



Technische Hochschule  
Ingolstadt  
Fakultät Informatik

# *Software-Sicherheit & Security Testing Kapitel 2: Grundlagen*

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

## *Ziele der heutigen Veranstaltung*



- **Studierende kennen grundlegende Begriffe der Softwaresicherheit**
  
- **Studierende wissen, wie Security-Probleme in Software entstehen**



**Definition:** Als Software werden ausführbare, nicht-physische Teile von Computern bezeichnet. Man kann die Menge von Software auf einem Computer unterscheiden in Systemsoftware und Anwendungssoftware.

**Definition:** Systemsoftware ist die Software, die zum Betrieb des Computers notwendig ist. Sie stellt insbesondere die Verbindung zwischen Hardware und Software her und ermöglicht die Ausführung von Anwendungssoftware.

**Definition:** Anwendungssoftware ist Software, die nicht zur Systemsoftware gehört und es Benutzern erlaubt, eigene Aufgaben auf dem Computer zu lösen.

Entsprechend unterscheidet man in der Softwaresicherheit auch oft **Application Security** und **System Security**

## *Grundlegende Begriffe*

### *Definition Security-Mangel*



**Definition:** Als Mangel (engl. **flaw**) bezeichnet man in einer Software einen Fehler oder eine Auslassung, der/die dazu führt, dass sich die Software nicht wie beabsichtigt und gewünscht verhält.

**Definition:** Ein Security Mangel (engl. **security flaw**) ist ein Mangel in einer Software, der grundsätzlich geeignet ist, die an eine Software gestellten Sicherheitsziele zu verletzen.

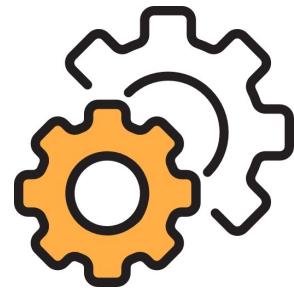
**Definition:** Eine Verwundbarkeit (engl. **vulnerability**) ist eine ungewünschte, durch einen Angreifer ausnutzbare Eigenschaft einer Software, welche das Risiko einer unerlaubten Nutzung, Datenabfluss oder Manipulation erzeugt. Eine Verwundbarkeit ist also ein durch einen Angreifer ausnutzbarer **Security Mangel**.



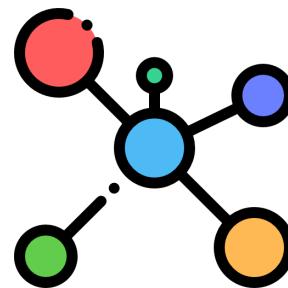
## Häufige Arten von Security-Mängeln



Programmierfehler  
Source Code



Mangelhafte  
Konfiguration



Abhängigkeit von mit  
Mängeln behafteten  
externen Komponenten



Mangelhaftes Design

## *Grundlegende Begriffe*

### *Definition Angriffsfläche*



**Definition:** Als Angriffsfläche (engl. attack surface) einer Software-Komponente oder einer Software als ganzes bezeichnet man die Menge der Zugangspunkte, die ein Angreifer nutzen kann, um die Software oder die Software-Komponente anzugreifen oder Daten abzugreifen.

- Es gibt weitere Angriffsflächen, z.B.
  - die physikalische Angriffsfläche bezeichnet alle Teile eines Systems, auf die ein Angreifer physikalischen Zugang hat.
  - die menschliche Angriffsfläche bezeichnet alle Individuen eines Systems, welche durch Social Engineering Angriffe erreicht werden können.
- Besonders wichtige Aufgabe: Identifikation der Angriffsfläche – dort verstärkte Aufmerksamkeit



## Angriffsfläche

### *Typische Bestandteile*

- User Interface in allen Formen
- Verarbeitungsfunktionen von HTTP Headers und Cookies in Webanwendungen
- APIs
- Dateien, die Verarbeitet werden (z.B. auch Konfigurationsdateien)
- Interface zu einer externen Datenbank
- Interface von Funktionen, Modulen, Komponente, ... (zur Laufzeit übergebene Argumente)

- Werkzeuge zur Identifizierung der Angriffsüberfläche werden Enumeration Tools genannt
- Beispiele:
  - Nmap (Netzwerke): Scannt in Netzwerken, Analyse von offenen Ports auf Servern, auch Analyse von Webseiten
  - Bloodhound (Active Directory): Analysiert Active Directory Objekten
  - PowerView (Active Directory) : Analyse von Active Directory Objekten
  - Dirbuster (WWW): Findet Verzeichnisse auf Webservern (Wordlist)
  - hakrawler (WWW): Crawler zur Auflistung von Webseiten



## Angriffsoberfläche

Beispiel nmap mit http-enum Skript

```
(kali㉿kali)-[~]
└─$ nmap -script=http-enum www.██████████
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-04 04:32 EDT
Nmap scan report for www.██████████ (194.██████████)
Host is up (0.0017s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
113/tcp   closed ident
443/tcp   open  https
| http-enum:
|   /robots.txt: Robots file
|   /typo3/index.php: Typo3 Installation
|_  /uploads/: Potentially interesting folder
2000/tcp  open  cisco-sccp
5060/tcp  open  sip

Nmap done: 1 IP address (1 host up) scanned in 36.65 seconds
```

# Angriffsfläche

## Beispiel dirbuster

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

https://prakharprasad.com:443/

(Scan Information) Results - List View: Dirs: 59 Files: 58 \ Results - Tree View \ Errors: 57 \

Type	Found	Response	Size
Dir	/	200	9246
Dir	/about/	200	664
Dir	/rss/	200	663
Dir	/login/	302	670
Dir	/security/	200	664
File	/cdn-cgi/l/email-protection	200	381
File	/rss	200	595
File	/rss/index.php	302	644
File	/rss/images.php	302	645
Dir	/rss/images/	302	668
Dir	/rss/index/	302	668
Dir	/rss/download/	302	668
File	/rss/2006.php	302	643
Dir	/rss/2006/	302	670

Current speed: 1 requests/sec (Select and right click for more options)

Average speed: (T) 10, (C) 1 requests/sec

Parse Queue Size: 0 Current number of running threads: 20

Total Requests: 2006/9795735 Change

Time To Finish: 113 Days

Back Pause Stop Report

Program paused! /rss/09/privacy.php

Bildquelle: <https://subscription.packtpub.com/book/security/9781785284588/2/ch02lvl1sec20/dirbuster>



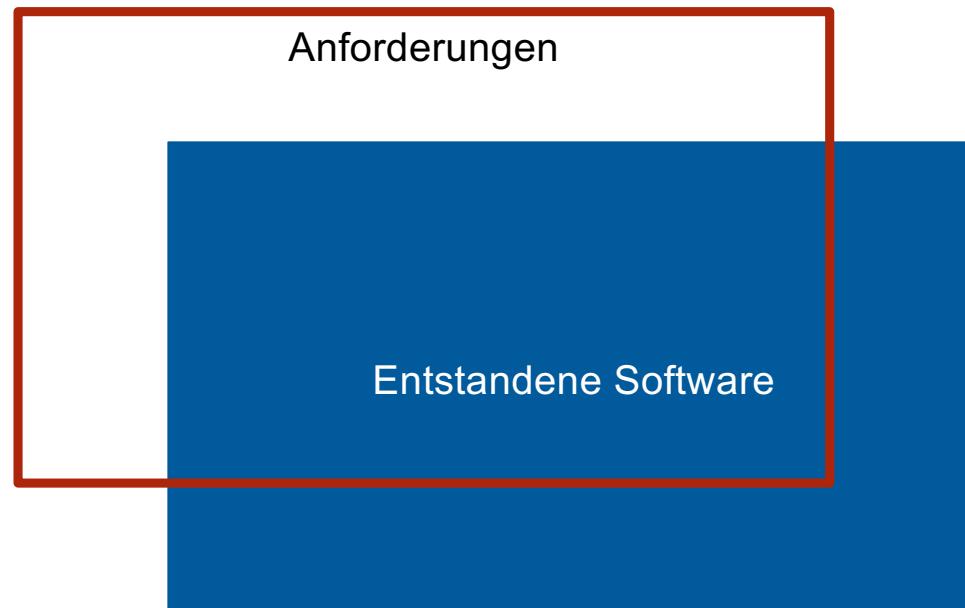
## Angriffsfläche

*Identifikation während der Entwicklung*

- Wie können Sie die Angriffsfläche während dem Design der Software identifizieren?

## Angriffsfläche

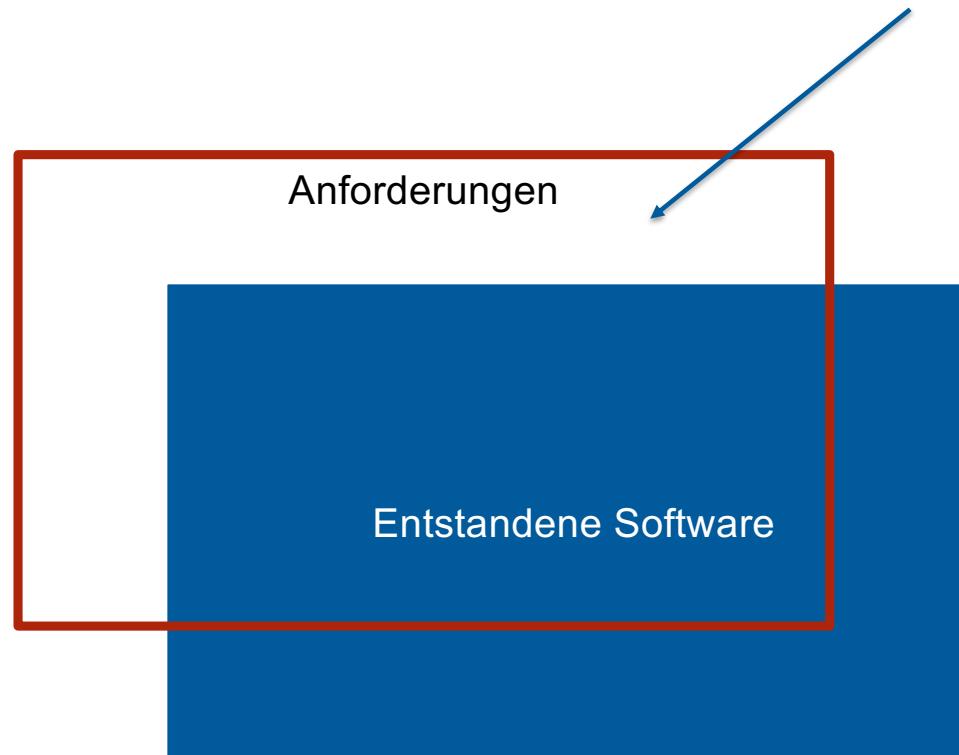
Oft unspezifizierter Teil von Software



## Angriffsfläche

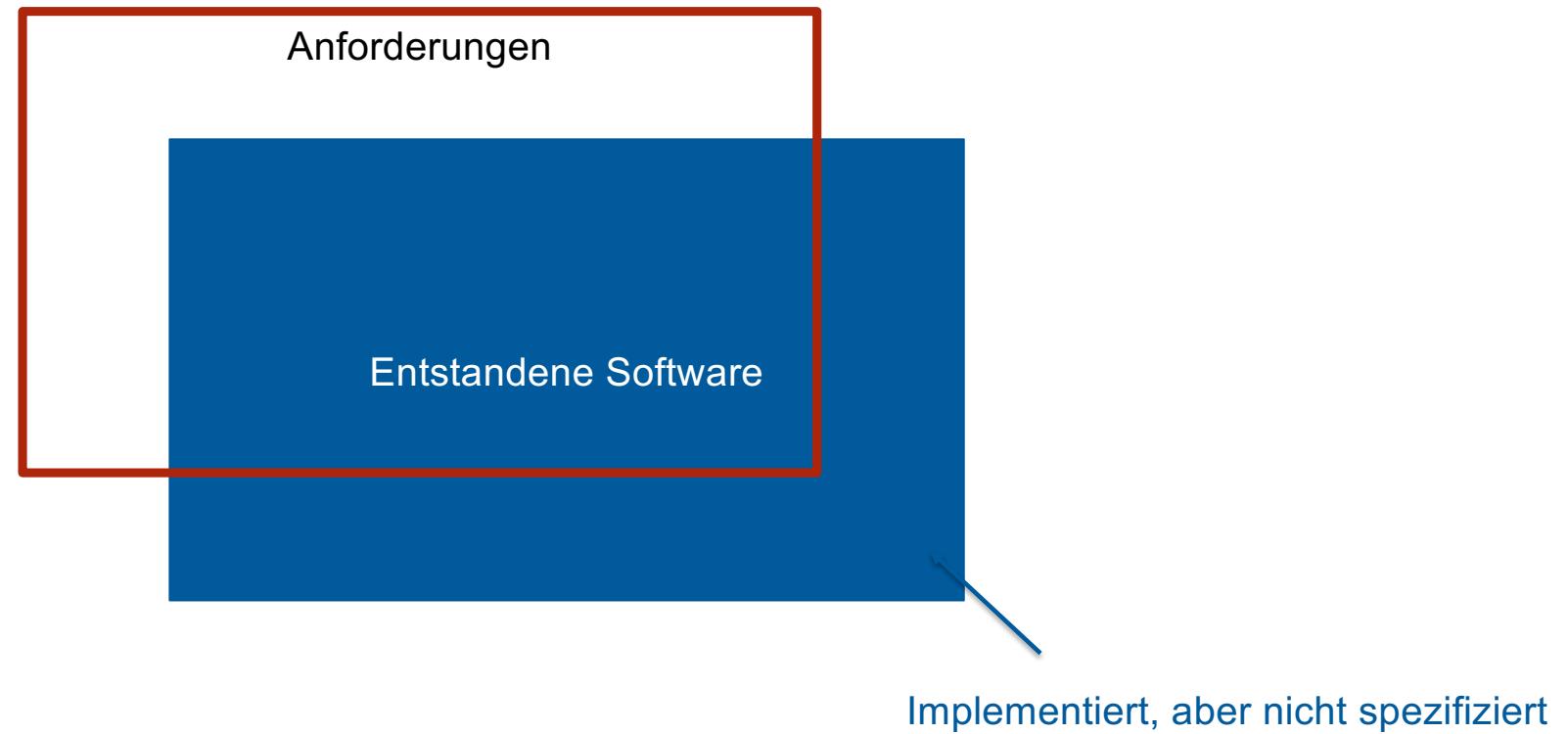
Oft unspezifizierter Teil von Software

Spezifiziert, aber nicht umgesetzt



## Angriffsfläche

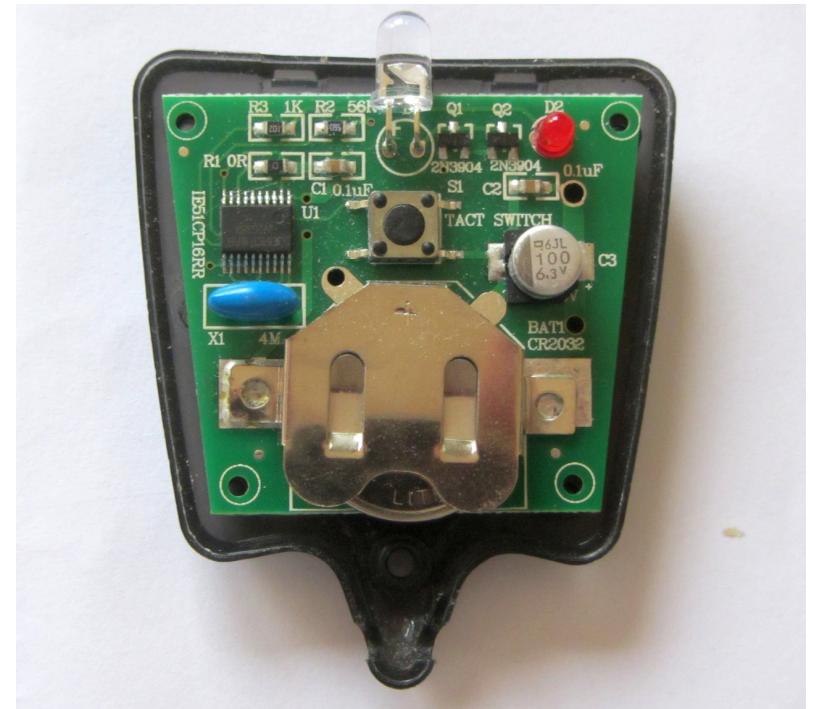
Oft unspezifizierter Teil von Software



## Physikalische Angriffsfläche

### Identifikation

- Identifikation von
  - Wartungsklappen
  - verschraubten Abdeckungen
  - offenen oder versteckten USB Ports
  - JTAG-Schnittstelle
  - andere Schnittstellen
- Nicht Fokus dieser Vorlesung, bei Interesse, z.B.  
[1] als Beispiel zum Vorgehen



TV-B-Gone  
(Bildquelle Wikipedia)

# Menschliche Angriffssoberfläche

## Werkzeuge

- Zur Enumeration der Menschlichen Angriffssoberfläche eignen sich besonders berufliche soziale Netzwerke
- Nicht Fokus dieser Vorlesung



technische hochschule ingolstadt

◀ Previous

1 **2** 3 4 5 6 7 8 ... 100

Next ▶

- [REDACTED]  
Laboringenieurin bei Technische Hochschule Ingolstadt  
Ingolstadt  
Current: Laboringenieurin für [REDACTED] at Technische Hochschule  
Ingolstadt  
[REDACTED]
- 
- Prof. Dr. [REDACTED]  
CEO ICE  
Greater Munich Metropolitan Area  
Current: Professor at Technische Hochschule Ingolstadt  
[REDACTED]
- 
- [REDACTED]  
Leiter Technologietransfer & Internationale Projekte [REDACTED]  
Ingolstadt  
Current: Leiter Technologietransfer & Internationale Projekte at Technische Hochschule Ingolstadt  
[REDACTED]

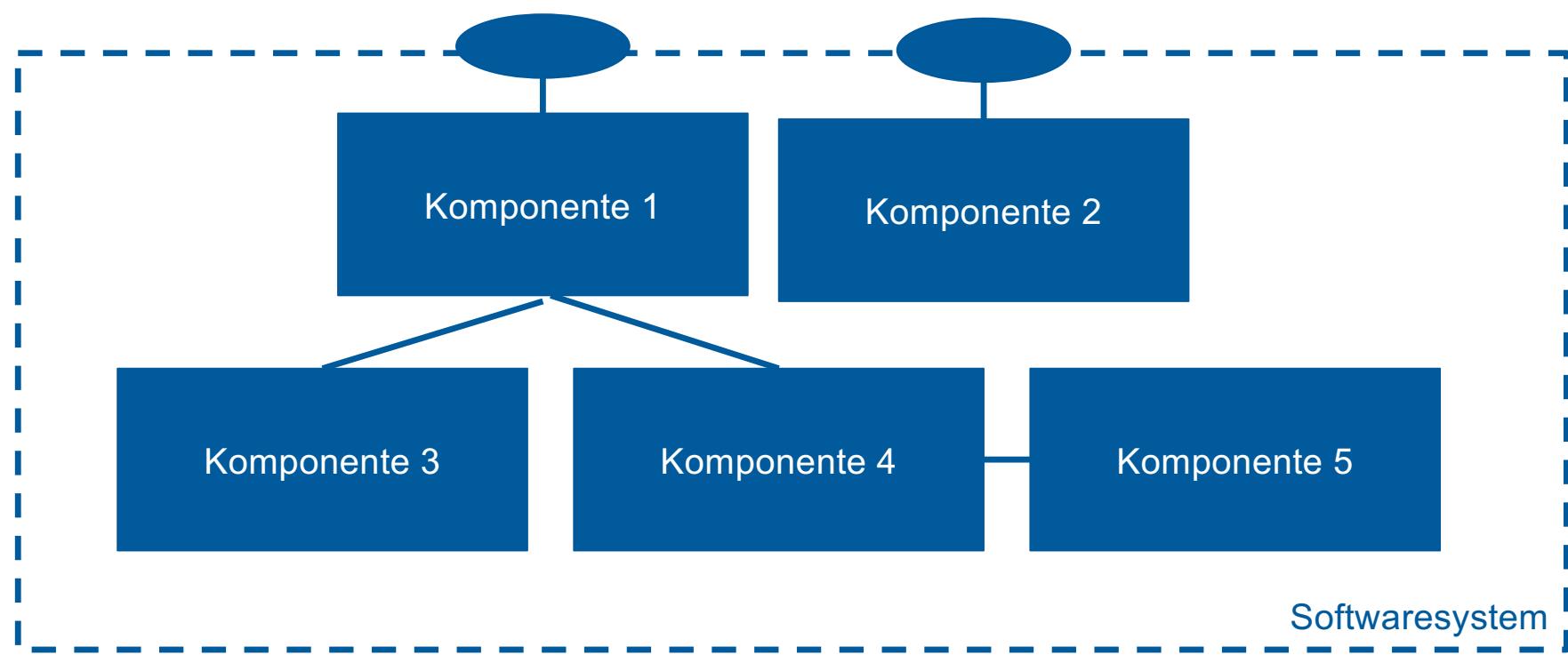
## Grundlegende Begriffe

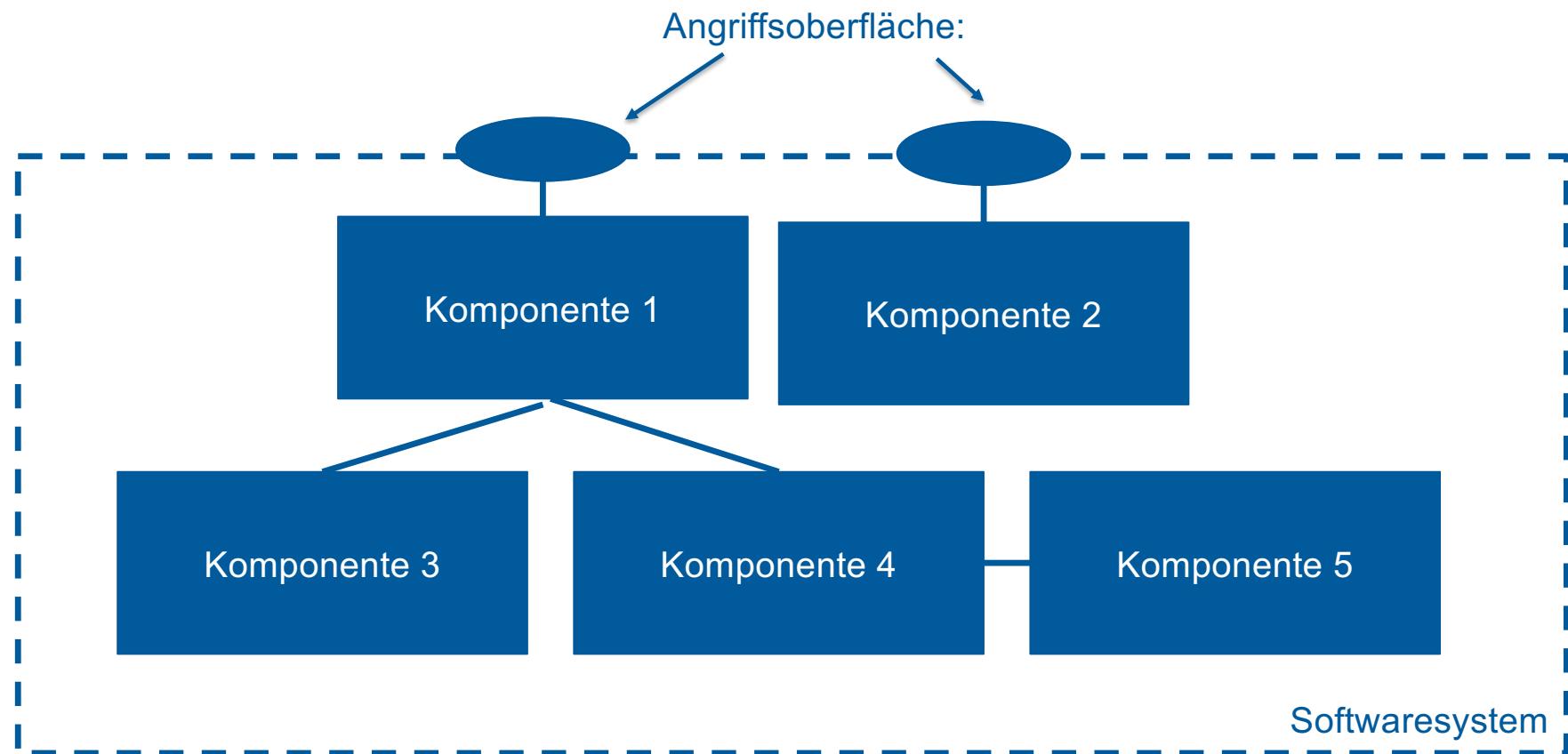
### Definition Angriff

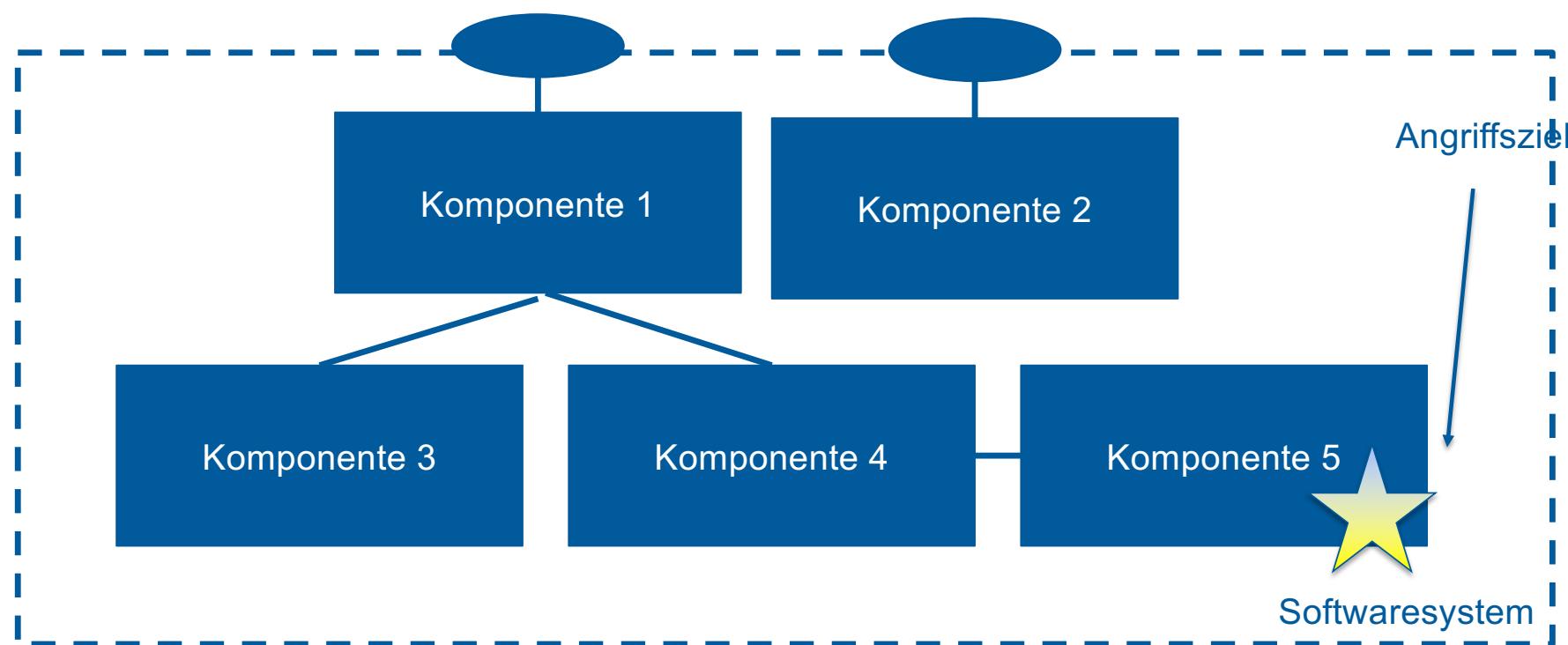


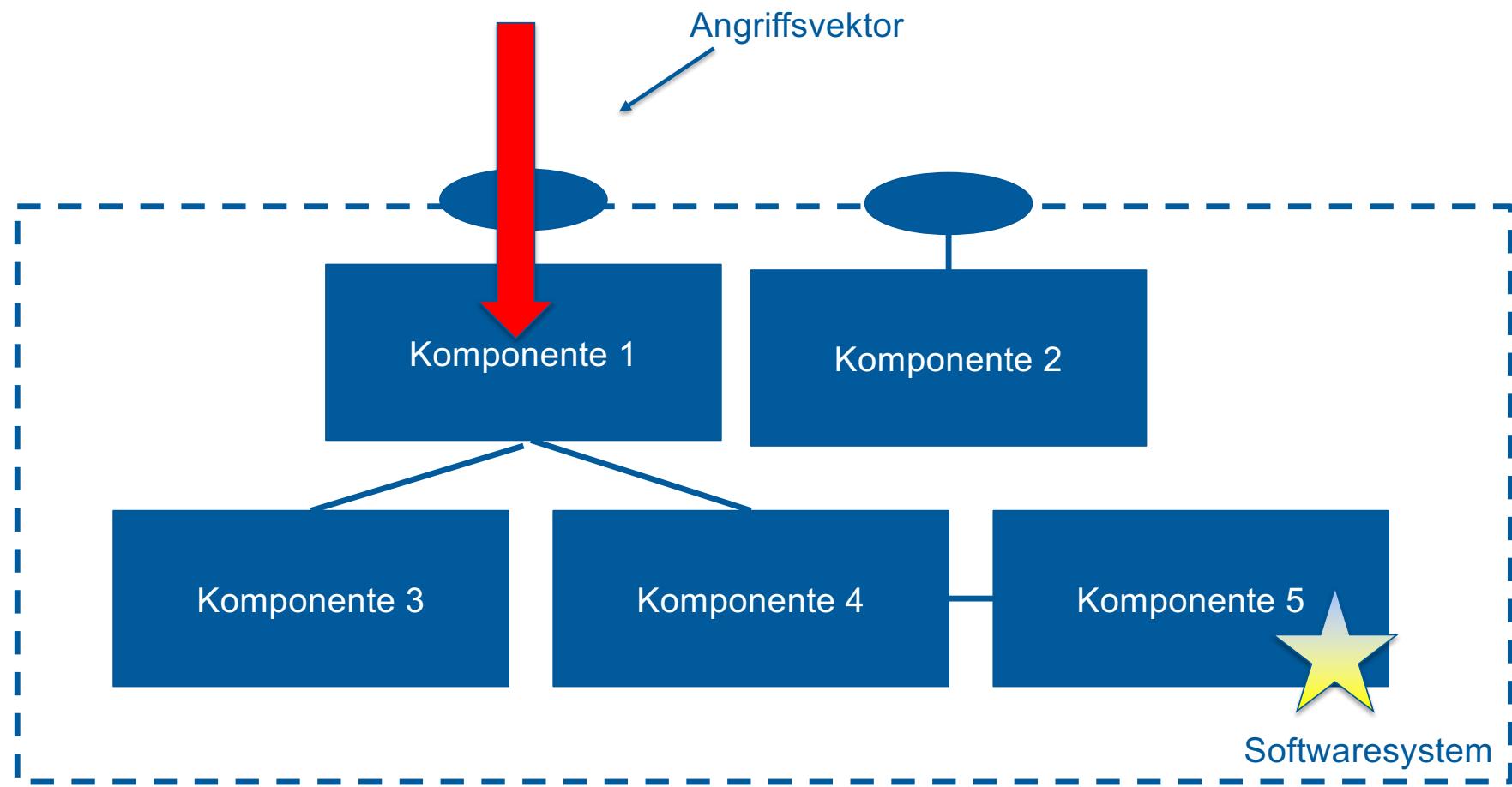
**Definition:** Ein Angriffsvektor (engl. *attack vector*) ist eine Methode, die ein Angreifer einsetzt, um von der Angriffsfläche ins System einzudringen. Ein Attack Vector stellt also den Eintrittspunkt in ein System dar.

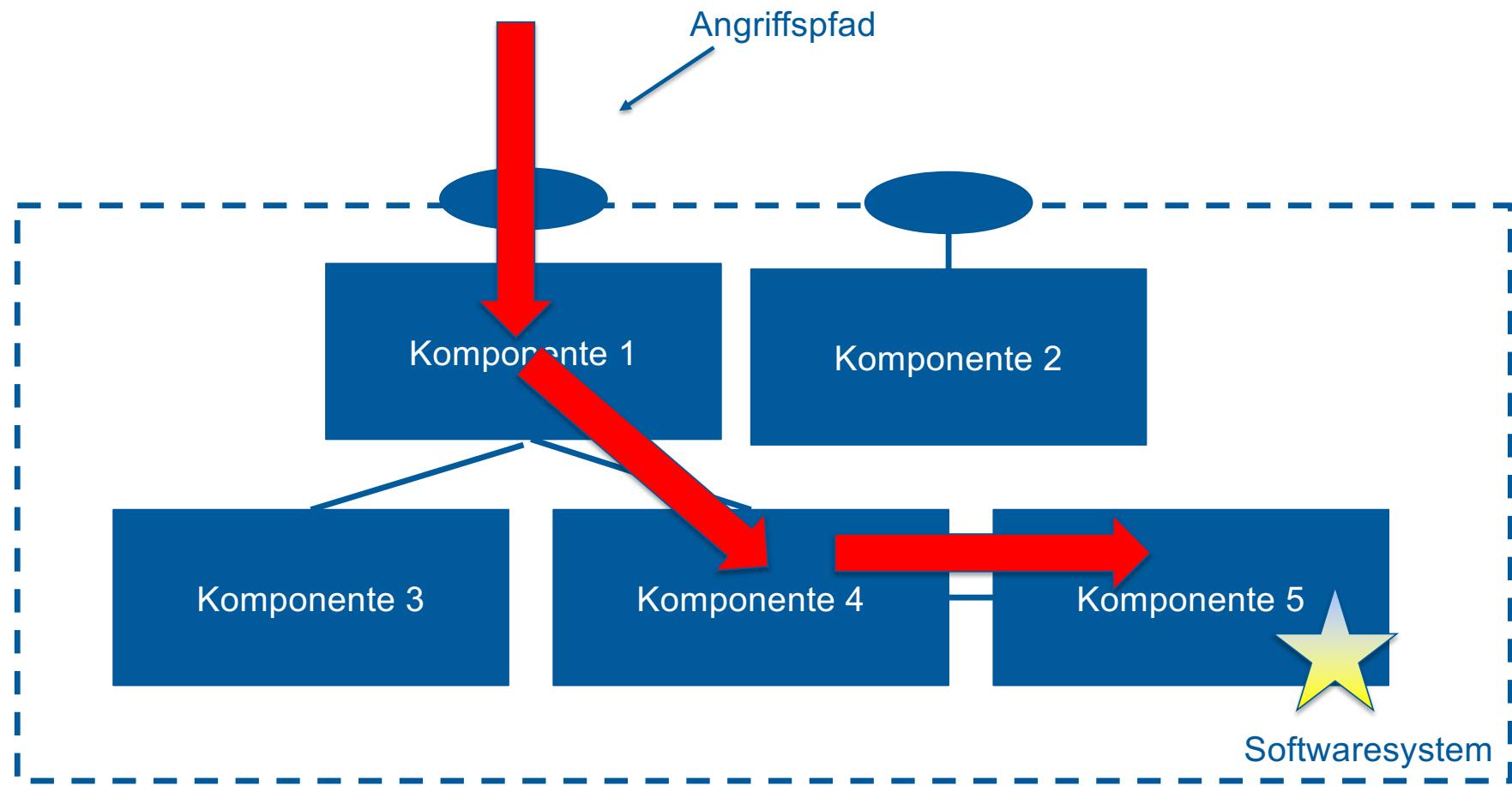
**Definition:** Ein Angriffspfad (engl. *attack path*) ist der Pfad, den ein Angreifer während eines Angriffs von der Angriffsfläche bis zum Angriffsziel nimmt. Der Pfad kann innerhalb einer Software über mehrere Software-Komponenten führen. Der Angriffspfad wird in der Überwachung oft sichtbar über die Ereignisse, die auftreten, wenn ein Angreifer einen Angriffsvektor für einen Angriff ausnutzt.













**Was ist die Cyber-Kill-Chain?**

**Bildet die Cyber-Kill-Chain einen Angriffsvektor oder einen Angriffspfad ab? Welcher Teil davon?**

**Was ist das MITRE ATT&CK Framework?**



**Definition: Als Exploit bezeichnet man**

- (a) Software oder Daten, mit denen eine Verwundbarkeit erfolgreich ausgenutzt wird.
- (b) die Beschreibung des Vorgehens zur Ausnutzung einer Verwundbarkeit

**Wird ein Exploit an der Angriffsfläche eingesetzt, so handelt es sich um die technische Realisierung eines Angriffsvektors.**

**Definition: Als Zero-Day-Exploit bezeichnet man einen Exploit, der eine bisher in der Öffentlichkeit unbekannte Schwachstelle ausnutzt.**



- Eine Anwendung nimmt Benutzerdaten entgegen und erzeugt daraus eine SQL-Anfrage

```
# Benutzername in der Variable name  
sql_query = "Select * from users where user_name=\\""+name+"\\";"
```

- Benutzerinput Achim

```
sql_query="Select * from users where user_name='Achim';"
```

- Exploit: Benutzerinput ' OR '1'='1

```
sql_query="Select * from users where user_name=' OR '1'='1';"
```

- **Definition:** Shellcode bezeichnet Binary Code, der zur Manipulation von Systemen geeignet ist und üblicherweise dem Angriffsziel als Input übergeben wird.
- **Shellcode ist die Nutzlast, die in einem Exploit übergeben wird.**
- Je nachdem wie die Daten verarbeitet werden, gibt es bei Shellcode verschiedene Einschränkungen, z.B.
  - Längenbeschränkung
  - Keine Null-Bytes erlaubt
  - Frage: Warum?



## Beispiel Shellcode

### Historisches Beispiel

#### ■ Älteres Beispiel (i386, Linux):

```
char shellcode[] = "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

#### ■ Disassembled:

```
jmp 0x1f
popl %esi
movl %esi,0x8(%esi)
xorl %eax,%eax
movb %eax,0x7(%esi)
movl %eax,0xc(%esi)
movb $0xb,%al
movl %esi,%ebx
leal 0x8(%esi),%ecx
leal 0xc(%esi),%edx
int $0x80
xorl %ebx,%ebx
movl %ebx,%eax
inc %eax
int $0x80
call -0x24
.string \"/bin/sh\"
```

Quelle: [2]

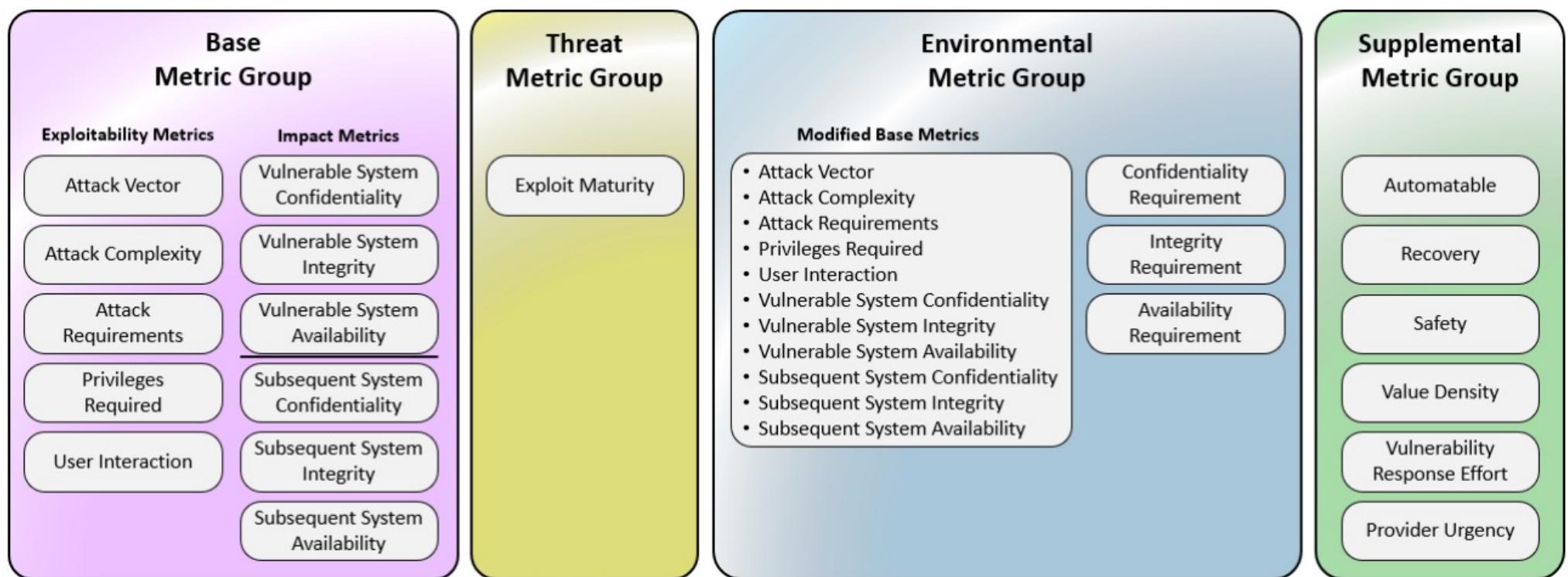
## *Common Vulnerability Scoring System (CVSS)*

*Punktesystem für die Bewertung von Verwundbarkeit*



- **Frage: Wozu Bewertung von Verwundbarkeiten? Warum einheitliche Bewertung angestrebt?**
- **02/2005 wurde Version 1 veröffentlicht, nach viel Kritik am Score entwickelt seit 04/2005 das Forum of Incident Response and Security Teams (FIRST) CVSS weiter.**
- **Im Jahr 2022 wurde CVSS Version 4.0 für Public Review veröffentlicht siehe [3]**





Bildquelle: [3]



- **Spiegeln den Schweregrad einer Verwundbarkeit wieder**
- **Beziehen sich auf über die Zeit konstante, intrinsische Merkmale**
  - Dynamische Merkmale werden dann in den anderen Metrik-Gruppen betrachtet
- **Geht von den schlimmsten realistisch zu erwartenden Auswirkungen in verschiedenen Umgebungen aus**
- **Zwei Gruppen innerhalb der Basis Metriken**
  - Ausnutzbarkeit
  - Auswirkung



- **Attack Vector (AV): von wo ist der Angriff möglich, mögliche Werte:**
  - Network(N): kann aus der Ferne ausgenutzt werden, üblicherweise aus einem WAN/Internet (mehr als A, L, P)
  - Adjacent (A): lokaler Zugang, z.B. über Tastatur, Konsole, oder Remote über SSH
  - Physical (P): Physikalischer Zugriff zum verwundbaren System ist notwendig
- **Attack Complexity (AC): wie schwer ist es, Sicherheitsmaßnahmen für einen Exploit zu umgehen:**
  - Low (L): Keine Umgehung notwendig
  - High (H): Umgehung notwendig, z.B. Address Space Randomization (ASLR) oder Data Execution Prevention (DEP) oder der Exploit benötigt ein für das Opfersystem spezifisches Geheimnis

## Base Metriken

*Exploitability Metrics: Eigenschaften des Angriffs auf die Verwendbarkeit*



- **Attack Requirements (AT): Welche Vorbedingungen gibt es, dass der Exploit im verwundbaren System funktioniert? Hier sind die Security-Maßnahmen nicht gemeint sondern andere Systemeigenschaften**
  - None (N): Keine besonderen Bedingungen an Umgebung, Exploit funktioniert immer
  - Present (P): Spezielle Eigenschaften müssen hergestellt werden, z.B. eine Race Condition oder ein Angreifer muss sich auf dem Netzwerkpfad befinden für einen Man-in-the-Middle Angriff
- **Privileges Required (PR): Welchen Rechtelevel muss ein Angreifer vor dem Angriff besitzen, damit der Exploit durchgeführt werden kann?**
  - None (N): Exploit funktioniert auch bei unautorisierten Benutzern
  - Low (L): Unpriviligerter Benutzer
  - High (H): Priviligerter Benutzer (z.B. Admin)

## Base Metriken



*Exploitability Metrics: Eigenschaften des Angriffs auf die Verwendbarkeit*

- **User Interaction (UI): Braucht der Angriff eine Interaktion mit einem weiteren Benutzer?**
  - None(N): Kein weiterer Benutzer notwendig
  - Passive (P): Benutzer notwendig, er umgeht aber keine Schutzmechanismen
  - Active (A): Benutzer interagiert bewusst mit dem verwundbaren System und dem Payload des Angreifers oder der Benutzer umgeht aktiv Schutzmechanismen



## Base Metriken

*Impact Metrics: Auswirkungen eines erfolgreichen Angriffs*

- **Confidentiality (VC/SC)**
  - Confidentiality Impact to the Vulnerable System (VC)
    - High (H): Totalverlust der Vertraulichkeit oder Verlust der Vertraulichkeit einer Gruppe von hochgeheimen Daten
    - Low (L): Vertraulichkeit weniger Daten betroffen oder Angreifer kann die Daten, die bekannt werden, nicht aussuchen
    - None (N): Kein Verlust der Vertraulichkeit
  - Confidentiality Impact to the Subsequent System (SC)
    - High (H): Wie oben, aber für Folgesystem
    - Low (L): Wie oben, aber für Folgesystem
    - Negligible (N): Kein Vertrauensverlust im Folgesystem (aber eventuell im verwundbaren System)



## Base Metriken

*Impact Metrics: Auswirkungen eines erfolgreichen Angriffs*

- **Integrity (VI/SI): Definition für Vulnerable System und Subsequent System analog zu Confidentiality (VC/SC)**
  
- **Availability (VA/SA): Definition für Vulnerable System und Subsequent System analog zu Confidentiality (VC/SC)**



- Spiegeln die Merkmale einer Schwachstelle im Bezug auf die Bedrohungslage wieder
- Kann sich über die Zeit ändern
- Beispiel: wenn eine Verwundbarkeit weder in der Praxis ausgenutzt wird noch es eine Proof of Concept Implementierung oder eine Beschreibung eines Angriffs gibt, führt zu einem niedrigeren CVSS Wert.

- **Exploit Maturity: Wie weit fortgeschritten ist die Entwicklung von Exploits und wie bekannt sind die Exploits?**

- Not Defined (X): Metrik wird nicht benutzt
- Attacked (A):
  - Threat Intelligence ergibt, dass es Angriffe auf diese Verwundbarkeiten gibt oder
  - Es existieren Programme zur Vereinfachung von Exploits (z.B. Exploit Toolkits)
- Proof-of-Concept (P):
  - Es gibt einen Proof-of-Concept eines Exploits, der die Verwundbarkeit ausnutzt und
  - Threat Intelligence ergibt, dass es noch keine Angriffe auf diese Verwundbarkeiten gibt und
  - Es existieren noch keine Programme zur Vereinfachung von Exploits (z.B. Exploit Toolkits)
- Unreported (U):
  - Kein Proof-of-Concept und
  - Keine Angriffe und
  - Keine Programme zur Vereinfachung

- Spiegeln die Merkmale einer Schwachstelle ab, die speziell für einzelne Umgebungen von Benutzern sind
- Ermöglicht es, den Score z.B. an die Wichtigkeit von IT Assets anzupassen
- Beispiel: Mit den Environmental Metriken kann berücksichtigt werden, falls Security Mechanismen in der Benutzerumgebung existieren, die die Auswirkungen von erfolgreichen Angriffen abmildern. Die Wichtigkeit des verwundbaren Systems in einer komplexen Infrastruktur kann berücksichtigt werden.

- **Confidentiality Requirements (CR)**
  - Not Defined (X): Metrik wird nicht verwendet
  - High (H): Verlust der Vertraulichkeit hat katastrophale Auswirkungen auf die Organisation oder Individuen
  - Medium (M): Verlust der Vertraulichkeit hat ernsthafte Auswirkungen auf die Organisation oder Individuen
  - Low (L): Verlust der Vertraulichkeit hat nur sehr begrenzte Auswirkungen auf die Organisation oder Individuen
- **Integrity Requirements (IR): Analog zu CR**
- **Availability Requirements (AR): Analog zu CR**



## *Metrik. Gruppen*

### *Supplemental Metriken*

- **Zusätzliche Metriken, die von den Benutzern des CVSS Scores berücksichtigt werden können in eigenen Analysen**
- **Die Informationen in diesen Metriken fließen nicht in den CSS Score ein.**
- **Hier nicht näher betrachtet, bei Interesse siehe [3]**



## CVSS Scores

*Darstellung als Vector String*

- Neben dem Score an sich können auch die einzelnen Werte in einem String dargestellt werden, der mit CVSS: und der Versionsnummer von CVSS beginnt und mindestens die vollständigen Base Metrics enthalten muss.
- Beispiel:
  - CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N
  - CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N/E:A

## CVSS Scores

### Verschiedene Scores

- Scores zwischen 0,0 und 10,0

CVSS Nomenclature	CVSS Metrics Used
CVSS-B	Base metrics
CVSS-BE	Base and Environmental metrics
CVSS-BT	Base and Threat metrics
CVSS-BTE	Base, Threat, Environmental metrics

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

- Rechner zur Berechnung der Scores: <https://redhatproductsecurity.github.io/cvss-v4-calculator/>



- Aufgabe: In [3] wird auf den Seiten 33-37 (Kapitel 8.2) die Berechnung des CVSS Scores im Details beschrieben. Erarbeiten Sie sich diese Berechnung.
- Überprüfen Sie, ob der Score in diesem Vector korrekt berechnet ist:
- **CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N**



## Software-Entwicklungszyklus (Software Development Lifecycle)

Recap: Vorgehensmodelle für die Softwareentwicklung

- Welche Vorgehensmodelle für die Softwareentwicklung kennen Sie aus der Vorlesung „Software-Entwicklungsmethodik“?
  
- Aufgabe (20 Minuten): Diskutieren Sie, welche Art von Security-Mängeln in den einzelnen Software-Entwicklungsmethodiken in welchem Entwicklungsstadium entstehen.

## Bekannte Institutionen im Bereich Software Security

### MITRE



- Spinoff aus dem Massachusetts Institute of Technology (MIT)
- MITRE Corporation ist amerikanische Non-Profit-Organization
- Ziele: „We live with the reality of a world in conflict, in which long-term strategic competition will challenge the security of our nation. Our vision of pioneering for a better future helps ensure our nation's role as a leading force for good“ (Jason Providakes, PhD, President and CEO of MITRE)
- Betreibt eine Reihe von staatlich finanzierten Laboren in den USA und international, in Deutschland:
  - Ramstein Airforce Base
  - Stuttgart Patch Barracks (unter anderem auch Hauptquartier NSA, US Streitkräfte, Special Operations für Europa)
  - Wiesbaden



- **MITRE ATT&CK-Matrix**
  - Operative IT-Security, bildet Vorgehen von Angreifern ab, siehe Vorlesung „Grundlagen der IT-Sicherheit“
- **Common Weakness Enumeration (CWE) <https://cwe.mitre.org/>**
  - Eine Gemeinschaftsinitiative, die eine einheitliche Liste von Software- und Hardware-Security Mängeln definiert, die oft zu Verwundbarkeiten führen.
  - Gibt eine Liste der gefährlichen Software-Mängel heraus (2023 CWE Top 25 Most Dangerous Software Weaknesses)
- **Common Vulnerabilities and Exposures (CVE) <https://cve.mitre.org/>**
  - Katalogisiert bekannt gewordene Verwundbarkeiten
  - Man kann z.B. mit der CWE-Nummer eines Software-Mangels nach bekannten Verwundbarkeiten suchen, die auf dieser Schwachstellen basieren



## *Bekannte Institutionen im Bereich Software Security*

SANS Institute

- Schulungsinsititution aus den USA
- Gibt zusammen mit MITRE die Top 25 Liste der häufigsten Security-Mängel heraus.
- Vielfältige Ressourcen zu Software-Sicherheit, teilweise auch kostenlos



- **National Institute of Standards and Technology ist eine Behörde der USA**
- **Vor allem für technische Standardisierung zuständig, z.B. AES, SHA-Familie sowie die Federal Information Processing Standards (FIPS), die für US-Behörden verpflichtend sind**
- **Betreibt unter anderem die National Vulnerability Database**
  - Liste mit veröffentlichten Verwundbarkeiten, gibt unter anderem CVS-Nummer und CVSS Score an

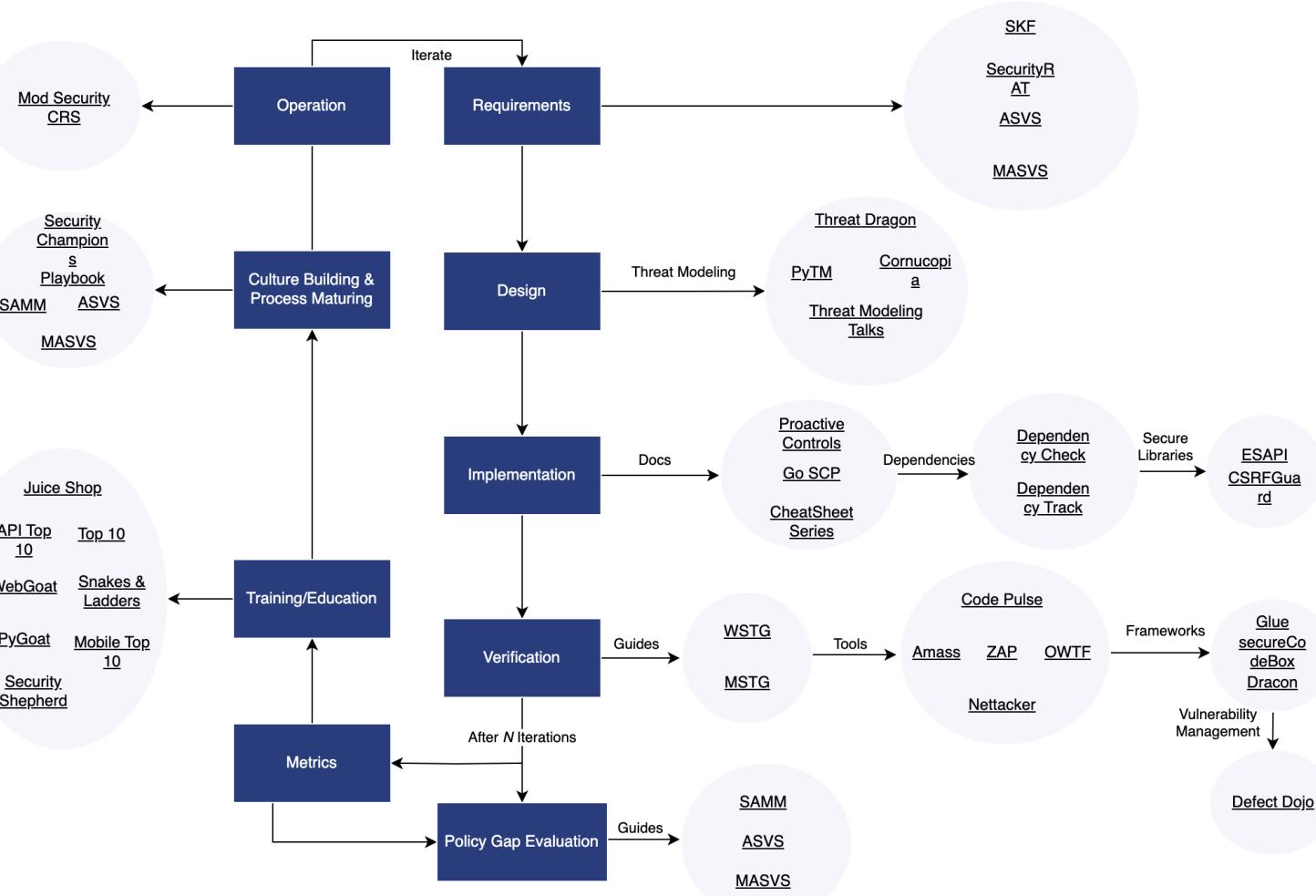
## *Bekannte Institutionen im Bereich Software-Sicherheit*

OWASP



- **Open Worldwide Application Security Project (OWASP), ursprünglich Open Web Application Security Project – [www.owasp.org](http://www.owasp.org)**
- **Gemeinnützige Stiftung, die Software-Sicherheit verbessern wollen**
- **Eine Vielzahl von Community-Projekten im Bereich Softwaresicherheit (10/2023: 302 Projekte)**

## Überblick über Software Security Projekte



[1] SEC Consult, „Reverse Engineering Architecture And Pinout of Custom Asics“, <https://sec-consult.com/blog/detail/reverse-engineering-architecture-pinout-plc/>

[2] Phrack 49, File 14 of 16, „Smashing the Stack For Fun And Profit“

<https://web.archive.org/web/20080211101739/http://www.phrack.org/archives/49/P49-14>

[3] FIRST, „CVSS - Common Vulnerability Scoring System version 4.0 – Specification Document“,

<https://www.first.org/cvss/v4-0/cvss-v40-specification.pdf>