Von der Bedeutung von Schwachstellen

CVSS, CVE, CWE, CVD, KEV, EPSS, VEP

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de
Version: 1.4.0.1

Folien:
HTML
https://delors.github.io/sec-cvss-cve-vep/folien.de.md.html

PDF
https://delors.github.io/sec-cvss-cve-vep/folien.de.md.html.pdf

Fehler melden: https://github.com/Delors/delors.github.io/issues

KI Verwendung: Bei der Erstellung der Unterlagen wurden KI Assistenten (insbesondere Claude aber ggf. auch ChatGPT, Ollama mit Gemma/LLama/Qwen oder OpenCode mit Kimi/Qwen/Deepseek...) unterstützend eingesetzt. Dies erfolgte insbesondere zur Unterstützung bei der Generierung von Grafiken (d. h. SVG Dateien), oder um sich Übersichtstabellen generieren zu lassen. Weiterhin wurde KI zur allgemeinen Qualitätssicherung eingesetzt. Inhalte, die ggf. von der KI vorgeschlagen wurden, wurden im Falle der Übernahme explizit validiert und angepasst.

Inhalt

- Common Vulnerability Scoring System (CVSS)
- Common Vulnerabilities and Exposures (CVE)
- National Vulnerability Database (NVD)
- Exploit Prediction Scoring System (EPSS)
- Schwachstellenmanagement (Vulnerability-Equity-Process (VEP))

1. Common Vulnerability Scoring System (CVSS)

Definition von Schwachstellen nach CVE

> "Eine Schwachstelle in der Berechnungslogik (z. B. Code), die in Software- und Hardwarekomponenten gefunden wird und die, wenn sie ausgenutzt wird, zu einer negativen Auswirkung auf die Vertraulichkeit, Integrität oder Verfügbarkeit führt. Die Behebung der Schwachstellen in diesem Zusammenhang umfasst in der Regel Änderungen am Code, kann aber auch Änderungen an der Spezifikation oder sogar die Ablehnung der Spezifikation (z. B. die vollständige Entfernung der betroffenen Protokolle oder Funktionen) beinhalten."
>
> —https://nvd.nist.gov/vuln (Übersetzt mit DeepL)

Unterscheidung von Schwachstellen:

0-Day: Eine Schwachstelle, die zum Zeitpunkt ihrer Entdeckung noch nicht (öffentlich) bekannt ist und für die es noch keinen (öffentlich verfügbaren) Patch gibt.

n-Day: Eine Schwachstelle, die bereits öffentlich bekannt ist und für die es einen Patch gibt, der (kontextabhängig) jedoch noch nicht angewendet wurde.

> ▲ Achtung!
>
> Auch N-Days sind häufig für Angreifer von Interesse, da es oft eine gewisse Zeit dauert, bis alle Systeme gepatcht sind bzw. überhaupt gepatcht werden (können).

> **Zero-Day Exploits Surge, Nearly 30% of Flaws Attacked Before Disclosure**
>
> [...] shows zero-day and one-day exploitation is accelerating, up from 23.6% in 2024. [...]
>
> —Infosecurity Magazine

Daten von Vulncheck zeigen, dass es in 2025 884 Schwachstellen gab, die ausgenutzt wurden:

- 254 waren *0-Day* Schwachstellen (28.7%)
- 80 waren *1-31-Days* Schwachstellen (9.0%)
- 228 waren *32-365-Days* Schwachstellen (25.8%)
- 92 waren *1-2-Year* Schwachstellen (10.4%)
- 117 waren *2-4-Years* Schwachstellen (13.2%)
- 111 waren *4+ Years* Schwachstellen (12.5%)

Das Common Vulnerability Scoring System (CVSS)[1]

CVSS 4.0 stellt einen Rahmen bereit für die Beschreibung und Bewertung des Schweregrads von Software-/Hardware-/Firmwareschwachstellen.

Die Bewertung der Basiskennzahlen ergibt eine Punktzahl zwischen 0,0 und 10,0. Wobei 0 bedeutet, dass die Schwachstelle (bisher) harmlos ist und 10,0 bedeutet, dass die Schwachstelle sehr gravierend ist.

Harmlos ist im Prinzip damit gleichzusetzen, dass die Schwachstelle nicht ausgenutzt werden kann oder dass die Auswirkungen nicht weiter relevant sind.

---

[1] CVSS 4.0

Gegensätzliche Kräfte bei der Bewertung

Hersteller versuchen tendenziell eine Bewertung zu erhalten, die möglichst niedrig ist.

Sicherheitsforscher versuchen eine möglichst hohe Bewertung zu erhalten.

CVSS umfasst vier Gruppen von Metriken

01 Basis-Metriken (🇺🇸 *Base Metrics*) erfassen die inhärenten Eigenschaften einer Schwachstelle, die sich nicht ändern, wenn sich die Umgebung ändert.

02 Bedrohungs-Metriken (🇺🇸 *Threat Metric Group*) spiegelt die Merkmale einer Schwachstelle wieder, die sich im Laufe der Zeit verändern.

03 Umgebungs-Metriken (🇺🇸 *Environmental Metric Group*) erfassen die Eigenschaften einer Schwachstelle, die sich ändern, wenn sich die Umgebung ändert.

04 Ergänzende-Metriken (🇺🇸 *Supplemental*) liefern zusätzliche Informationen, die für die Bewertung einer Schwachstelle nützlich sein können, aber den Schweregrad nicht direkt beeinflussen.

CVSS - Basis-Metriken (*Base Metric Group*)

Bewertung der Ausnutzbarkeit (*Exploitability Metrics*)

- Angriffsvektor (*Attack Vector*)
- Angriffskomplexität (*Attack Complexity*)
- Angriffsanforderungen (*Attack Requirements*)
- Benötigte Privilegien (*Privileges Required*)
- Erforderliche Benutzerinteraktion (*User Interaction*)

Bewertung der Auswirkungen (*Impact Metrics*)

bzgl. des betroffenen Systems (*Vulnerable System*)

- Vertraulichkeit (*Confidentiality Impact*)
- Integrität (*Integrity Impact*)
- Verfügbarkeit (*Availability Impact*)

bzgl. nachgelagerter Systeme (*Subsequent System*)

- Vertraulichkeit (*Confidentiality Impact*)
- Integrität (*Integrity Impact*)
- Verfügbarkeit (*Availability Impact*)

CVSS - Bedrohungs-Metriken (🇺🇸 *Threat Metric Group*)[2]

- Reifegrad des Exploits (🇺🇸 *Exploit Maturity*)

Gibt es bisher nur die Beschreibung der Schwachstelle oder gibt es bereits einen Proof-of-Concept (PoC) Exploit?

---

[2] Die Namen und der Gruppenzuschnitt (hier: 🇺🇸 *Temporal Metric Group*) waren unter CVSS 3.0 anders.

CVSS - Umgebungs-Metriken

Angepasste Basis-Metriken (🇺🇸 *Modified Base Metrics*)

- Angriffsvektor (🇺🇸 *Attack Vector*)
- Angriffskomplexität (🇺🇸 *Attack Complexity*)
- Angriffsanforderungen (🇺🇸 *Attack Requirements*)
- Benötigte Privilegien (🇺🇸 *Privileges Required*)
- Erforderliche Benutzerinteraktion (🇺🇸 *User Interaction*)

bzgl. des betroffenen Systems **und** auch der nachgelagerten Systeme:

- Vertraulichkeitsverlust (🇺🇸 *Confidentiality Impact*)
- Integritätsverlust (🇺🇸 *Integrity Impact*)
- Verfügbarkeitsverlust (🇺🇸 *Availability Impact*)
- Vertraulichkeitsanforderungen (🇺🇸 *Confidentiality Requirement*)
- Integritätsanforderungen (🇺🇸 *Integrity Requirement*)
- Verfügbarkeitsanforderungen (🇺🇸 *Availability Requirement*)

CVSS - Bewertung der Ausnutzbarkeit (*Exploitability Metrics*)

Attack Vector (AV): Network, Adjacent, Local, Physical

Attack Complexity (AC):
Low, High

Attack Requirements (AT):
None, Present

Privileges Required (PR):
None, Low, High

User Interaction (UI):
None, Passive, Active

Attack Vector

Network: Schwachstellen, die häufig „aus der Ferne ausnutzbar" sind und als ein Angriff betrachtet werden können, der auf Protokollebene über einen oder mehrere Netzknoten hinweg (z. B. über einen oder mehrere Router) ausgenutzt werden kann.

Adjacent: Der Angriff ist auf eine logisch benachbarte Topologie beschränkt. Dies kann z. B. bedeuten, dass ein Angriff aus demselben gemeinsamen Nahbereich (z. B. Bluetooth, NFC oder IEEE 802.11) oder logischen Netz (z. B. lokales IP-Subnetz) gestartet werden muss.

Local: Der Angreifer nutzt die Schwachstelle aus, indem er lokal auf das Zielsystem zugreift (z. B. Tastatur, Konsole) oder über eine Terminalemulation (z. B. SSH); oder der Angreifer verlässt sich auf die Interaktion des Benutzers, um die zum Ausnutzen der Schwachstelle erforderlichen Aktionen durchzuführen (z. B. mithilfe von Social-Engineering-Techniken, um einen legitimen Benutzer zum Öffnen eines bösartigen Dokuments zu verleiten).

Physical: Der Angreifer muss physisch Zugriff auf das Zielsystem haben, um die Schwachstelle auszunutzen.

Attack Complexity

Wie aufwendig ist es explizite Schutzmaßnahmen ((K)ASLR, Stack Canaries, ...) zu umgehen. Wie wahrscheinlich ist es, dass ein Angriff erfolgreich ist. Im Falle von *Race Conditions* können ggf. sehr viele Ausführungen notwendig sein bevor die Race Condition erfüllt ist.

Attack Requirements

Welcher Vorbedingungen (unabhängig von den expliziten Sicherungsmaßnahmen) müssen erfüllt sein, damit die Schwachstelle ausgenutzt werden kann. (z. B. der Nutzer muss sich an seinem Smartphone mindestens einmal seit dem Boot angemeldet haben (*After-First-Use* vs. *Before-First-Use*.))

Privileges Required

Welche Privilegien muss der Angreifer mindestens haben, um die Schwachstelle auszunutzen (Sind Adminstratorrechte erforderlich oder reichen normale Benutzerrechte).

User Interaction

Passiv bedeutet hier, dass der Nutzer unfreiwillig die Schwachstelle ausnutzt ohne bewusst Schutzmechanismen zu unterlaufen. Aktiv bedeutet, dass der Nutzer aktiv Interaktionen unternimmt, um die Schutzmechanismen des Systems auszuhebeln (z. B. durch das Installieren einer nicht-signierten Anwendung aus dem Internet).

CVSS - Bewertung der Auswirkung auf das betroffene System/Vulnerable System Impact Metrics

Confidentiality Impact (C):

None, Low, High

Integrity Impact (I):

None, Low, High

Availability Impact (A):

None, Low, High

CVSS - Bewertung der Auswirkung auf das nachgelagerte System/Vulnerable System Impact Metrics

Confidentiality Impact (C):

None, Low, High

Integrity Impact (I):

None, Low, High

Availability Impact (A):

None, Low, High

✎ Übung: Schwachstellen und Ihre Bewertung

Ihnen liegt eine externe Festplatte mit AES-256-XTX basierter Hardwareverschlüsselung vor. Das entsperren der Festplatte geschieht mit Hilfe eines speziellen Programms, dass ggf. vorher installiert werden muss und als Schutzmechanismus ein Passwort vorsieht.

Das Clientprogramm hasht erst das Passwort clientseitig, bevor es den Hash an den Controller der Festplatte überträgt. Die Firmware des Controllers validiert das Passwort in dem es den gesendeten Hash direkt mit dem bei der Einrichtung übermittelten Hash vergleicht; d. h. es finden keine weiteren sicherheitsrelevanten Operationen außer dem direkten Vergleich statt. Zum Entsperren der Festplatte ist es demzufolge ausreichend, den Hash aus der Hardware auszulesen und diesen an den Controller zu senden, um die Festplatte zu entsperren. Danach kann auf die Daten frei zugegriffen werden.

1. Ermitteln Sie den CVSS 4.0 Score für diese Schwachstelle. (CVSS Rechner)
2. Welche Anwendungsfälle sind für diese Schwachstelle denkbar?

✎ Übung: Schwachstellen und Ihre Bewertung

Durch die Analyse der Firmware eines Baseband-Prozessors haben Sie folgende Erkenntnisse erhalten: Wenn es Ihnen gelingt ein speziell manipuliertes Paket zu senden - welches außerhalb der Spezifikation liegt - dann kommt es zu einem Buffer-Overflow. Darauf aufbauend ist es möglich die Firmware des Baseband-Prozessors zum Absturz zu bringen. Dies führt direkt zu einem neustartet. Aufgrund des Neustarts muss der Nutzer dann jedoch seine SIM-Pin neu eingeben, um sich wieder gegenüber dem Mobilfunknetz zu authentifizieren.

Weitere Untersuchungen haben ergeben, dass es nicht möglich ist den Buffer-Overflow weitergehend auszunutzen, um nachgelagerte Systeme anzugreifen. Die Validierung der Kommunikation, die bei der Kommunikation des Baseband-Prozessors mit dem Hauptprozessor stattfindet, fängt Fehlerzustände effektiv ab.

1. Ermitteln Sie den CVSS 4.0 Score für diese Schwachstelle. (CVSS Rechner)
2. Welche Anwendungsfälle sind für diese Schwachstelle denkbar?

Baseband-Prozessoren

Dieser Chip Ihres Smartphones ist für die drahtlose Kommunikation zuständig. Als solcher hat der Baseband-Prozessor ein eigenes Betriebssystem; d. h. eine eigene Firmware. Diese wird typischerweise vom Hersteller des Baseband-Prozessors entwickelt. Die Kommunikation zwischen dem Baseband-Prozessor und dem Hauptprozessor erfolgt über eine wohl definierte, minimal gehaltene Schnittstelle, um die Auswirkungen von Sicherheitsproblemen ggf. eindämmen zu können.

Zuverlässigkeit von CVSS Bewertungen

Internet-wide Vulnerability Enables Giant DDoS Attacks

[...] MadeYouReset is officially tracked as CVE-2025-8671, with the same "high" 7.5 CVSS rating that Rapid Reset received.

The issue is also being tracked under a variety of other CVEs where it pertains to different HTTP/2 implementations. For example, MadeYouReset is a high-grade 8.2-rated issue (CVE-2025-55163) in the Netty application framework but only a medium-severity 6.9-rated issue (CVE-2025-54500) to F5 Networks' BIG-IP application delivery controllers (ADCs). [...]

—August, 18th, 2025 - Dark Reading

2. Common Vulnerabilities and Exposures (CVE)

Zweck von CVEs

- Schwachstellen eindeutig identifizieren und bestimmten Versionen eines Codes (z. B. Software und gemeinsam genutzte Bibliotheken) mit diesen Schwachstellen verknüpfen.
- Kommunikationsgrundlage bilden, damit mehrere Parteien über eine eindeutig identifizierte Sicherheitslücke diskutieren können. National Vulnerabilities Database - NIST

NVD Dashboard - 6. Juni 2026

Time Period	New CVEs Received	New CVEs Enriched
Today	26	0
This Week	1576	1068
This Month	1576	1068
Last Month	6984	3619
This Year	30688	20499

Exemplarische Beschreibung eines CVEs

Jeder CVE ist mit Hilfe eines wohldefinierten JSON-Dokuments beschrieben.

```
{ "dataVersion": "5.0",
  "cveMetadata": {
      "cveId": "CVE-2023-51034",
      "assignerOrgId": "8254265b-2729-46b6-b9e3-3dfca2d5bfca",
      "assignerShortName": "mitre",
      "datePublished": "2023-12-22T00:00:00" },
  "containers": { "cna": { ...,
        "descriptions": [ {
           "value": "TOTOlink [...] vulnerable to command execution [...]"
        } ], ...,
        "references": [{
           "url": "815yang.github.io/[...]totolink_UploadFirmwareFile/"
         } ], ...
} } }
```

National Vulnerability Database (NVD)[3]

- Auflistung aller CVEs und deren Bewertung
- Alle Schwachstellen in der NVD sind mit einer CVE-Kennung versehen
- Die NVD ist ein Produkt der NIST Computer Security Division, Information Technology Laboratory
- Verlinkt häufig weiterführend Seiten, die Lösungshinweise und Tools bereitstellen, um die Schwachstelle zu beheben
- Verweist auf entsprechende Schwachstellen gemäß CWEs
- Verlinkt gelegentlich *PoC* Exploits (🇺🇸 *Proof-of-Concept Exploits*)

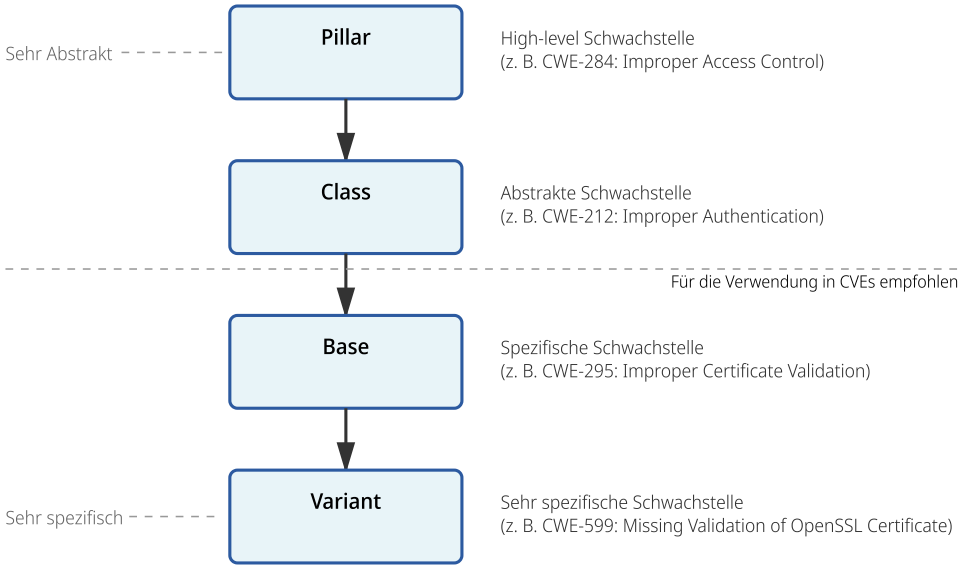
---

[3] Die NIS 2 Richtlinie der EU sieht auch den Aufbau einer europäischen Schwachstellen-Datenbank vor. Aktuell ist die NVD die zentrale Anlaufstelle bzgl. Schwachstellen.

Common Weakness Enumeration (CWE)

- eine kollaborativ entwickelte, vollständig durchsuchbare, kategorisierte Liste von Typen von Software- und Hardware-Schwachstellen und deren Beschreibung, dient als:
  - gemeinsame Sprache,
  - Messlatte für Sicherheitstools,
  - als Grundlage für die Identifizierung von Schwachstellen sowie für Maßnahmen zur Abschwächung und Prävention.

Beziehungen zwischen den CWEs[4]



[4] Inspiriert von: https://cwe.mitre.org/documents/cwe_usage/guidance.html

CWE - Schwachstellenkatalog TOP 25 in 2023

Rank	ID	Name	Rank Change vs. 2022
1	CWE-787	Out-of-bounds Write	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation (*Cross-site Scripting*)	0
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command (*SQL Injection*)	0
4	CWE-416	Use After Free	+3
5	CWE-78	Improper Neutralization of Special Elements used in an OS Command (*OS Command Injection*)	+1
6	CWE-20	Improper Input Validation	-2
7	CWE-125	Out-of-bounds Read	-2
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory (*Path Traversal*)	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	0
11	CWE-862	Missing Authorization	+5
12	CWE-476	NULL Pointer Dereference	-1
13	CWE-287	Improper Authentication	+1
14	CWE-190	Integer Overflow or Wraparound	-1
15	CWE-502	Deserialization of Untrusted Data	-3
16	CWE-77	Improper Neutralization of Special Elements used in a Command (*Command Injection*)	+1
17	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	+2
18	CWE-798	Use of Hard-coded Credentials	-3
19	CWE-918	Server-Side Request Forgery (SSRF)	+2
20	CWE-306	Missing Authentication for Critical Function	-2
21	CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization (*Race Condition*)	+1
22	CWE-269	Improper Privilege Management	+7
23	CWE-94	Improper Control of Generation of Code (*Code Injection*)	+2
24	CWE-863	Incorrect Authorization	+4
25	CWE-276	Incorrect Default Permissions	-5

Request Forgery = *Anfragefälschung*

Fest codierte Cloud-Zugangsdaten in populären Apps entdeckt

Betroffen sind mehrere Apps mit teils Millionen von Downloads. Den Entdeckern zufolge gefährdet dies nicht nur Backend-Dienste, sondern auch Nutzerdaten.

Sicherheitsforscher von Symantec haben Anwendungen aus dem Google Play Store und dem Apple App Store untersucht und dabei festgestellt, dass mehrere Apps mit teils Millionen von Downloads fest codierte und unverschlüsselte Anmeldedaten für verschiedene Clouddienste enthalten. Ent-

deckt wurden sowohl Schlüssel für den Zugang zu AWS-Ressourcen als auch solche für Microsoft Azure. […]

—23.10.2024 Golem.de

CVE-2023-50712 - *XSS* Schwachstelle[5]

> Iris is a web collaborative platform aiming to help incident responders sharing technical details during investigations. A stored Cross-Site Scripting (XSS) vulnerability has been identified in iris-web, affecting multiple locations in versions prior to v2.3.7. The vulnerability may allow an attacker to inject malicious scripts into the application, which could then be executed when a user visits the affected locations. This could lead to unauthorized access, data theft, or other related malicious activities. An attacker needs to be authenticated on the application to exploit this vulnerability. The issue is fixed in version v2.3.7 of iris-web. No known workarounds are available.
>
> —Published: December 22, 2023

Bewertung: CVSS V3.1: 5.4 MEDIUM

---

[5] 🇺🇸 *Cross-Site Scripting* (XSS) wird im nächsten Kapitel behandelt.

CVE-2023-51034 - *Arbitrary Code Execution*

> TOTOlink EX1200L V9.3.5u.6146_B20201023 is vulnerable to arbitrary command execution via the cstecgi.cgi UploadFirmwareFile interface.
>
> —Published: December 22, 2023; Last modified: January 2, 2024

Bewertung: CVSS V3.1: 9.8 Critical

PoC Exploit: 815yang.github.io

Weakness Enumeration:

CWE-434 Unrestricted Upload of File with Dangerous Type

Bei TOTOlink EX1200L handelt es sich um einen Wifi Range Expander.

PoC ≙ Proof-of-Concept

CWE-434 Unrestricted Upload of File with Dangerous Type

Beschreibung	Art der Einführung	Scope
Das Produkt ermöglicht es dem Angreifer, Dateien gefährlicher Typen hochzuladen oder zu übertragen, die in der Produktumgebung automatisch verarbeitet werden können.	Diese Schwäche wird durch das Fehlen einer Sicherheitstaktik während der Architektur- und Entwurfsphase verursacht.	Willkürliche Codeausführung ist möglich, wenn eine hochgeladene Datei vom Empfänger als Code interpretiert und ausgeführt wird. [...] Somit ist ggf. die Integrität, Vertraulichkeit und Verfügbarkeit betroffen.

—Mitre.org (2023; übersetzt mit DeepL)

CVE-2023-51034 - PoC (gekürzt)

Initiale Anfrage

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
[...]
{
    "FileName":
        ";ls../>/www/yf.txt;",
    "topicurl":
        "UploadFirmwareFile"
}
```

Abfrage der Datei (hier: `yf.txt`)

```
GET /yf.txt HTTP/1.1
[...]
Connection: close
```

Das Ergebnis ist eine Datei mit der Auflistung der Dateien im Verzeichnis (`..`).

CVE-2023-51034 - zugrundeliegende Schwachstelle

```
Var = (const char *)websGetVar(a1, "FileName", &byte_42FE28);
v3 = (const char *)websGetVar(a1, "FullName", &byte_42FE28);
v4 = (const char *)websGetVar(a1, "ContentLength", &word_42DD4C);
v5 = websGetVar(a1, "flags", &word_42DD4C);
v6 = atoi(v5);
Object = cJSON_CreateObject();
v8 = fopen("/dev/console", "a");
v9 = v8;
if ( v8 ) {
    fprintf(v8, "[%s:%d] FileName=%s,FullName=%s,ContentLength=%s\n",
                "UploadFirmwareFile", 751, Var, v3, v4);
    fclose(v9);
}
v10 = strtol(v4, 0, 10) + 1;
strcpy(v52, "/tmp/myImage.img");
doSystem("mv %s %s", Var, v52);
```

Die Lücke ist auf die folgenden Zeilen zurückzuführen:

```
Var = (const char *)websGetVar(a1, “FileName”, &byte_42FE28);
...
doSystem(“mv %s %s”, Var, v52);
```

Der Aufruf von doSystem ermöglicht die Ausführung von beliebigem Code. Der Angreifer kann den Wert von Var so manipulieren, dass er quasi beliebigen Code ausführen kann.

Ausgenutzte Schwachstellen

Der Known Exploited Vulnerabilities (KEV) Katalog der CISA umfasst Produkte deren Schwachstellen ausgenutzt wurden oder aktiv ausgenutzt werden.

- Kriterien für die Aufnahme in den KEV Katalog:
  1. Eine CVE-Id liegt vor.
  2. Die Schwachstelle wird aktiv ausgenutzt (🇺🇸 *Active Exploitation*) (ggf. reicht es jedoch wenn „nur“ ein *Honeypot* aktiv angegriffen wurde) - ein PoC reicht nicht aus.
  3. Eine Handlungsempfehlung liegt vor (z. B. Patch, Workaround oder vollständige Abschaltung).
- Firmen sollten die KEV *Schwachstellen priorisieren*, um die Wahrscheinlichkeit eines erfolgreichen Angriffs zu verringern.

CISA = *Cybersecurity and Infrastructure Security Agency* (oder *America's Cyber Defense Agency*)

Ausgewählte Amerikanische Behörden sind sogar verpflichtet innerhalb vorgegebener Zeiträume zu reagieren.

Erzwungene Außerbetriebnahme von Produkten

Abschaltbefehl: US-Behörden müssen Ivanti-Geräte vom Netz nehmen

[...] In einer jetzt veröffentlichten "Emergency Directive" weist die US-amerikanische Cybersicherheitsbehörde CISA alle Bundesbehörden an, Produkte vom Typ "Ivanti Connect Secure" oder "Ivanti Policy Secure" unverzüglich vom Netz zu nehmen. Sie reagiert damit auf massenhafte Angriffe gegen die schadhaften Geräte. [...]

—02.02.2024 - Heise.de

Alte CVEs können relevant und im KEV Katalog sein ...

CISA flags actively exploited Office relic [...]

CISA has added a [...] security holes to its actively exploited list [...] a years-old flaw in Microsoft Office.

The latest update to CISA's Known Exploited Vulnerabilities catalog flags [...] *CVE-2009-0556*, a PowerPoint code injection bug that's been lurking for more than 15 years.

[...] CISA also flagged CVE-2009-0556, a Microsoft Office PowerPoint code injection vulnerability rated 8.8 on the CVSS scale. The bug, confirmed by Microsoft back in 2009, allows remote attackers to execute arbitrary code via memory corruption when a user opens a specially crafted PowerPoint file. Microsoft patched the issue years ago as part of MS09-017, but its appearance in the KEV catalog indicates that unpatched or unsupported systems are still being successfully targeted.

—Jan. 2026 - The Register

2023 CWE Top 10 KEV Weaknesses

Schwachstelle	CWE ID	# CVE Mappings in KEV	Avg. CVSS
Use After Free	416	44	8.54
Heap-based Buffer Overflow	122	32	8.79
Out-of-bounds Write	787	34	8.19
Improper Input Validation	20	33	8.27
Improper Neutralization of Special Elements used in an OS Command (*OS Command Injection*)	78	25	9.36
Deserialization of Untrusted Data	502	16	9.06
Server-Side Request Forgery (SSRF)	918	16	8.72
Access of Resource Using Incompatible Type (*Type Confusion*)	843	16	8.61
Improper Limitation of a Pathname to a Restricted Directory (*Path Traversal*)	22	14	8.09
Missing Authentication for Critical Function	306	8	8.86

2024 CWE Top 10 KEV Weaknesses

CWE ID	Schwachstelle	Avg. CVSS	# CVE Mappings in KEV
CWE-787	Out-of-bounds Write	75.59	18
CWE-843	Access of Resource Using Incompatible Type ('Type Confusion')	24.91	6
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	24.27	6
CWE-94	Improper Control of Generation of Code ('Code Injection')	23.64	7
CWE-502	Deserialization of Untrusted Data	23.07	5
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	19.52	5
CWE-306	Missing Authentication for Critical Function	17.60	6
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	15.62	4
CWE-416	Use After Free	15.43	5
CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	14.90	4

Schwachstellen, die auf Fehler beim Speicherzugriff zurückzuführen sind, sind nicht (mehr) notwendig!

Google hails move to Rust for huge drop in memory vulnerabilities

[...] Memory access vulnerabilities often occur in programming languages that are not memory safe. In 2019, memory safety issues accounted for 76% of all Android vulnerabilities.

[...] the transition to memory safe languages through the gradual use of memory safe code in new projects and developments over a five year period. The results showed that despite a gradual rise in code [still] being written in memory unsafe languages, memory safety vulnerabilities dropped significantly.

[...] there has been a significant drop in the number of memory-related vulnerabilities, with memory safe vulnerabilities down to 24% in 2024 [...]

—26. September 2024 - Techradar.com

Offenlegung von Sicherheitslücken nach CISA [6]

Coordinated Vulnerability Disclosure (CVD)

1. Sammlung von Schwachstellenmeldungen
2. Analyse der Schwachstellenmeldungen zusammen mit den Herstellern, um die Sicherheitsauswirkungen zu verstehen.
3. Entwicklung von Strategien zur Eindämmung der Schwachstellen; insbesondere Entwicklung von notwendigen Patches.
4. Anwendung der Strategien zur Eindämmung der Schwachstellen in Zusammenarbeit mit dem Hersteller und ggf. betroffenen Nutzern.
5. Veröffentlichung der Schwachstellenmeldung in Abstimmung mit der Quelle des Schwachstellenberichts und dem Hersteller.

CISA (America's Cybersecurity and Infrastructure Security Agency/Cyber Defense Agency).

Quellen für Schwachstellen

- Eigene Schwachstellenanalysen
- Überwachung öffentlicher Quellen
- Direkte Meldungen von Herstellern, Forschern und Nutzern

---

[6] Das BSI verfährt ähnlich; tut sich aber sehr schwer.

Zeitlicher Rahmen für die Offenlegung von Sicherheitslücken

Der Zeitrahmen für die Offenlegung von Sicherheitslücken wird durch folgende Faktoren bestimmt:

- Aktive Ausnutzung der Schwachstelle
- besonders kritische Schwachstellen
- Auswirkungen auf Standards
- bereits öffentlich bekannt (zum Beispiel durch einen „naïven“` Forscher)
- Auswirkungen auf die kritische Infrastruktur, öffentliche Gesundheit und Sicherheit
- die Verfügbarkeit von effektiven Eindämmungsmaßnahmen
- das Verhalten des Herstellers und die Möglichkeit der Entwicklung eines Patches
- Schätzung des Herstellers wie lange es dauert einen Patch zu entwickeln, zu testen und auszurollen.

Welche neuen Schwachstellen werden in absehbarer Zeit ausgenutzt?

⊙ Beobachtung

Am 1. Oktober 2023 hat die NVD 139.473 CVEs veröffentlicht. In den folgenden 30 Tagen wurden 3.852 CVEs beobachtet, die ausgenutzt (🇺🇸 *exploited*) wurden.

Ca. 5-6% aller Schwachstellen werden „irgendwann“ ausgenutzt. [7]

? Frage

Wie stelle ich sicher, dass ich meine Bemühungen zum Beseitigen der Schwachstellen auf diejenigen konzentriere, die am wahrscheinlichsten zeitnahe ausgenutzt werden?

---

[7] Fortinet, Threat Landscape Report Q2 2018

Nutzung des CVSS als Grundlage für die Schätzung?

Annahme: Schwachstellen mit einem CVSS Score ≥ 7 (d. h. mit einer Bewertung von Hoch oder kritisch) werden ausgenutzt.

- 80.024 Schwachstellen haben einen CVSS Score ≥ 7

  Ausgenutzt wurden: 3.166

- 59.449 Schwachstellen haben eine CVSS < 7

  Ausgenutzt wurden: 686

Zusammenfassung

Die Strategie „Priorisierung von Schwachstellen mit einem bestimmten CVSS Score“ (hier ≥ 7) ist keine geeignete Strategie, da sie nicht alle relevanten Schwachstellen erfasst (686 *False Negatives*) und - ganz insbesondere - zu viele Schwachstellen (76.858 *False Positives*) erfasst, die nicht ausgenutzt werden.

Exploit Prediction Scoring System (EPSS)

- EPSS ist eine Methode zur *Bewertung der Wahrscheinlichkeit*, dass eine Schwachstelle in den nächsten 30 Tagen ausgenutzt wird.
- EPSS basiert auf der Analyse von Schwachstellen, die in den letzten 12 Monaten ausgenutzt wurden.
- EPSS nutzt KI basierend auf folgenden Informationen (Stand Jan. 2024):
  - Hersteller
  - Alter der Schwachstelle (Tage seit der Veröffentlichung des CVEs)
  - die Beschreibung der Schwachstelle
  - betroffene CWEs
  - CVSS Bewertungen der Schwachstellen
  - Wird der CVE auf bekannten Listen diskutiert bzw. aufgelistet?
  - Gibt es öffentliche verfügbare Exploits?

Nutzung des EPSS für die Schätzung? [8]

Annahme: Schwachstellen mit EPSS 10% und größer sind werden ausgenutzt werden.

- 3.735 Schwachstellen haben ein Wahrscheinlichkeit von EPSS 10% und größer

  Ausgenutzt wurden: 2.435

- 135.738 Schwachstellen haben ein EPSS $<$ 10%

  Ausgenutzt wurden: 1.417

Zusammenfassung

Die Strategie „Priorisierung von Schwachstellen mit einem EPSS von 10% und höher“ ist eine geeignete*re* Strategie, da sehr viele relevante Schwachstellen erfasst werden und - ganz insbesondere - die Anzahl der zu beachtenden Schwachstellen ganz massiv reduziert wird ohne die Gesamtqualität *zu stark* zu beeinflussen.

---

[8] Enhancing Vulnerability Prioritization: Data-Driven Exploit Predictions with Community-Driven Insights

3. Schwachstellenmanagement

Vulnerabilities Equities Process (VEP) (USA) [9]

> [...] Der *Vulnerability-Equity-Process (VEP)* wägt ab, ob Informationen über Schwachstellen an den Hersteller/Lieferanten weitergegeben werden sollen, in der Erwartung, dass sie gepatcht werden, oder ob die Kenntnis der Schwachstelle vorübergehend auf die US-Regierung und möglicherweise andere Partner beschränkt werden soll, damit sie für Zwecke der nationalen Sicherheit und der Strafverfolgung, wie z. B. nachrichtendienstliche Erfassung, militärische Operationen und/oder Spionageabwehr, genutzt werden können. [...]
>
> —Übersetzung: DeepL

Insbesondere durch die föderale Struktur in Deutschland kann es ggf. dazu kommen, dass bezüglich der Handhabung von Schwachstellen unterschiedliche rechtliche Regelungen gelten werden - je nachdem ob die Behörde eine Bundes- oder Landesbehörde ist.

---

[9] Die rechtlichen Rahmenbedingungen bzgl. eines effektiven Schwachstellenmanagement sind in Deutschland gerade in der Diskussion. (Stand Jul. 2024); Schwachstellen, die direkt an das BSI gemeldet werden, unterliegen dem vorher diskutierten CVD.

Vulnerabilities Equities Process (VEP) (USA)

> [...] Die Entscheidung der US-Regierung, ob eine Schwachstelle veröffentlicht oder eingeschränkt werden soll, ist nur ein Element des Prozesses zur Bewertung der Schwachstellen und ist nicht immer eine binäre Entscheidung. Andere Optionen, die in Betracht gezogen werden können, sind die Verbreitung von Informationen zur Schadensbegrenzung an bestimmte Stellen, ohne die jeweilige Schwachstelle offenzulegen, die Einschränkung der Nutzung der Schwachstelle durch die US-Regierung in irgendeiner Weise, die Information von Regierungsstellen der USA und verbündeter Staaten über die Schwachstelle [...].
>
> —Übersetzung: DeepL

Vulnerabilities Equities Process (VEP) (USA)

> [...] Alle diese Entscheidungen müssen auf der Grundlage des Verständnisses der Risiken einer Verbreitung, des potenziellen Nutzens von Schwachstellen durch die Regierung sowie der Risiken und Vorteile aller dazwischen liegenden Optionen getroffen werden. [...]
>
> —Übersetzung: DeepL

Schwachstellenmanagement in Deutschland - Quo Vadis?

> [...] Die Ausnutzung von Schwachstellen von IT-Systemen steht in einem hochproblematischen Spannungsverhältnis zur IT-Sicherheit und den Bürgerrechten. Der Staat wird daher keine Sicherheitslücken ankaufen oder offenhalten, sondern sich in einem Schwachstellenmanagement unter Federführung eines unabhängigeren Bundesamtes für Sicherheit in der Informationstechnik immer um die schnellstmögliche Schließung bemühen.[...]
>
> —KOALITIONSVERTRAG 2021—2025 (SPD, BÜNDNIS 90/DIE GRÜNEN, FDP)

Schwachstellenmanagement und die NIS 2 Richtlinie der EU

Artikel 7 - Nationale Cybersicherheitsstrategie

(2) Im Rahmen der nationalen Cybersicherheitsstrategie nehmen die Mitgliedstaaten insbesondere Konzepte an …

c) für das Vorgehen bei Schwachstellen, das die Förderung und Erleichterung der koordinierten Offenlegung von Schwachstellen nach Artikel 12 Absatz 1 umfasst;

Artikel 12 - Koordinierte Offenlegung von Schwachstellen und eine europäische Schwachstellendatenbank

(1) Jeder Mitgliedstaat benennt [einen] Koordinator für die Zwecke einer koordinierten Offenlegung von Schwachstellen. [Der Koordinator] fungiert als vertrauenswürdiger Vermittler und erleichtert erforderlichenfalls die Interaktion zwischen der eine Schwachstelle meldenden natürlichen oder juristischen Person und dem Hersteller oder Anbieter der potenziell gefährdeten IKT-Produkte oder IKT-Dienste auf Ersuchen einer der beiden Seiten.

✎ CVEs - Übung

1. Finden Sie Schwachstellen, die macOS Tahoe betreffen.
2. Finden Sie heraus um was es bei CVE-2020-20095 geht.

✎ Schwachstellenmanagement - Übung

Teilen Sie sich in drei Gruppen auf und bereiten Sie jeweils ein kurzes Statement (zwischen 1,5 und 3 Minuten) vor - gemäß Ihrer Gruppenzuteilung [10]. Bereiten Sie sich auch darauf vor, Ihr Statement zu verteidigen bzw. die anderen Gruppen zu kritisieren.

Sie haben 15 Minuten Zeit. Danach werden wir die Statements präsentiert und verteidigt.

Schwachstellen sofort schließen	Schwachstellen begrenzt nutzen	Schwachstellen zurückhalten
Jede Schwachstelle ist ein Risiko und sollte sofort geschlossen werden.	Eine begrenzte Nutzung von Schwachstellen ist für die nationale Sicherheit notwendig; eine Befeuerung des Markets sollte aber nicht erfolgen.	Für Zwecke der nationalen Sicherheit sollen Schwachstellen zurückgehalten und ggf. auch gekauft werden.

[10]D. h. versetzen sie sich in die zugewiesene Rolle und finden Sie entsprechende Argumente!