



Technische Hochschule  
Ingolstadt  
Fakultät Informatik

# *Softwaresicherheit & Security Testing Kapitel 1: Einleitung*

*Prof. Dr.-Ing. Hans-Joachim Hof*

## *Ziele der heutigen Veranstaltung*



- **Dozent und Studenten haben sich gegenseitig kennengelernt**
- **Die Erwartungen der Studenten sind klar**
- **Dem Dozenten ist der Kenntnisstand der Studenten klar**
- **Dozent und Studenten haben Spielregeln für die weitere Veranstaltung vereinbart**
- **Die Studenten haben einen Überblick über die Vorlesung Softwaresicherheit & Security-Testing erhalten**
- **Die Studenten kennen die Bedeutung von Softwaresicherheit**



## Prof. Dr. Hans-Joachim Hof (Gerne siezen)

 [Jetzt verifizieren](#)

Vice President of Technische Hochschule Ingolstadt

Metropolregion München · [Kontaktinfo](#)

**13.555 Follower:innen · 500+ Kontakte**

 Technische Hochschule  
Ingolstadt

 Karlsruher Institut für  
Technologie (KIT)

### Berufserfahrung

 Technische Hochschule Ingolstadt  
7 Jahre 11 Monate

Chief Information Officer  
Vollzeit  
Okt. 2022–Heute · 1 Jahr 10 Monate  
Ingolstadt, Bavaria, Germany

Vice President Digitalization  
Vollzeit  
Okt. 2022–Heute · 1 Jahr 10 Monate  
Ingolstadt, Bavaria, Germany

Research Professor Automotive Cybersecurity  
Vollzeit  
Okt. 2022–Heute · 1 Jahr 10 Monate  
Ingolstadt, Bavaria, Germany

Head of research group "Security in Mobility" with a focus on automotive security, pentesting, static code analysis.

Kenntnisse: ISO 21434

Head of Research Group „Security in Mobility“  
Vollzeit  
Juli 2020–Heute · 4 Jahre 1 Monat  
Ingolstadt

Full Professor  
Sept. 2016–Heute · 7 Jahre 11 Monate  
Ingolstadt, Germany

Vice President University Expansion and Special Projects  
Vollzeit  
Okt. 2019–Sept. 2022 · 3 Jahre  
Ingolstadt, Germany

### Member Of The Supervisory Board

 AININ  
Apr. 2019–Heute · 5 Jahre 4 Monate  
Ingolstadt, Germany

 Research Group Leader  
INSicherheit - Ingolstädter Forschungsgruppe Angewandte IT-Sicherheit  
Sept. 2016–Heute · 7 Jahre 11 Monate  
Ingolstadt, Germany

 Member Board Of Directors  
Gesellschaft für Informatik e.V.  
Jan. 2016–Mai 2023 · 7 Jahre 5 Monate

 Member Of The Supervisory Board  
bizon AG  
Feb. 2016–Dez. 2019 · 3 Jahre 11 Monate  
Munich

### Hochschule für angewandte Wissenschaften München

Lecturer  
Befristet  
Sept. 2016–Feb. 2018 · 1 Jahr 6 Monate  
Munich, Germany

Full Professor  
März 2011–Aug. 2016 · 5 Jahre 6 Monate  
  
As a full professor at Munich University of Applied Sciences, my fields of research include IT security, network security, software security, web applications.

 Research Group Leader  
MuSe – Munich IT Security Research Group  
Jan. 2012–Feb. 2018 · 6 Jahre 2 Monate  
Munich  
<http://muse.bayern>

 Research Scientist  
Siemens AG, Corporate Technology, IC3, Communication and Network Security  
Jan. 2008–März 2011 · 3 Jahre 3 Monate  
  
Research in the fields of network security, especially security of sensor networks

 Researcher  
Karlsruher Institut für Technologie (KIT)  
Jan. 2003–Dez. 2007 · 5 Jahre  
  
PhD student researching security in sensor networks and general IT security.

 Intern  
SAP Markets  
2001–2001 · Weniger als ein Jahr

 GChACM, German Chapter of the ACM  
10 Jahre 7 Monate

Past Chairman (Member of the board)  
Jan. 2020–Heute · 4 Jahre 7 Monate

Chairman  
Jan. 2016–Dez. 2019 · 4 Jahre  
Munich

Vice Chairman  
Jan. 2014–Dez. 2015 · 2 Jahre  
Munich Area, Germany

# THI - Forschungsstarke Hochschule

Institute



In-Institute



Forschungs- und Testzentrum  
CARISSMA:

**C-ISAFE**

Institute of Safety in  
Future Mobility

**C-IAD**

Institute of  
Automated Driving

**C-ECOS**

Institute of Electric, Con-  
nected & Secure Mobility

Institut für Innovative Mobilität **IIMo**  
„Elektromobilität und Lernfähige Systeme“,  
„Microelectronic Packaging“,  
„Leistungselektronik“ und  
„Motor und Antriebsstrang“

Institut für neue Energie Systeme **InES**  
„Industrielle Energiesysteme“,  
„Gebäudeenergie-systeme“,  
„Energiesystemtechnik“



**HIGHTECH**  
Agenda Bayern

**AI Motion Bavaria**  
KI-Mobilitäts-Knoten



Forschungs- und Transfer- **ForTraNN**  
zentrum Nachhaltigkeit Neuburg

Bayerisches  
Foresight Institut

**BayForesight**

An-Institute

**inas**

Institut für angewandte  
Nachhaltigkeit

A I N I N  
• • • •

**Fraunhofer**  
Anwendungszentrum

# Forschungs- und Testzentrum CARISSMA

Wissenschaftliches Leitzentrum für Fahrzeugsicherheit in Deutschland



## Center of Automotive Research on Integrated Safety Systems and Measurement Area



- > Realisierung, Erprobung und Absicherung innovativer Sicherheitssysteme
- > Steigerung der Sicherheit im Straßenverkehr mit Hilfe von integralen und kooperativen Sicherheitsfunktionen als Beitrag zur "Vision Zero"



Inbetriebnahme 06/2016



3 Institute  
(C-ISAFE, C-IAD, C-ECOS)



Kosten 28 Mio. €  
(50 % Bund, 50 % Freistaat Bayern)



10 Laborbereiche



17 Professoren



100 Mitarbeiter



- Simulationssysteme Car2X-Kommunikation
- Testmethoden und Optimierung von SW-basierten Systemen
- Fahrzeugsicherheitssysteme und kooperative Fahrzeugsteuerung
- Mobile Edge Computing
- Teleoperiertes Fahren

### Car2X Kommunikation und Cyber-Physikalische Systeme

Prof. Dr. C. Facchi

### Car2X Kommunikation und Intelligente Verkehrssysteme

Prof. A. Festag

- Kommunikationsgestützte Fahrzeugsicherheit
- Verkehrseffizienz und Fahrzeugautomatisierung
- Car2X Protokolldesign, Prototyping, Leistungsbewertung
- Verkehrsinfrastruktur, Verkehrssteuerung
- Standardisierung

### Security in Mobility

Prof. Dr. H.-J. Hof

- KI-gestütztes Automotive Penetration Testing
- Automotive Forensics
- Secure Automotive Software
- Security Controls für Fahrzeuge
- Security for Vehicle Infrastructure

### Sichere Elektromobilität und Unfallanalyse

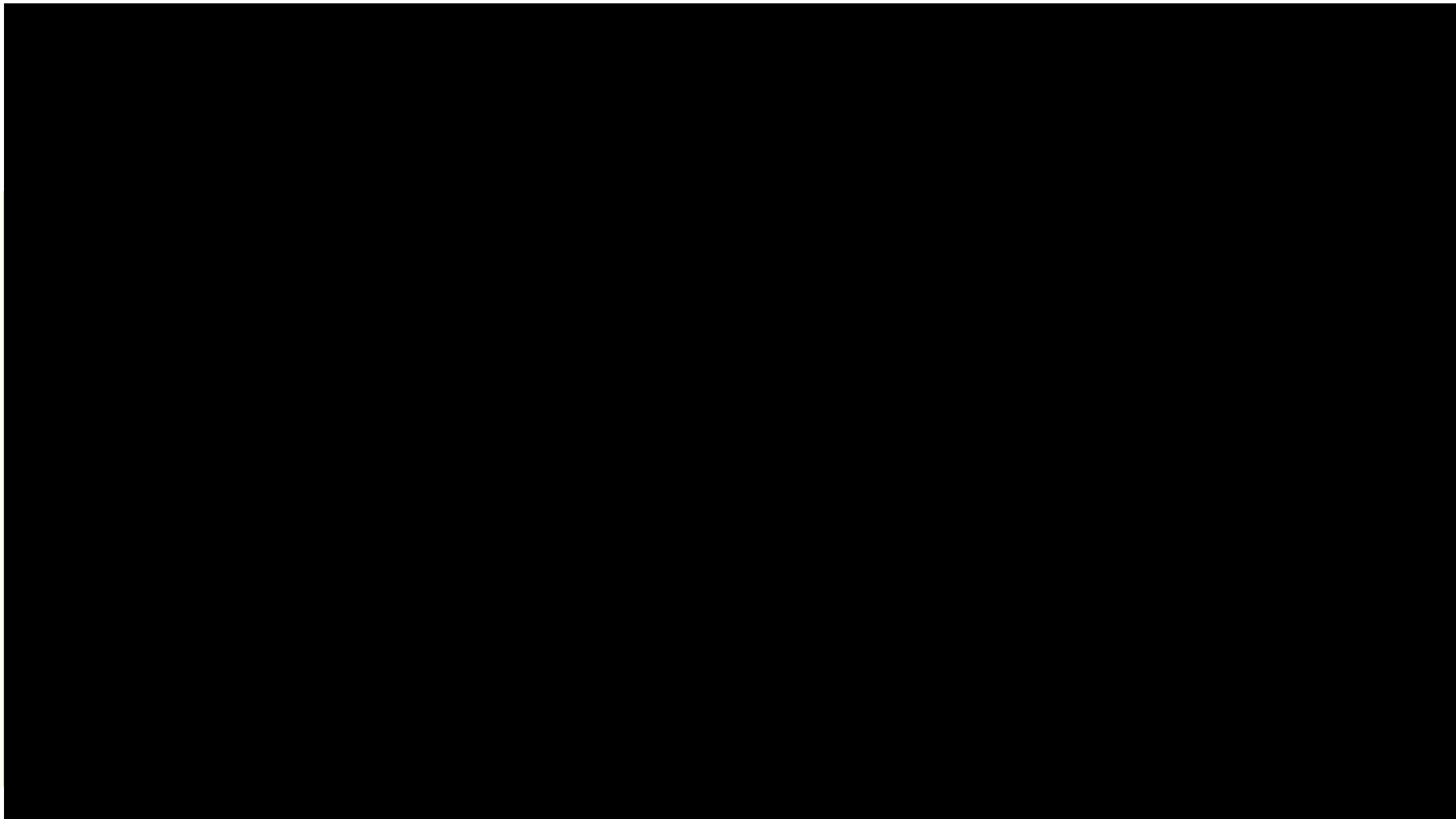
Prof. Dr. H.-G. Schweiger

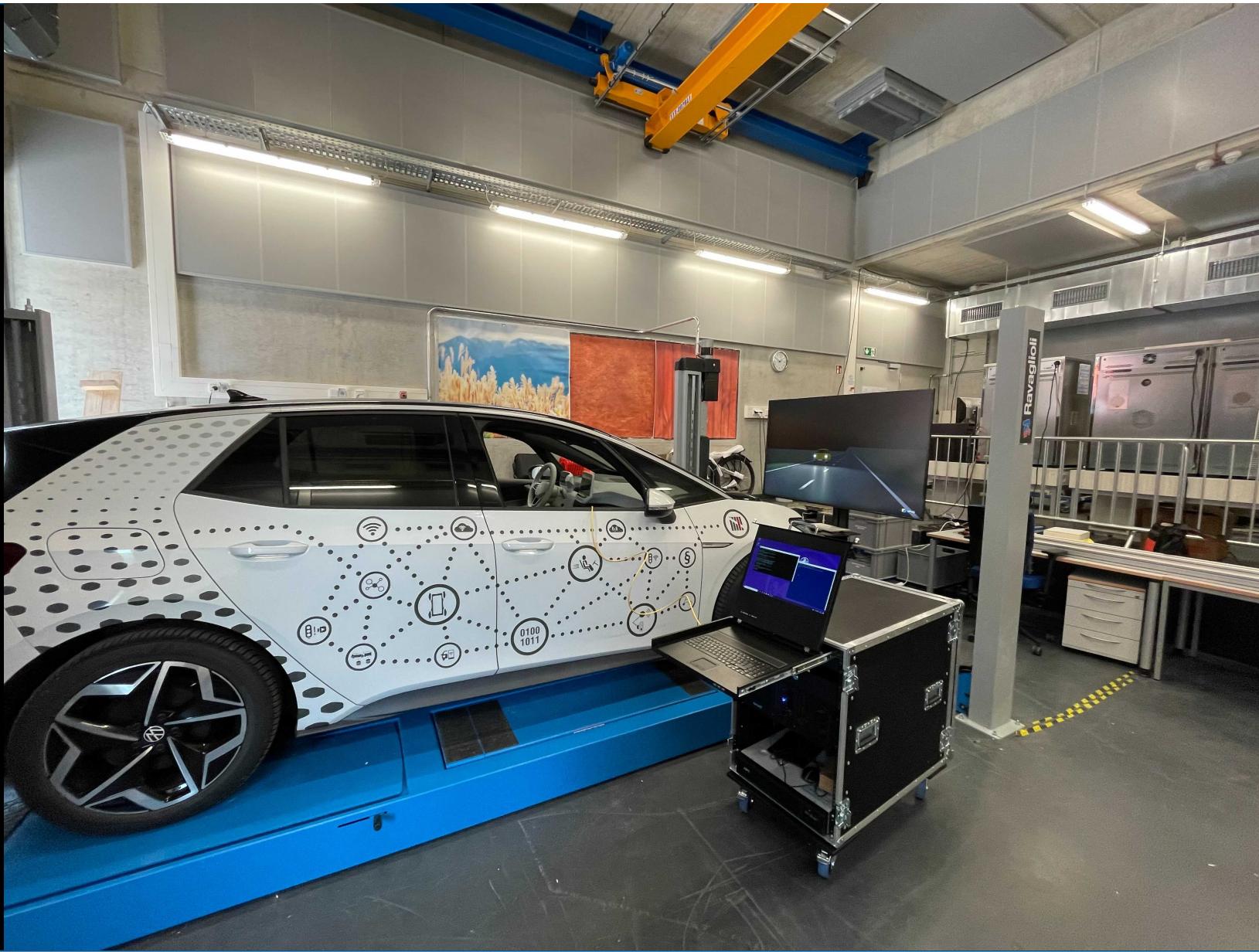
- Sicherheit Energiespeichersysteme
- Sichere, nachhaltige und robuste Speicherkonzepte
- Batteriemodllierung und Batteriecharakterisierung
- Abuse-Tests und Brandforschung
- Wasserstoffspeicher und Brennstoffzellen
- Unfallanalytik und Forensik

### Impact Analysis and Structural Integrity

Prof. Dr. U. Burger

- Interaktion von Batteriesystemen und Leichtbaustrukturen
- Crash- und Impactvorgänge
- Foreign Object Damage
- Anwendungen in der Luftfahrttechnik
- Numerische Simulation
- Analyse von Impactor und Strukturauteilen
- Modellierung von Batteriestrukturen











## Research Group „Security in Mobility“



**Prof. Dr.-Ing. Hans-Joachim Hof**  
Research Professor „Automotive Security“  
Vice President  
Chief Information Officer



**Prof. Patrizia Heinl**  
Junior Professor  
Network security and Artificial Intelligence



**Dr. Kevin Gomez Buquerin**  
Senior Researcher  
Automotive forensics



**Tina Volkersdorfer**  
Researcher  
Security modeling



**Marco Michl**  
Researcher  
Automotive blockchain



**Julian Blümke**  
Researcher  
Security for battery management systems



**Jakob Löw**  
Researcher  
Automotive penetration testing



**Dominik Bayerl**  
Researcher  
Automotive penetration testing  
AI static binary analysis



**Vishwa Vimukthi Vasu Ashoka**  
Researcher  
Cyber security for EVs



**Claudius Laves**  
Researcher  
Security for mobility data spaces



**Lea Achter**  
Researcher  
AI Security



**N.N.**  
Researcher  
Automotive Quantum Security-Architectures



**Parul Gupta**  
Researcher  
GenAI for Code Security



**Christopher Corbett**  
External PhD student (Audi)  
Automotive Intrusion Detection



**Lukas Eder**  
Researcher  
Cyber security for EVs



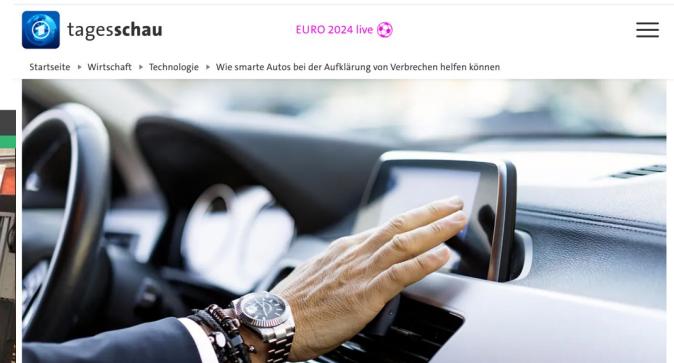
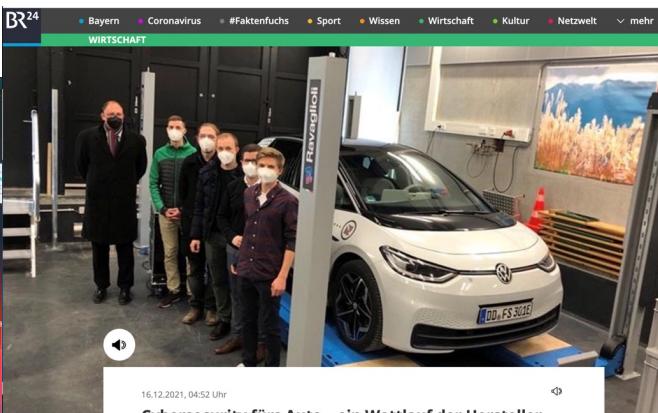
**Julianne Eder**  
Researcher  
AI Security



**Jenny H.**  
Student researcher  
Vehicle Security Operations Center



## Unsere Forschungsgruppe hat einen exzellenten Ruf



Bordtechnik und Verbrechensbekämpfung  
**Wie smarte Autos den Ermittlungsbehörden helfen**

Stand: 02.06.2024 15:08 Uhr  
Fahrzeile mit dem Smartphone ans Auto schicken oder arglos über die Freisprecheinrichtung quatseln: Was für Fahrer moderner Autos zum Komfort gehört, macht auch die Arbeit von Ermittlern einfacher.

Von Juri Sonnenholzer, SWR

## Handelsblatt

Anmelden

> IT + Telekommunikation > BlackBerry: QNX-Sicherheitslücke gefährdet „hochsensible Systeme“

Betriebssystem QNX

## Schwere Sicherheitslücke bei BlackBerry gefährdet Millionen Autos

Vernetzte Autos, Produktionsmaschinen, Medizinprodukte: Böswillige Angreifer könnten die Kontrolle über hochsensible Systeme erlangen. Bei der Warnung der Kunden schlampete BlackBerry.

Christof Kerkmann  
18.08.2021 - 17:13 Uhr aktualisiert

## Curriculum

### 1. Semester

Forschungsmethoden und -strategien 1	Projektarbeit 1
Projektseminar	Mathematisch-physikalisch-naturwissenschaftliches Modul

### 2. Semester

Forschungsmethoden und -strategien 2	Projektarbeit 2	Projektseminar
Technologisches Modul	Interdisziplinäres Modul	

### 3. Semester

Masterarbeit
--------------

## Weltweit erster Quantenschlüsselaustausch in ein Fahrzeug

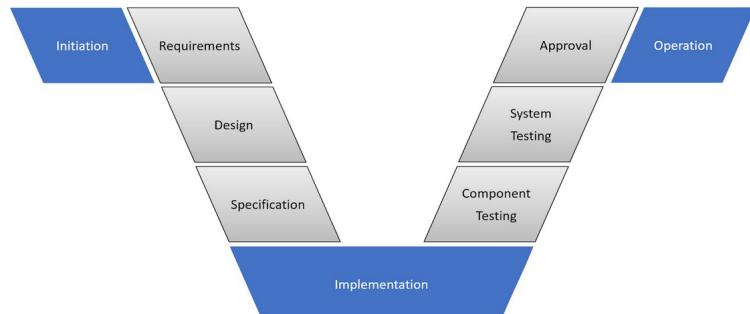
20.03.2023@Technische Hochschule Ingolstadt



## Aktuelle Forschungsprojekte



### Fahrzeug-Entwicklung



### Infrastruktur



DevGPT: Generative AI for Automated automotive code generation (2024-2026)



AutoBERT: Static binary analysis of automotive firmware images using AI



MASSiF: Angreifer- und Angriffsmodellierung zur Erzeugung von Testfällen (2019-2022)



CANoRa: CAN Bus Obfuscation (2016-2018)



HATS3: Design of an automotive penetration testbench (2020-2022)



TRADE: Automotive blockchain for decentralized identities (2021-2024)



LLMs for hacking (ongoing)



SELFY: Vehicle Security Operations Center and Intrusion Detection (2022-2025)



PLIADES: Secure mobility data spaces (2024-2027)

iBattMan: Security for transition of Battery Management System from first to second life (2024-2027)



SQUIRRAL: Integration of Quantum Keys in Electric Vehicles Architectures (2024-2027)



REBORN: Security for Battery Management Systems for second life (2024-2027)



MARBEL: secure cloud for battery management systems(2021-2024)



Automotive forensics and accident analysis (ongoing)





# Vorlesung Softwaresicherheit & Security Testing

## Voraussetzungen

### Curriculum

1. Semester		
Mathematik 1 1	Grundlagen der Programmierung	Einführung in die Informatik 1
Grundlagen der IT-Sicherheit	Einführungsprojekt	Gesellschaftliche Verantwortung sowie Innere und Äußere Sicherheit
2. Semester		
Mathematik 2 2	Grundlagen der Programmierung	Einführung in die Informatik 2
Software-Entwicklungsmethodik	Sichere Systeme	
3. Semester		
Angewandte Mathematik für IT-Sicherheit	Web-Technologien	Netzwerke
Software-Design / SW-Architektur und Datenbanken	Softwaresicherheit & Security Testing	
4. Semester		
Cloud-Architekturen und -Dienste	Projekt-, Qualitäts- und Risikomanagement	Protokolle der Netzsicherheit
Security Architekturen & Security Engineering	Ethical Hacking-Praktikum	Fachwissenschaftliches Seminar
5. Semester		
Kommunikations- und Teamkompetenz	Praktikum	Nachbereitendes Praxisseminar
6. Semester		
Recht für IT-Sicherheit und Datenschutz	Grundlagen Künstliche Intelligenz und deren Anwendung in der IT-Sicherheit	Incident Response & Netzwerkmonitoring
Sichere Netzwerkarchitekturen und Sicherheit vernetzter Anwendungen	Projekt	Grundlagen der Betriebswirtschaft und des Gründertums
7. Semester		
Fachwissenschaftliche Wahl-pflichtmodule 1	Fachwissenschaftliche Wahl-pflichtmodule 2	Fachwissenschaftliche Wahl-pflichtmodule 3
Seminar Bachelorarbeit	Bachelorarbeit	



## Die Studenten

Erwartungen an die Vorlesung „Softwaresicherheit & Security Testing“ (vorab ausgefüllt)



<https://padlet.com/hof14/9hmhib1ww4pp3e0c>

- Was ist Ihnen wichtig für den Umgang in der Vorlesung „Softwaresicherheit & Security Testing“?

- Das Modul "Softwaresicherheit & Security Testing" zielt darauf ab, die Studierenden mit Techniken der Software-Sicherheit und Security-Testing-Verfahren vertraut zu machen. Dabei liegt der Fokus auf den technischen Verfahren der Softwaresicherheit.
  
- Die Studierenden sollen in der Lage sein, sichere Software zu entwickeln und bestehende Software auf Sicherheitslücken zu überprüfen. Sie erlernen Techniken und Strategien zur Sicherstellung der Code-Qualität, einschließlich statischer Code-Analyse und manueller Reviews.

## Aus der Modulbeschreibung

### Inhalte

- **Sicheres Programmieren**
- **Sicherheitslücken**
- **Sicherheit von Programmiersprachen**
- **Security Testing**
  - Pentesting
  - Source Code Reviews
  - Statische Code-Analyse
  - Manuelle Reveiws
- **Security Assessments**
- **Relevante Standards**

# Die Vorlesung im Studiengang CSI

## Abgrenzung



### Curriculum

1. Semester		
Mathematik 1	Grundlagen der Programmierung 1	Einführung in die Informatik 1
Grundlagen der IT-Sicherheit	Einführungsprojekt	sowie
2. Semester		
Mathematik 2	Grundlagen der Programmierung 2	Einführung in die Informatik 2
Software-Entwicklungsmethodik	Sichere Systeme	
3. Semester		
Angewandte Mathematik für IT-Sicherheit	Web-Technologien	Netzwerke
Software-Engineering, Struktur und Datenbanken	Softwaresicherheit & Security Testing	
4. Semester		
Cloud-Architekturen und -Design	Ebene: Low Level Design und Implementierung	
Security Architekturen & Security Engineering	Ethical Hacking-Praktikum	Fachwissenschaftliches Seminar
Ebene: High Level Design und Prozesse		
Teamkompetenz	Praktikum	Nachbereitendes Praxisseminar
6. Semester		
Recht für IT-Sicherheit und Datenschutz	Grundlagen Künstliche Intelligenz und deren Anwendung in der IT-Sicherheit	Incident Response & Netzwerkmonitoring
Sichere Netzwerkarchitekturen und Sicherheit vernetzter Anwendungen	Projekt	Grundlagen der Betriebswirtschaft und des Gründertums
7. Semester		
Fachwissenschaftliche Wahl-pflichtmodule 1	Fachwissenschaftliche Wahl-pflichtmodule 2	Fachwissenschaftliche Wahl-pflichtmodule 3
Seminar Bachelorarbeit	Bachelorarbeit	

Prozess

Fertigkeit

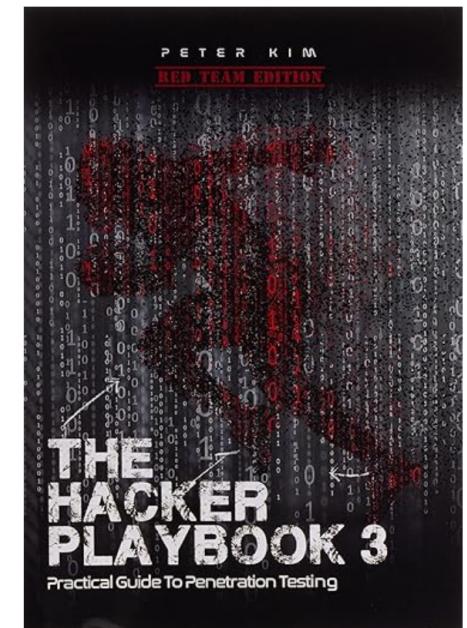
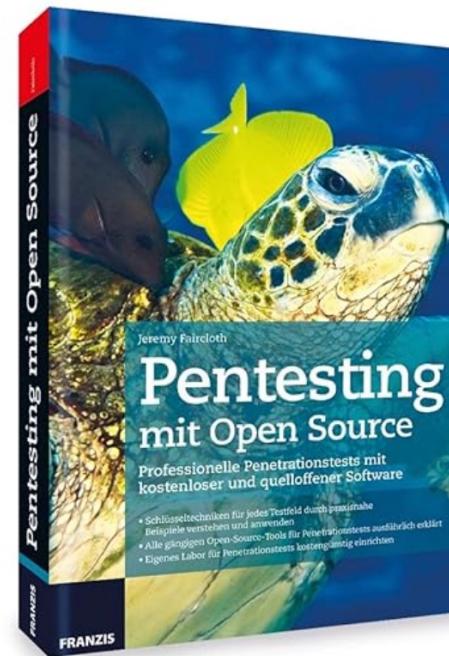
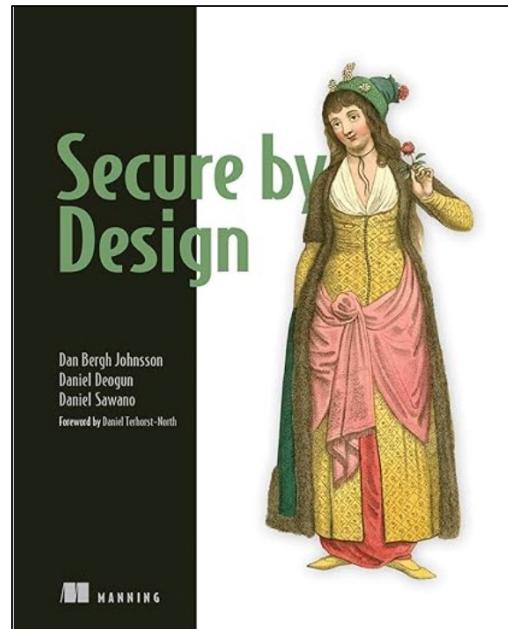
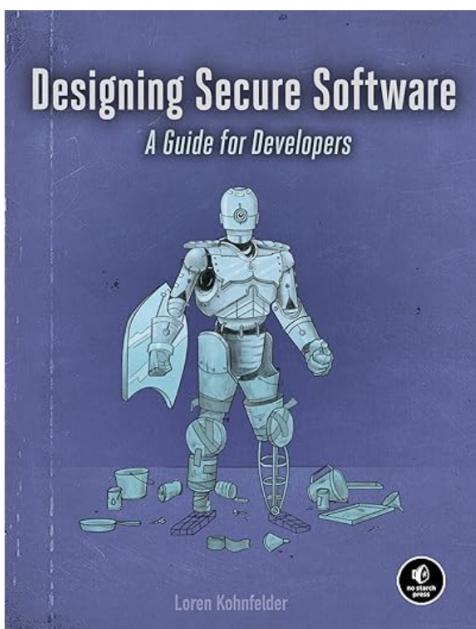
- **Moodle-Kursraum (<https://moodle.thi.de>)**
  - Vorlesungsplan (Termine für Präsenz, Online und Aufgaben) - wird ständig aktualisiert
  - Unterlagen (diese stehen bis zum Ende der Vorlesungszeit zur Verfügung und werden dann gelöscht)
  - Aufzeichnungen der Vorlesung (ohne Garantie)
  - Zusatzmaterial
  - Literaturliste
  - Nachrichtenbrett
  - Diskussionsforum
- **Einschreibeschlüssel: softSichST#WS23**
- **Leistungsnachweis: Schriftliche Prüfung, 90 Minuten, keine Hilfsmittel zugelassen**

## *Voraussichtlicher Aufbau der Vorlesung*



- **Kapitel 1: Einleitung**
- **Kapitel 2: Grundlagen**
- **Kapitel 3: Security by Design**
- **Kapitel 4: Sicheres Programmieren**
- **Kapitel 5: Statisches Security Testen**
- **Kapitel 6: Dynamisches Security Testen**

## Empfohlene Literatur





■ Herausgeber: Veracode

- Dienstleister im Bericht Softwaresicherheit
- Bericht erscheint 2024 zum 14. Mal
- Report liegt im Moodle

■ Methodik

- Analyse von Code, der zum Testen die Tools von Veracode nutzen (statische Analyse, dynamische Analyse, Software Composition Analyse, Cloud-based Pentesting)
- Mehrmalige Untersuchung gleiche Software nur als einzelne Software gezählt,
- Große und kleine Unternehmen, kommerzielle und Open-Source-Softwarehersteller

This research draws from the following:

**1,007,133**  
applications across all scan types

**1,553,022**  
dynamic analysis scans

**11,429,365**  
static analysis scans

All those scans produced:

**96.0 million**  
raw static findings

**4.0 million**  
raw dynamic findings

**12.2 million**  
raw software composition analysis  
findings



■ **Aus der Studie:**

## A Note on Mass Closures

While preparing the data for our analysis, we noticed several large single-day closure events. While it's not strange for a scan to discover that dozens, or even hundreds, of findings have been fixed (50 percent of scans closed fewer than three findings; 75 percent closed fewer than eight), we did find it strange to see some applications closing thousands of findings in a single scan. Upon further exploration, we found many of these to be invalid. These large collections of flaws were both added and removed in single scans: Developers would scan entire filesystems, invalid branches, or previous branches, and when they would rescan the valid code, every finding not found again would be marked as "fixed."

These mistakes had a large effect:

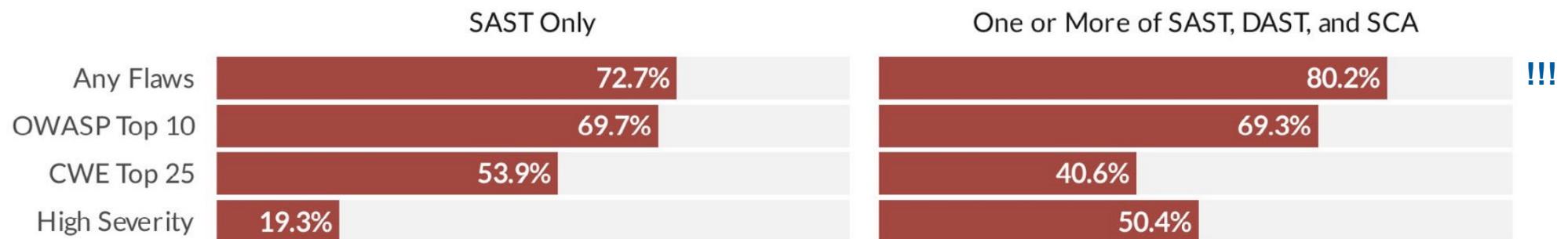
The top 0.02-percenters, or about 1 in every 6650 scans (0.02%) accounted for almost a quarter (24.6%) of all the closed findings

These "mass closure" events have significant effects on measuring flaw persistence and time to remediation and were ultimately excluded from the analysis.

## Studie: Annual Report on the State of Application Security (2024)

### Häufigkeit des Auftretens von Sicherheitslücken (security flaw) in Software

Mindestens eine Schwachstelle gefunden:



- **SAST: Static Application Security Testing (testet Source Code)**
- **DAST: Dynamic Application Security Testing (testet ausgeführte Anwendung)**
- **SCA: Software Composition Analysis (testet Abhängigkeit von verwundbaren Komponenten)**
  
- **Fragen:**
  - Unterschied SAST/SCA?
  - Warum gibt es Unterschiede in den Fehlern, die man mit SAST bzw. DAST findet?

## Studie: Annual Report on the State of Application Security (2024)

Häufigkeit von schweren Sicherheitsschwachstellen hat sich seit 2016 mehr als halbiert

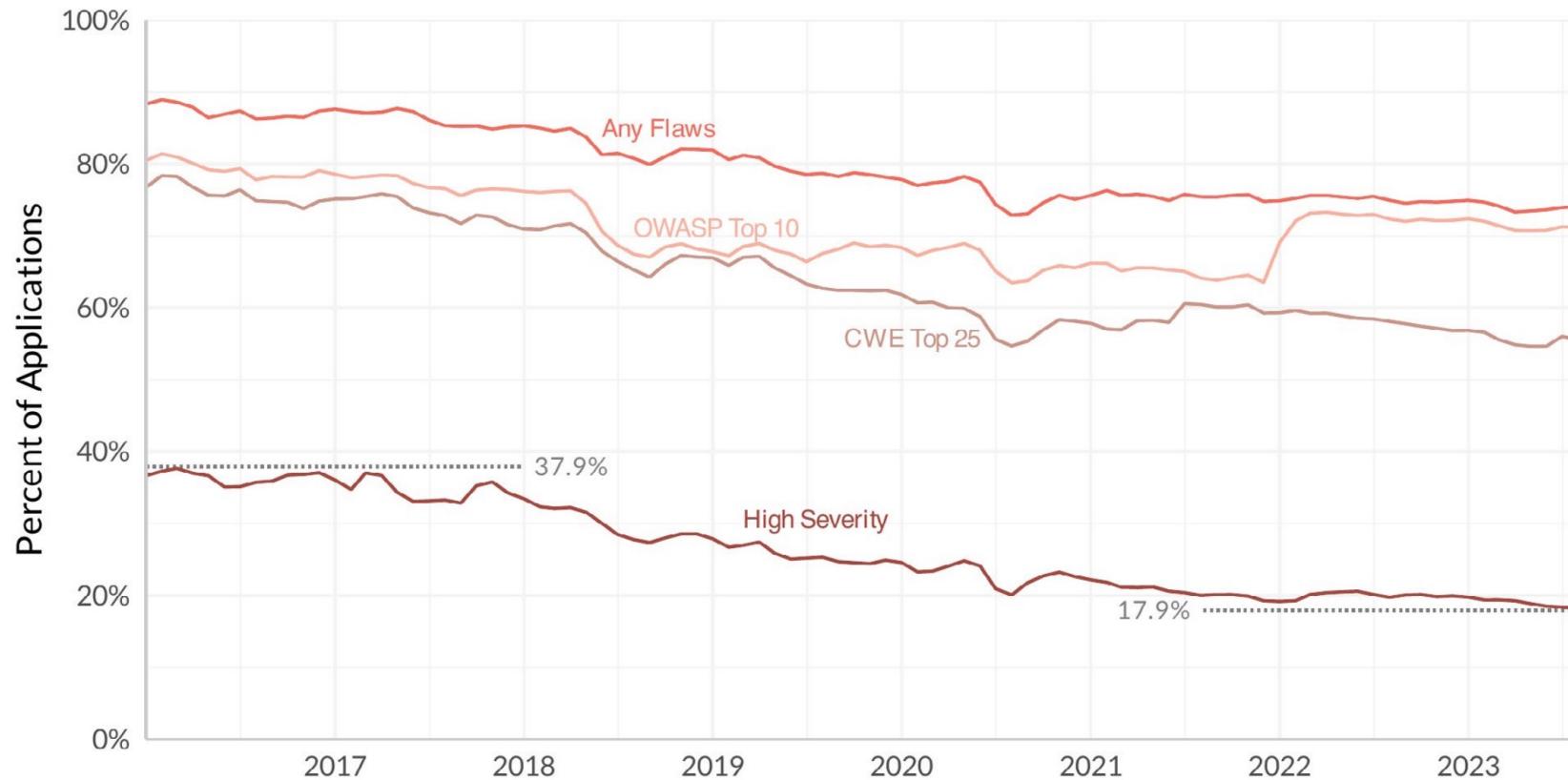


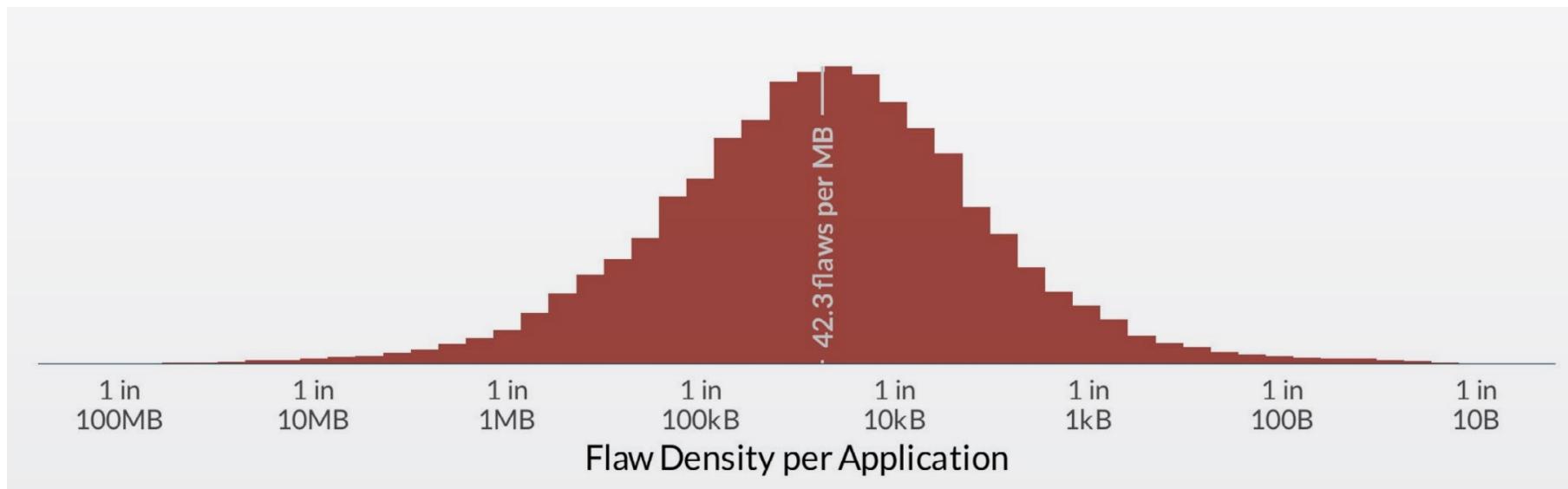
Figure 2: Trend of flaw detection rates over time



## Studie: Annual Report on the State of Application Security (2024)

Anzahl von Fehlern in einer Anwendung

- Im Durchschnitt 42 Sicherheitsschwachstellen pro 1 MB Code
  - Ja, 42, die Antwort auf die endgültige Frage nach dem Leben, dem Universum und dem ganzen Rest
- Histogramm:



- **Severity:** potentielle Auswirkungen auf Vertraulichkeit, Integrität und Verfügbarkeit
- **Exploitability:** Wahrscheinlichkeit für Ausnutzung durch Angreifer (wie einfach ist das?)

		Severity			
		Low (25.6%)	Medium (59.7%)	High (11.5%)	Very High (3.2%)
Exploitability	V. Likely (15.9%)	1.3%	10.5%	3.4%	0.7%
	Likely (36.5%)	2.1%	33.0%	1.2%	0.3%
	Neutral (37.9%)	14.4%	14.4%	6.9%	2.2%
	Unlikely (9.5%)	7.6%	1.9%	0.0%	0.0%
	V. Unlikely (0.2%)	0.2%	0.0%	0.0%	0.0%



## Studie: Annual Report on the State of Application Security (2024)

### Häufigste Sicherheitslücken

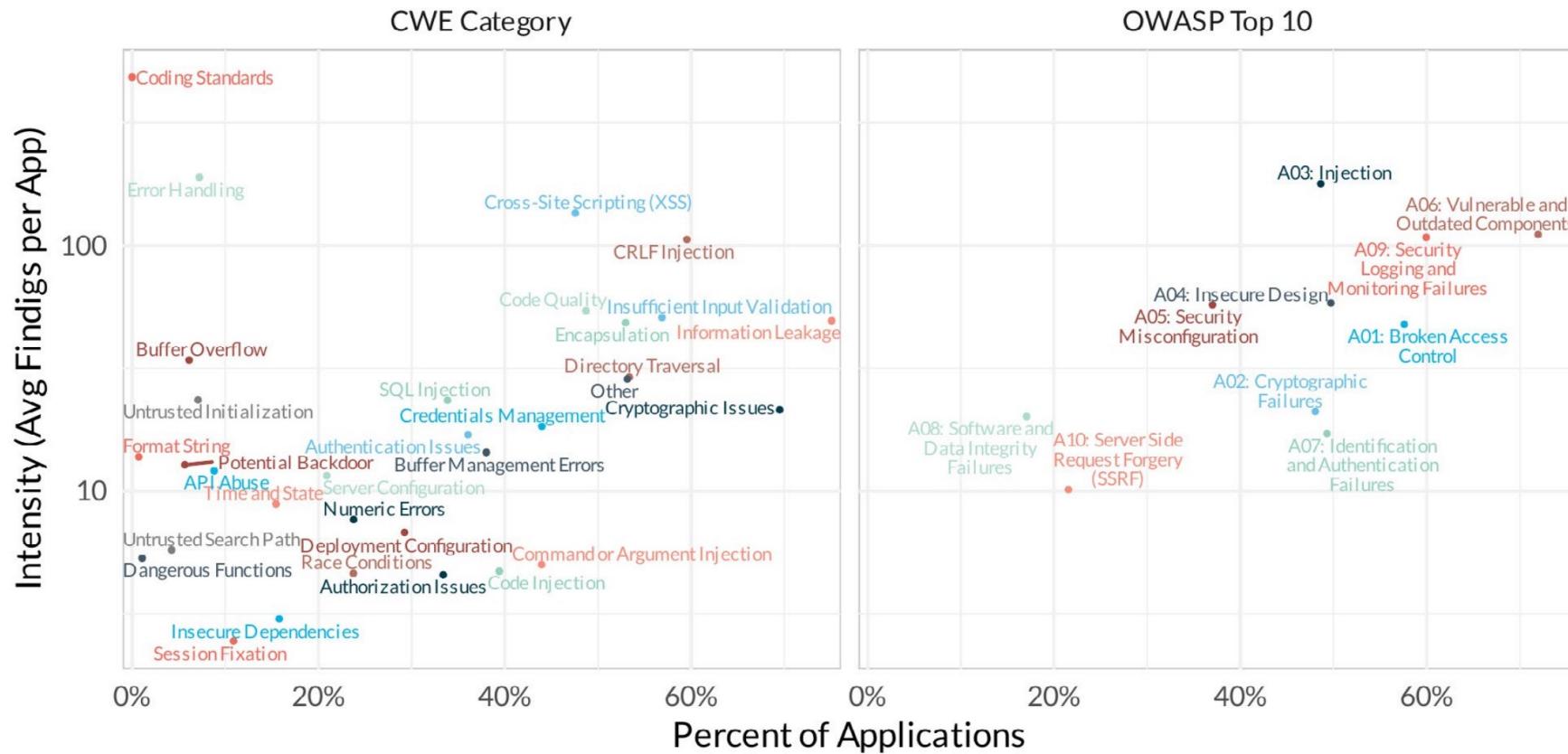


Figure 5: Prevalence and intensity of CWE and OWASP flaws in applications

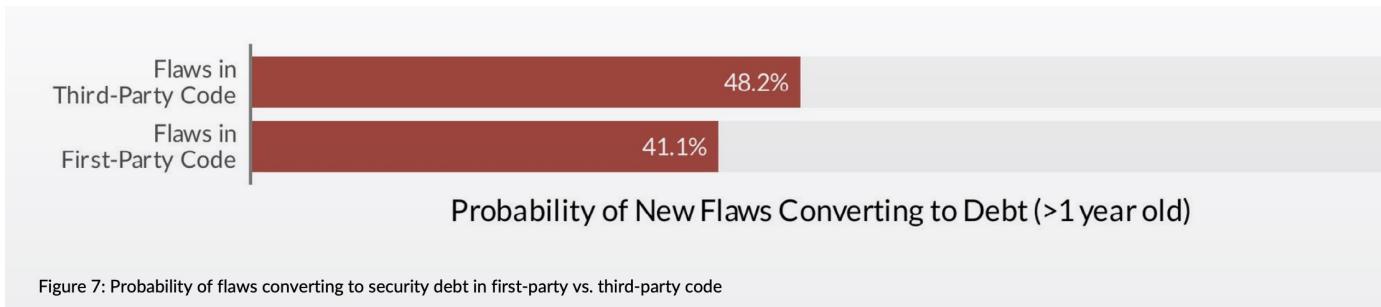


Figure 6: Prevalence of flaws in first-party (left) vs. third-party (right) code among applications

- **First-Party Code: selbst geschrieben (aus Sicht desjenigen, der testet)**
- **Third-Party Code: Code, der von extern aufgenommen wird (z.B. eingebundene Bibliotheken)**

## Studie: Annual Report on the State of Application Security (2024)

### Dauer der Behebung von Sicherheitslücken



## ■ Nach einem Jahr gehen Sicherheitslücken in so genannten Sicherheits-Schulden (Security Debt) über

- Schulden häufen sich über die Zeit auf und werden immer schwieriger zu beseitigen
- In wenigen Fällen: bewusste Entscheidung, gewisse Sicherheitslücken nicht zu beseitigen

## Studie: Annual Report on the State of Application Security (2024)

### Häufigkeit von Security Debt

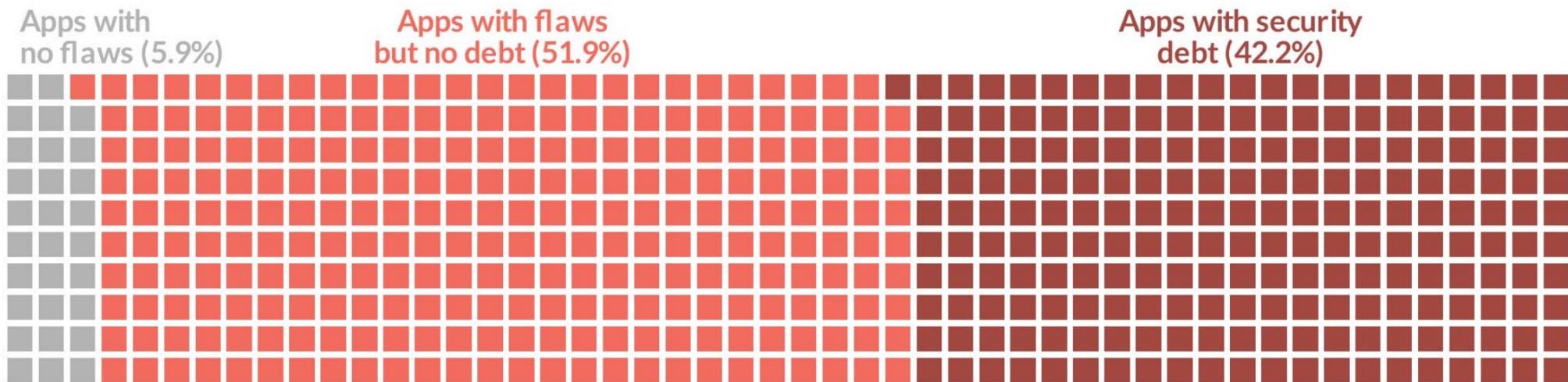


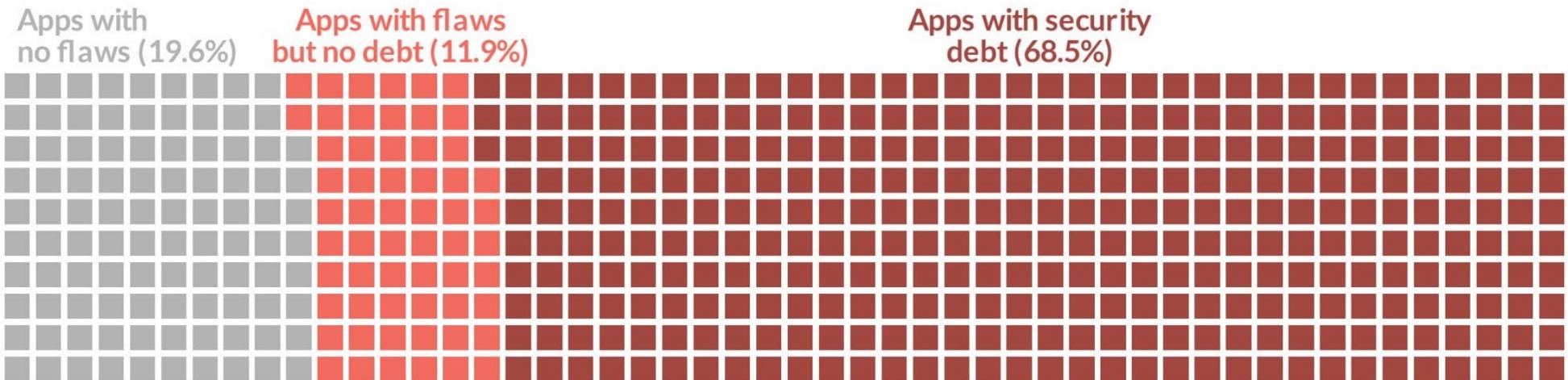
Figure 8: Prevalence of security debt across all applications greater than one year old

- **Hinweis: Zahlen beziehen sich auf Anwendungen, die wenigstens über ein Jahr hinweg gescannt wurden.**

## Studie: Annual Report on the State of Application Security (2024)



### Häufigkeit von Security Debt im eigenen Code



- Gleiche Grafik, allerdings nur aus den Ergebnissen von SAST und DAST, also OHNE SCA
  - Man sieht den Einfluss, den das Einbinden von Open Source Libraries auf das Overall Security Debt hat (geringer)

## Studie: Annual Report on the State of Application Security (2024)

Einflussfaktoren für Security Dept: Alter der Anwendung

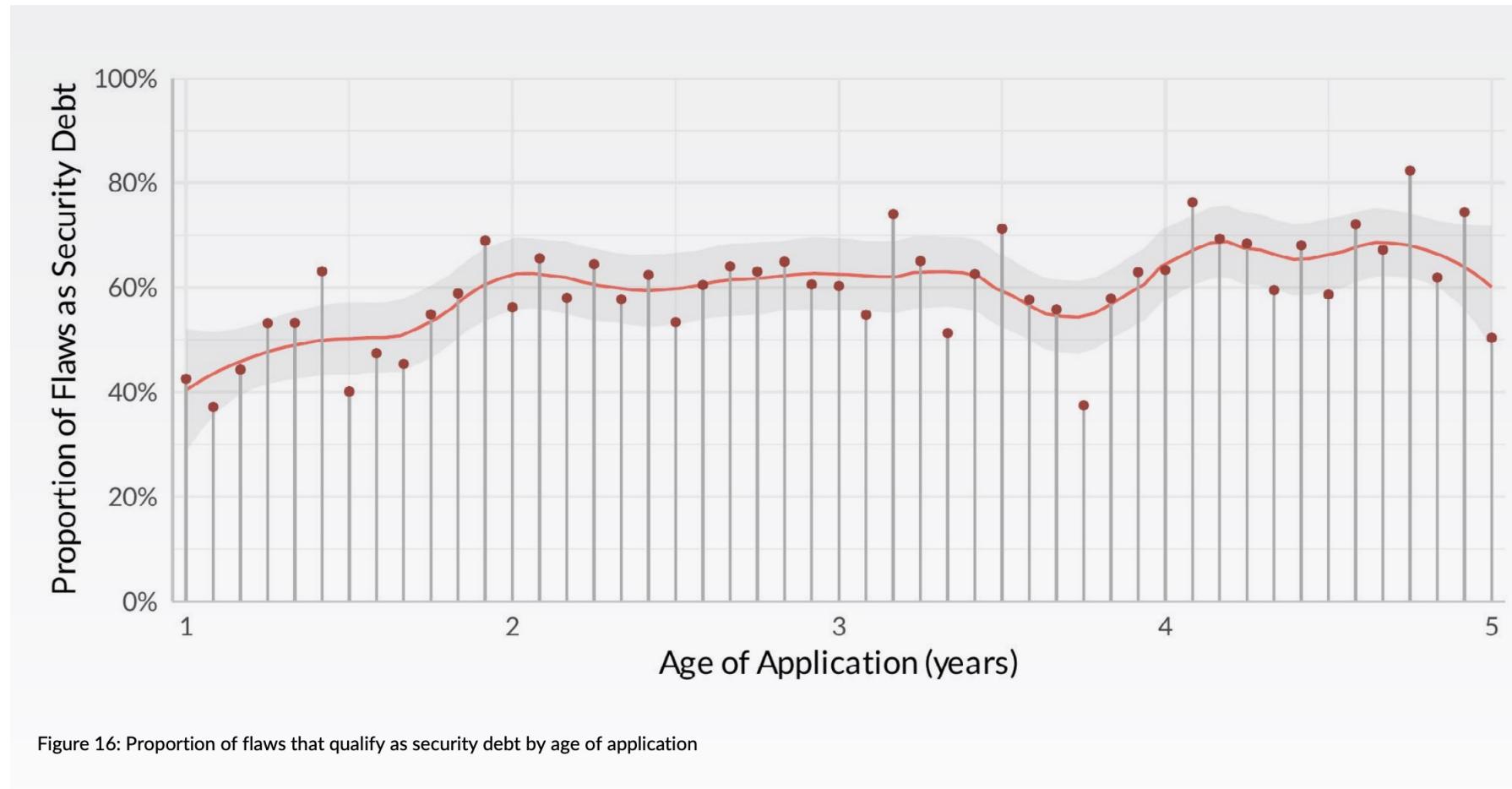
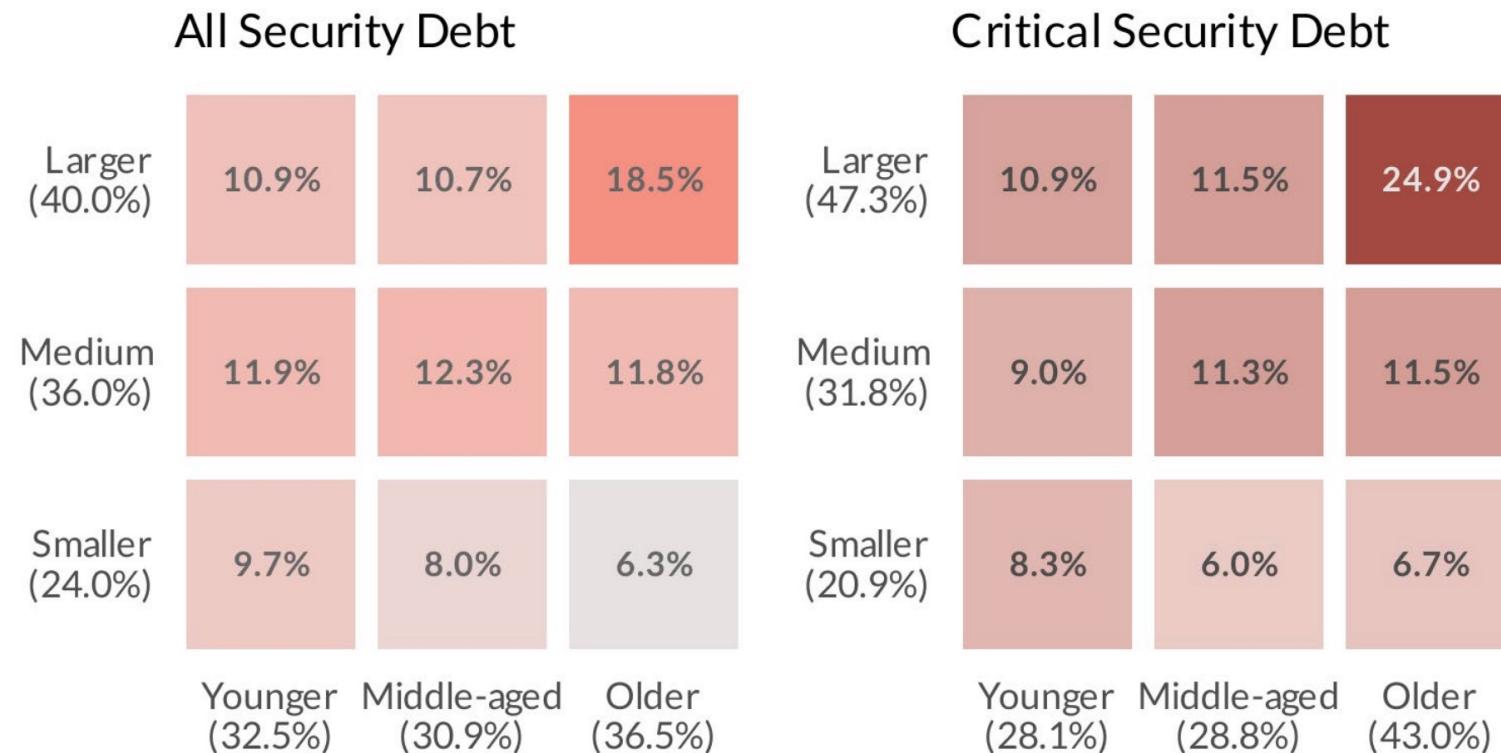


Figure 16: Proportion of flaws that qualify as security debt by age of application

## Studie: Annual Report on the State of Application Security (2024)

### Einflussfaktoren für Security Dept: Größe der Anwendung

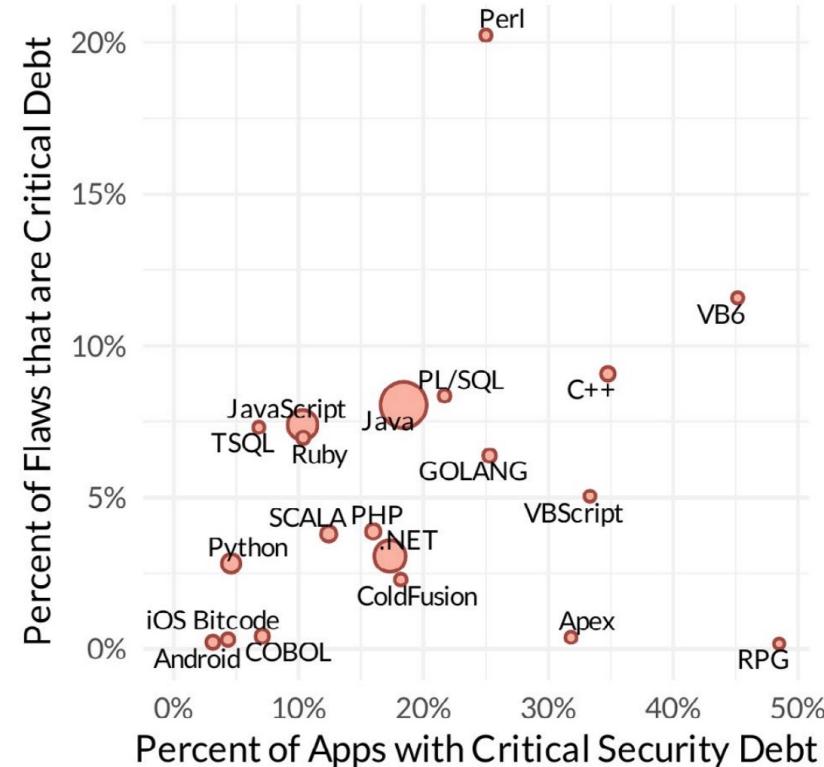
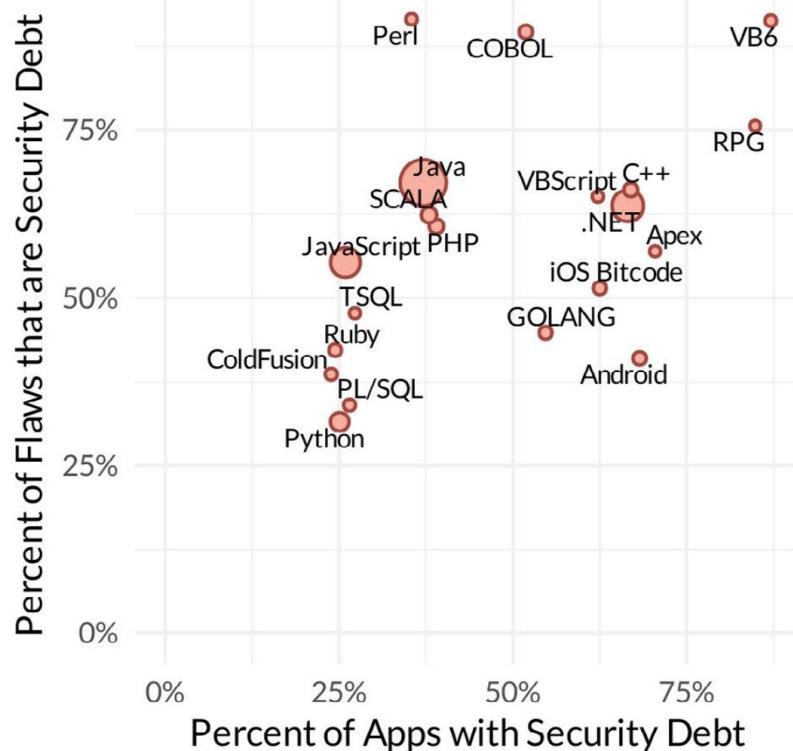


- Alter: Younger (1 - 2,1 Jahre), Middle-aged (2,1 – 3,4 Jahre), Older (> 3,4 Jahre)
- Größe: Small (<250 kB), Medium ( 250 kB – 1,55 MB), Large (> 1,55 MB) der Codebase



## Studie: Annual Report on the State of Application Security (2024)

### Einflussfaktoren für Security Dept: Sprache



*Studie: Annual Report on the State of Application Security (2024)*  
*Abhängigkeiten*

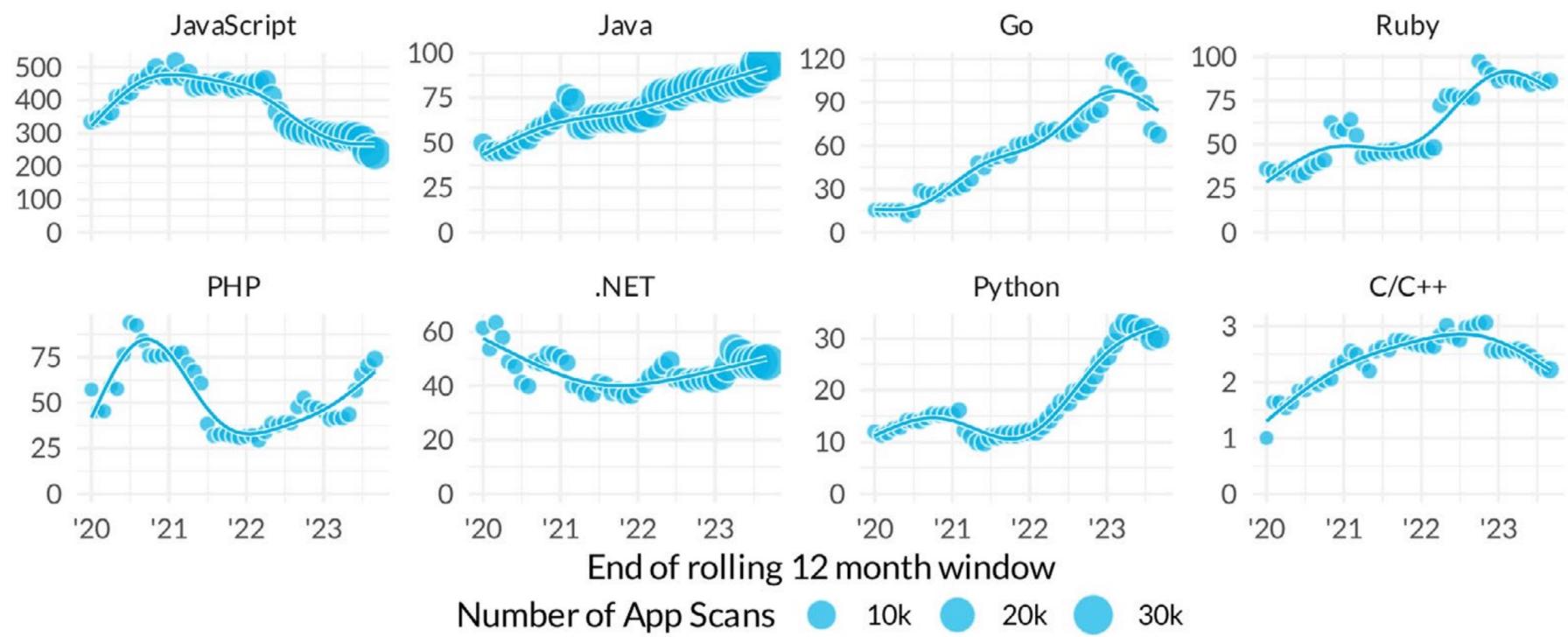


Figure 33: Number of libraries per application over time (rolling geometric mean)

## Studie: Annual Report on the State of Application Security (2024)

### Anteil Third Party Code an der Anwendung

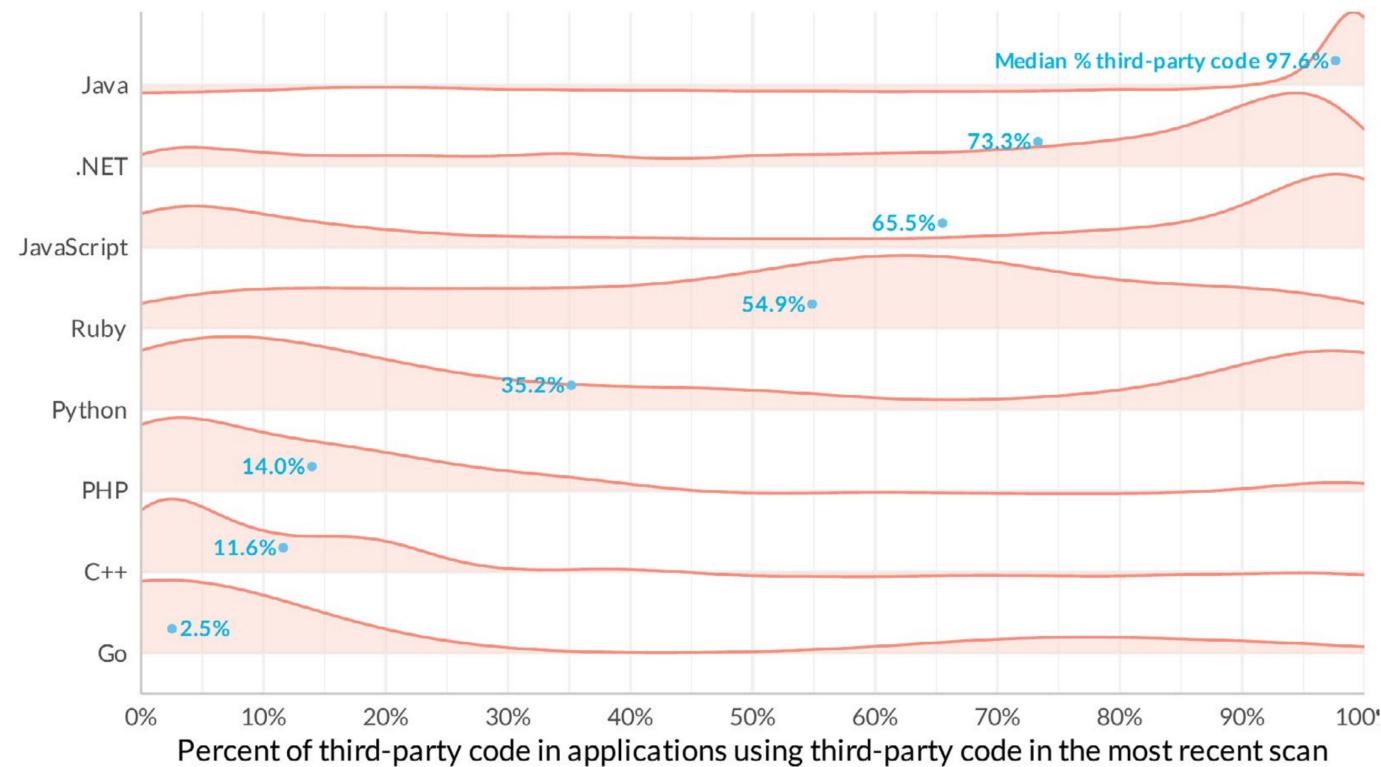
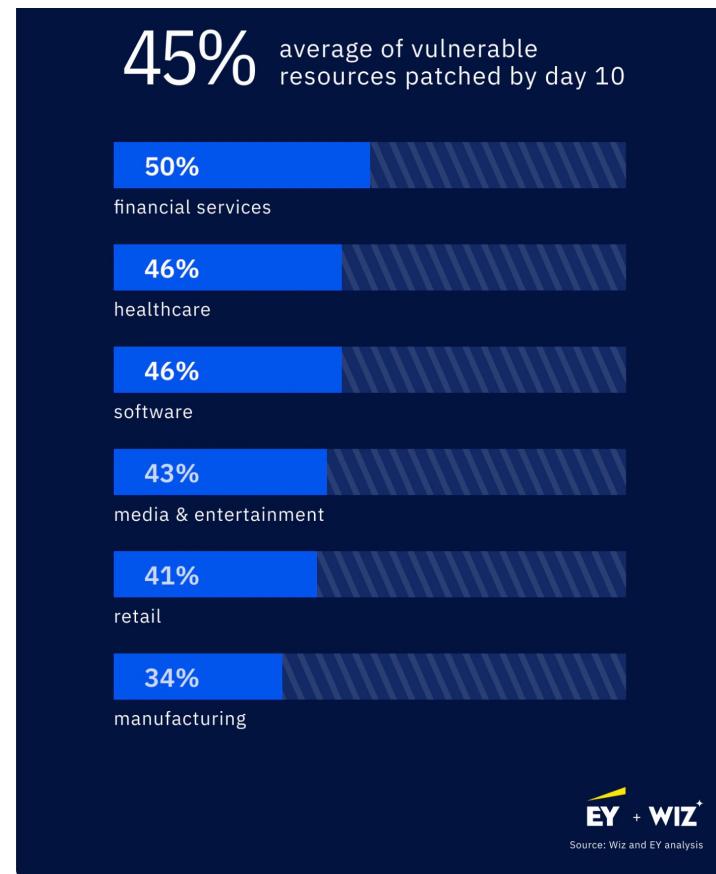
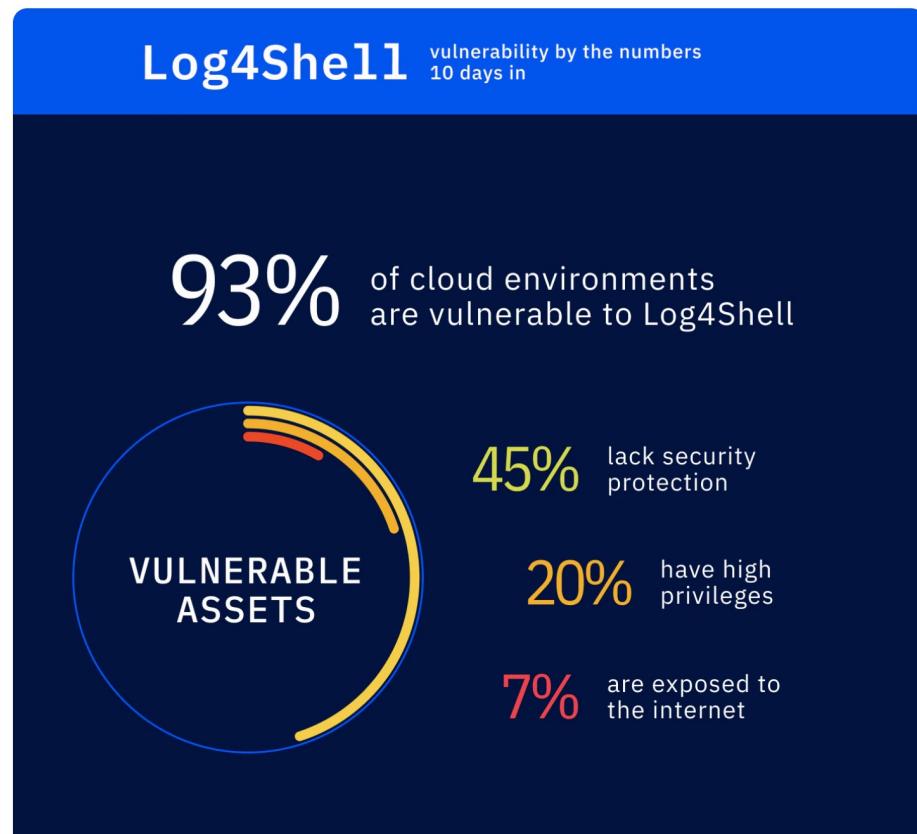


Figure 34: Percentage of an application that is third-party code. The light coral represents the distribution of apps in that language, while the blue points are the median for that language.



## Auswirkungen

Studie EY (2021)

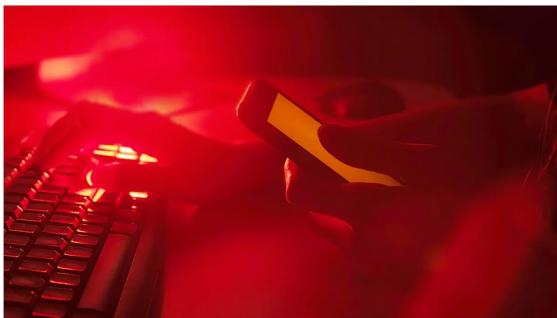


## Log4j: Angriff auf Netz des belgischen Verteidigungsministeriums

Nach einem Angriff über die schwere Sicherheitslücke in der Java-Bibliothek Log4j musste das belgische Verteidigungsministerium Teile seines Netzes abschalten.

Lesezeit: 2 Min.  In Pocket speichern

46



(Bild: Katya Rekina/Shutterstock.com)

20.12.2021 18:12 Uhr

## Juniper Networks stopft zahlreiche Sicherheitslücken

In Geräten und Diensten von Juniper hätten Angreifer Schwachstellen etwa für DoS-Angriffe, die Ausweitung von Rechten oder Schlimmeres missbrauchen können.

Lesezeit: 2 Min.  In Pocket speichern

4



(Bild: asharkyu/Shutterstock.com)

14.01.2022 15:29 Uhr | Security

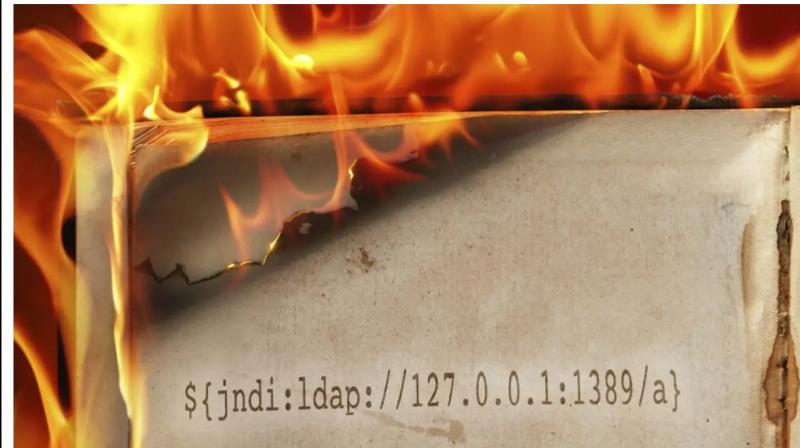
Von Dirk Knop

## Sicherheitslücke Log4Shell: Internet in Flammen

Die Zero-Day-Sicherheitslücke Log4Shell war zu leicht auszunutzen. Das Ausmaß lässt sich noch immer nicht abschätzen.

Lesezeit: 10 Min.  In Pocket speichern

18



(Bild: Composing | Quelle: Misha - stock.adobe.com)

31.12.2021 06:00 Uhr | c't Magazin

Von Mirko Dölle

Security Testing

## Erpressergruppe Conti nutzt Sicherheitslücke "Log4Shell" für ihre Ransomware

Der Erpressungstrojaner der bekannten Conti-Gang wird bereits auf die Lücke "Log4Shell" losgelassen. Damit wächst das Bedrohungspotenzial deutlich.

Lesezeit: 2 Min.  In Pocket speichern

55



(Bild: Sashkin/Shutterstock.com)

19.12.2021 16:31 Uhr | Security

Von Tilman Wittenhorst

## Webseite des Bundesfinanzhofs nach Log4j-Angriff offline

Aufgrund eines Angriffs auf die Log4Shell-Schwachstelle haben die Behörden die Webseite abgeschaltet. Das interne Netz sei jedoch nicht betroffen.

Lesezeit: 1 Min.  In Pocket speichern

29



(Bild: solarseven/Shutterstock.com)

17.12.2021 14:10 Uhr | Security