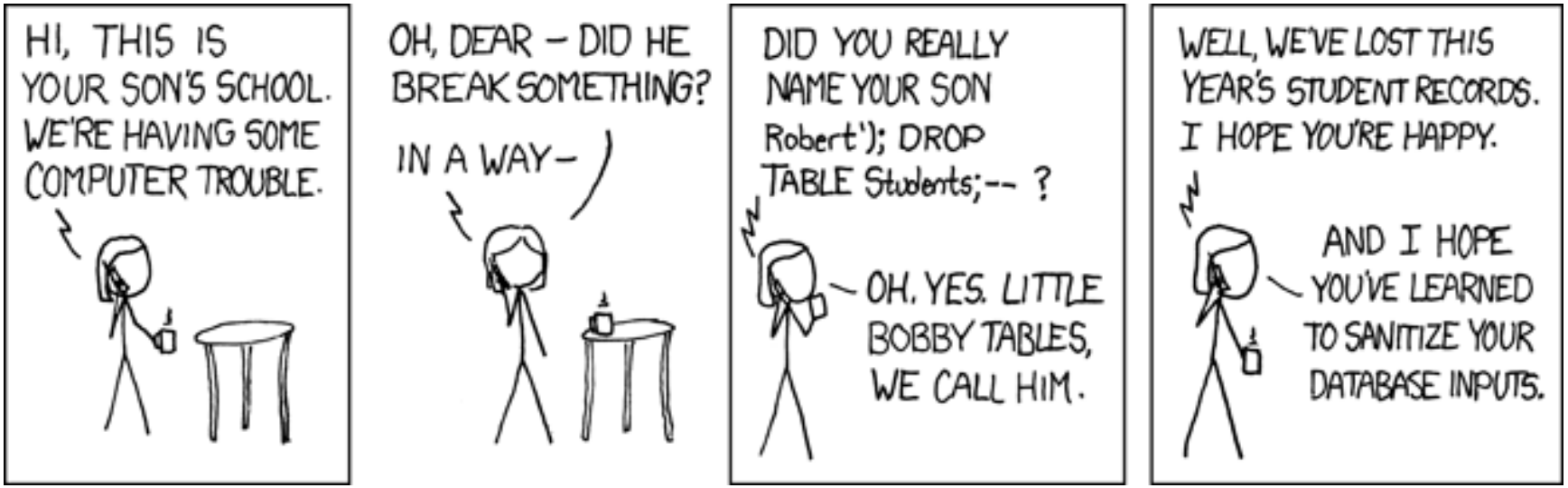
Ausarbeitung von Gegenmaßnahmen



Agenda

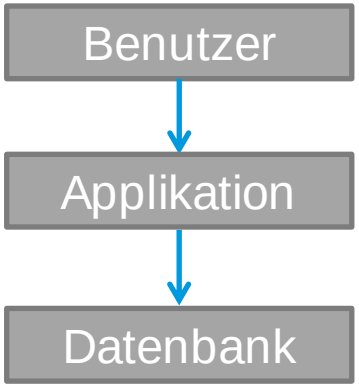
1. Wie sichere ich mein System ab?
2. Bedrohungsmodellierung allgemein
3. Bedrohungs- und Risikoanalyse an einem Beispiel
4. **Verbreitete Schwachstellen und Gegenmaßnahmen**

xkcd – Exploits of a Mom



Quelle: http://imgs.xkcd.com/comics/exploits_of_a_mom.png

Funktionsweise:



Eingabe

Übersetzt in Datenbankabfrage und führt diese aus

Beispiel:

Eingabe:

UserId:

```
userid = getRequestString("UserName");

sql = "SELECT * FROM Users WHERE Name ='" + userid;

execute(sql);
```

→ Tommy
→ Tommy or 1 == 1
→ Tommy; DROP Table Students

SQL – Injection: Gegenmaßnahmen

- Wenn möglich Bibliothek für den Datenbankzugriff verwenden
- Wenn nicht möglich, dann:
  - Benutzung von „Prepared Statements“ (parametrisierte Statements)
  - Stored Procedures
  - Benutzereingaben filtern (sqlSafe Funktion)

Unsichere Passwortspeicherung

LinkedIn-Hack: 117 Millionen Passwort-Hashes zum Download aufgetaucht

01.06.2016 15:39 Uhr – Uli Ries, Fabian A. Scherschel vorlesen

Die Angebote mit riesigen Passwort-Hash-Listen im Netz häufen sich: 117 Millionen LinkedIn-Hashes, 360 Millionen MySpace-Konten und 65 Millionen Tumblr-Hashes befeuern die Algorithmen der Cracker. Alle stammen aus alten Hackerangriffen von 2012 und 2013.

BKA findet eine halbe Milliarde ausgespähte Zugangsdaten

08.07.2017 12:34 Uhr – Thomas Hoffmann vorlesen

(Bild: dpa, Oliver Berg/Illustration)

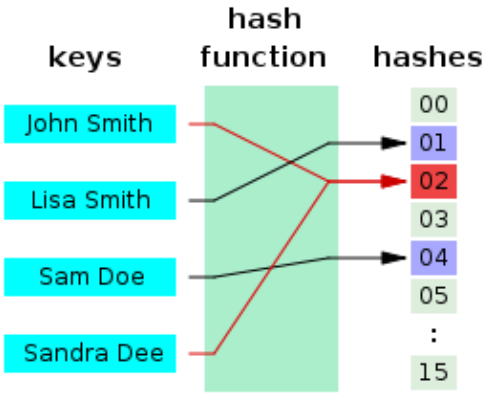
Kein Salz = schneller Erfolg für Cracker

Nachdem LinkedIn die Hashwerte lediglich mit dem Algorithmus SHA1 erzeugte und seinerzeit noch auf das Salzen der Werte verzichtet hat haben Cracker leichtes Spiel: Eine einzelne Grafikkarte AMD Radeon R9 290X erzeugt pro Sekunde über 4 Milliarden solcher Hashwerte.

… Quelle: heise online

- Basis: Hash Funktion

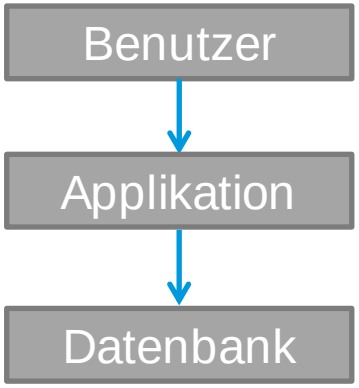
  Hashfunktionen (Streufunktionen) sind eine Abbildung einer Eingabemenge (Schlüssel) auf eine Ausgabemenge (Hashwerte)



- Deterministisch: Für eine Eingabe erfolgt immer die gleiche Ausgabe
- Es gibt keinen (vertretbaren) Weg vom Hashwert zum Schlüssel zurück
- Streuend: Ähnliche Eingaben führen zu komplett verschiedenen Ausgaben

Speicher von Passworten als Hash

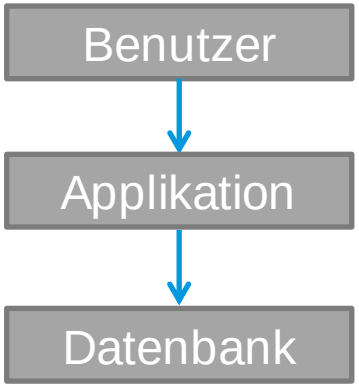
Festlegen des Passworts



setPassword(username, password)

passwordHash = generateHash(password)
storeCredentials(username, passwordHash)

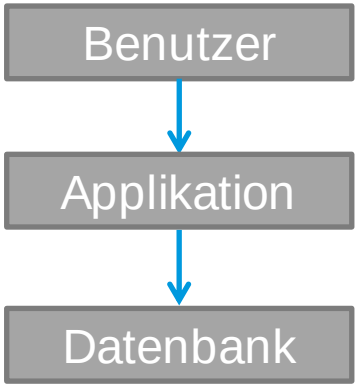
Einloggen mit festgelegtem Passwort



login(username, password)

passwordHash = generateHash(password)
isValidUser(username, passwordHash))

Festlegen des Passworts



setPassword(username, password)

```
salt = generateSalt()
saltedPasswordHash = generateHash(salt + password)
storeCredentials(username, salt, saltedPasswordHash)
```

Verbreitete Schwachstellen (nach OWASP IoT)

- Unverschlüsselte Übertragung oder Speicherung
- Schlecht oder falsch implementierte Verschlüsselung
- Nicht fest verbaute Speichermedien
- …

Verbreitete Gegenmaßnahmen

- Verschlüsselte Übertragung / Speicherung
- Richtige Rechte- und Rollenverwaltung
- Passwörter nicht im Klartext speichern
- Redundante Instanzen einführen
- …

Handlungsempfehlungen

- Nicht-vertrauenswürdige Komponenten in separates Netz
- Unnötigen Netzwerkverkehr unterbinden
- Updates regelmäßig einspielen
- Augen auf bei der Produktauswahl
- Standardpassworte immer ändern
- Netzwerkverkehr überwachen
- …

Was fällt Ihnen noch ein?

Neugierig auf Mixed Mode?

MIXED MODE

Embedded & Software Engineering

technik.mensch.leidenschaft

Professional User Interface

Embedded Linux

jobs@mixed-mode.de

Test & Quality

Internet of Things

We need you!

Embedded Security