Propädeutik der Datensicherheit

https://iuk.one/DS-PROP.pdf

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock, Germany
clemens.cap@uni-rostock.de

2026-11-09 16:53



Scheinbar ganz einfach...

Das ist verschlüsselt.

Daher ist es sicher!

Und wir lernen, warum das falsch ist...

Übersicht

1. **Grundbegriffe**

2. **Tiefere Ursachen**

2.1. Menschliches Fehlverhalten

2.2. Gestiegene Komplexität

2.3. Ebene und Meta-Ebene

3. **Mindset**

4. **Rechtslage**

1. Grundbegriffe

2. Tiefere Ursachen

3. Mindset

4. Rechtslage

1. Grundbegriffe

Auf ersten Blick ganz einfach...

Grundbegriffe der Verschlüsselung

Bauteile

Klartext (plain text): Die für jeden offen lesbare (ursprüngliche) Information.

Chiffretext (cipher text): Veränderte, verschlüsselte Information, nur mit geheimem Zusatzwissen ("Schlüssel") lesbar.

Schlüssel

Geheimes Zusatzwissen, das man kennen muß, um den Chiffretext lesen zu können.

Verfahren

Verschlüsseln (encrypt): Verfahren, das aus Klartext Chiffretext macht.

Entschlüsseln (decrypt): Verfahren, das aus Chiffretext wieder Klartext macht.

Grundbegriffe der Digitalen Unterschrift

Dokument

Die Information, die unterschrieben wird. Sie bleibt bei der Unterschrift lesbar.

Variante: Statt eines Dokuments wird ein "Fingerabdruck" des Dokuments unterschrieben.

Verfahren

Signieren: Aus Dokument und Schlüssel die Unterschrift (Signatur) berechnen.
Verifizieren: Prüfen, ob eine angebliche Unterschrift mit dem Signierschlüssel erstellt wurde.

Signierschlüssel

Geheimer Schlüssel, der zur Erstellung der Unterschrift benötigt wird.

Was die Unterschrift garantiert

Echtheit: Die Unterschrift stammt von einem Inhaber des Signierschlüssels.

Ein Beispiel

Was ist das?

050211 043516 005015 005032 000000 006400
044111 051104

Das ist jetzt

1. Klartext
2. Chiffretext



https://twbk.de/2ta6



Abb. 1: Der Linux Pinguin. (Sein Geheimnis wurde in der Vorlesung verraten!) Rechte s. Anhang.

Symmetrische Verfahren

Symmetrische Verfahren (auch: **secret key** Verfahren)

- Zum Verschlüsseln und Entschlüsseln wird **derselbe** Schlüssel genutzt.
- Kenntnis des **einen** Schlüssels erlaubt Verschlüsseln und Entschlüsseln.

Probleme:

1. **Schlüssel-Verteilung:** Wie verteile ich den geheimen Schlüssel auf sichere Weise?
   Die Verschlüsselung liefert ja erst die sichere (geheime) Übertragung
2. **Vertrauen:** Jeder, der den Schlüssel hat, kann entschlüsseln.
   Jeder, der verschlüsseln will, muß den Schlüssel kennen.
   Muß jedem vertrauen, der den Schlüssel kennt,
   daß er den Schlüssel nicht weitergibt.

Asymmetrische Verfahren

Asymmetrische Verfahren (auch: **public-private key** Verfahren)

- Jeder Teilnehmer hat zwei zueinander passende Schlüssel.
- **Verschlüsselungs-Schlüssel (public key):** **Zum Zusperren**
  Jeder kann an jeden verschlüsseln und kennt die public keys aller.
- **Entschlüsselungsschlüssel (private key):** **Zum Aufsperren**
  Nur berechtigter Empfänger kann entschlüsseln, mit seinem geheim gehaltenen private key.
- **Wichtig:** Aus dem public key kann der private key nicht bestimmt werden.
- Vorteil: Kein Problem mit der Schlüssel-Verteilung.
- Bis ca. 1970 nicht klar, daß es so etwas überhaupt gibt.

Probleme:

1. **Schlüssel-Zuordnung:** Wie erhalte ich den **richtigen** public key des Empfängers.
2. **Performanz:** Im Vergleich zu symmetrischen Verfahren rechenaufwendig.
   Symmetrische Verfahren: 1-10 GB pro Sekunde
   Asymmetrischer Verfahren: 10-100 KB pro Sekunde

Hybride Verfahren

Problem: Beide Verfahren haben Probleme.

- **Symmetrische** sind schnell,
  aber der gemeinsame Schlüssel muß vorher sicher ausgetauscht werden.
- **Asymmetrische** bieten ein Verfahren für sicheren Schlüsselaustausch,
  aber sind ca. 1000-mal langsamer.

Idee: Vorteile beider Verfahren kombinieren

1. Asymmetrisches Verfahren übermittelt geheimen Schlüssel eines symmetrischen Verfahrens.
2. Symmetrisches Verfahren verschlüsselt die eigentliche (lange) Nachricht.
3. Nach einiger Zeit: Symmetrischen Schlüssel verwerfen und einen neuen aushandeln.

Beispiele: HTTPS (TLS), Signal, WhatsApp, PGP / GnuPG, S/MIME.

Ciphertext?

Text 1: Hello World.

Text 2: Ifmmp Xpsme.

Text 3:
132H424e968l346l111l437o299W446o232r195l666d

Text 1 ist vermutlich **Plaintext**.

Was ist mit Text 2 und Text 3? Was gilt?

1. Text 2 ist Ciphertext
2. Text 3 ist Ciphertext

https://twbk.de/2ta6

Problem 1: Falscher Umgang mit dem Schlüssel (1)

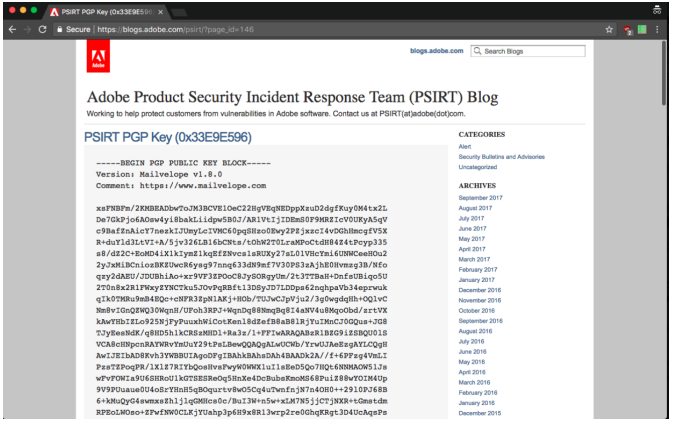


Abb. 2: Adobe publiziert den öffentlichen Schlüssel… Rechte s. Anhang.

Problem 1: Falscher Umgang mit dem Schlüssel (2)

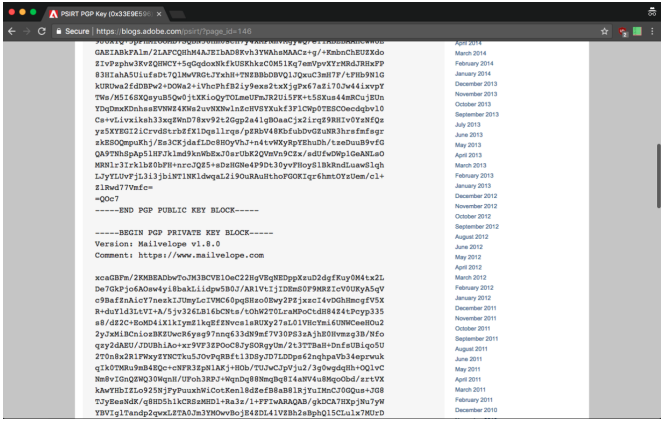


Abb. 3: und **versehentlich auch den privaten** Schlüssel. Das ist ein Problem, da der private Schlüssel geheim gehalten werden muß. Bericht 1 Bericht 2 Rechte s. Anhang.

Problem 1: Falscher Umgang mit dem Schlüssel (3)

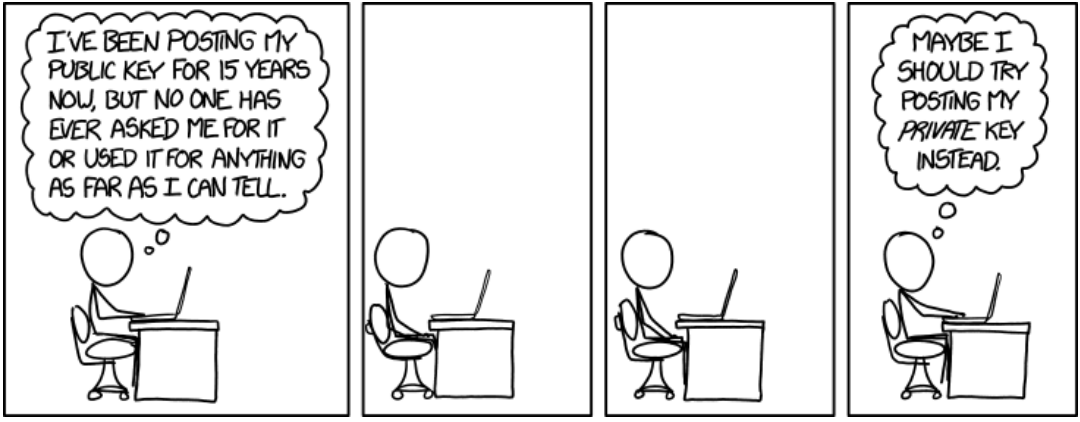


Abb. 4: Oftmals ist auch **Unkenntnis** des Benutzers das Problem. XKCD1553. Quelle und Erklärung Rechte s. Anhang.

Problem 2: Falsches Angreifermodell

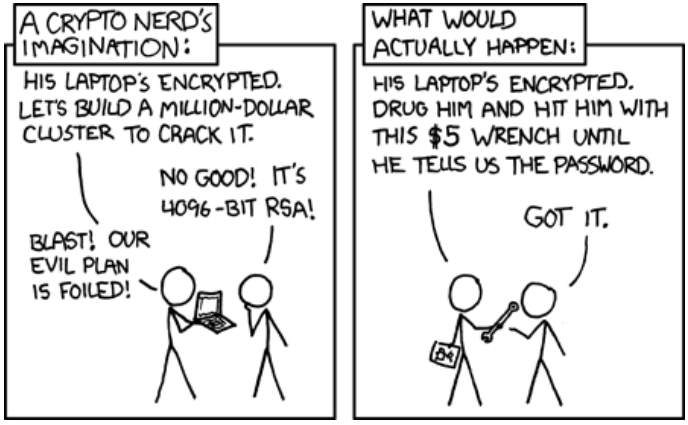


Abb. 5: Das **falsche Angreifermodell**, also die falsche Einschätzung der Fähigkeiten des Angreifers. Neben Folter der Personen, die über den privaten Schlüssel verfügen, ist bei 4096-Bit RSA auch die Verfügbarkeit eines Quantencomputers eine gute Möglichkeit für Angreifer. XKCD 538. Quelle und Erklärung Rechte s. Anhang.

Problem 3: Fehlende echte Ende-zu-Ende Verschlüsselung (1)

Ende-zu-Ende Verschlüsselung (end-to-end encryption, E2EE) bedeutet:

1. Nachricht beim Absender verschlüsselt und erst beim Empfänger wieder entschlüsselt.
2. Nur Absender und Empfänger kennen die jeweils erforderlichen Schlüssel.
3. Auf dem gesamten Weg dazwischen liegt nur Chiffretext vor.

Transportverschlüsselung leistet weniger als Ende-zu-Ende Verschlüsselung

Die Nachricht wird nur abschnittsweise verschlüsselt.

Bsp: Vom Absender zum Server und vom Server zum Empfänger ist verschlüsselt.
Auf dem Server liegt die Nachricht als Klartext vor.

Folge

Bei Transportverschlüsselung kann der Betreiber des Servers (und jeder, der Zugriff auf den Server hat) mitlesen und verändern. Bei Ende-zu-Ende Verschlüsselung geht das nicht.

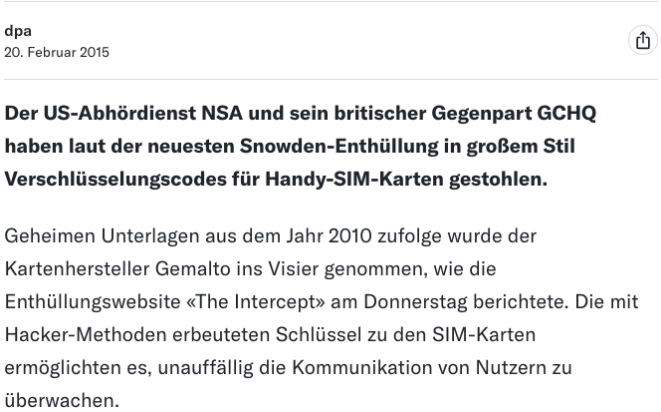
Problem 3: Fehlende echte Ende-zu-Ende Verschlüsselung (2)

Hartmut Pohl, Sprecher des Arbeitskreises „Datenschutz und IT-Sicherheit“ der Gesellschaft für Informatik e.V. (GI): „Grundsätzlich begrüßen wir die Möglichkeit eines Kontaktes mit Behörden auf elektronischem Weg. Allerdings bietet die De-Mail in ihrer derzeitigen Form keine wirklich höhere Sicherheit gegenüber der herkömmlichen E-Mail. Bürgern darf hier nicht falsche Sicherheit vorgetäuscht werden.“

De-Mails werden nämlich standardmäßig nicht Ende-zu-Ende-verschlüsselt, obwohl die Möglichkeit dieser Verschlüsselung prinzipiell vorgesehen ist. Irreführenderweise ist die standardmäßig eingestellte Verschlüsselung eine reine Transportverschlüsselung, die den Datentransfer von Server zu Server gegen Abhören absichert – allerdings alle Mails in den Servern unverschlüsselt zwischenspeichert. Diese Transportverschlüsselung ist nicht neu, sondern wird vielmehr schon seit Jahren von E-Mail-Anbietern verwendet.

Abb. 6: Der Unterschied zwischen Ende-zu-Ende Verschlüsselung und Transportverschlüsselung wird oft nicht klar genug dargestellt. Quelle: Gesellschaft für Informatik Rechte s. Anhang.

Problem 4: Jemand hat einen Nachschlüssel (1)



dpa
20. Februar 2015

Der US-Abhördienst NSA und sein britischer Gegenpart GCHQ haben laut der neuesten Snowden-Enthüllung in großem Stil Verschlüsselungscodes für Handy-SIM-Karten gestohlen.

Geheimen Unterlagen aus dem Jahr 2010 zufolge wurde der Kartenhersteller Gemalto ins Visier genommen, wie die Enthüllungswebsite «The Intercept» am Donnerstag berichtete. Die mit Hacker-Methoden erbeuteten Schlüssel zu den SIM-Karten ermöglichten es, unauffällig die Kommunikation von Nutzern zu überwachen.

Abb. 7: Der amerikanische Geheimdienst NSA und der britische Geheimdienst GCHQ **haben Nachschlüssel** zu den Verschlüsselungs-Systemen in den Mobilfunk-Netzen 3G und 4G (LTE). Rechte s. Anhang.

Problem 4: Jemand hat einen Nachschlüssel (2)

One of the world's largest manufacturers of SIM cards has acknowledged evidence of security agency attacks on the company's internal networks, but it's denying that American and British intelligence agents were able to get access to billions of mobile phone users' secure data.

Gemalto, a French-Dutch supplier of SIM cards, found "reasonable grounds" of an attack by U.S. National Security Agency and its British counterpart, the Government Communications Headquarters (GCHQ) following an internal investigation into a series of security incidents. The audits came after online publication The Intercept reported on what it said was a joint British-American operation to covertly hack Gemalto's stash of SIM encryption keys, based on documents leaked by Edward Snowden.

Abb. 8: Zweite, unabhängige Quelle. Rechte s. Anhang.

Problem 4: Jemand hat einen Nachschlüssel (3)

Wenn ich über WhatsApp kommuniziere, dann ist das verschlüsselt.
Mein geheimer, privater Schlüssel wurde durch WhatsApp für mich erzeugt.
Woher weiß ich, daß WhatsApp den Schlüssel nicht an Mark Zuckerberg sendet?
Eigentlich müßte ich einen **selbsterstellten** Schlüssel einspeisen können.
Und woher weiß ich dann, daß die Anwendung diesen nicht auch weiterleitet?

Zentrale Fragen:

1. Habe ich den Schlüssel **sicher erzeugt**?
2. Kann ich den Schlüssel selber in das System einspeisen?
3. Kann jemand anderes den Schlüssel **stehlen** oder **austauschen**?

Problem 5: Der Schlüsselgenerator hat ein Problem

Dual-EC-DRBG: Dual Elliptic Curve Deterministic Random Bit Generator.

1. Vom amerikanischen Geheimdienst NSA entwickelt.
2. Vom NIST National Institute of Standards ad Technology als Standard akzeptiert.
3. Hat nachgewiesenermaßen eine Hintertüre. [BLN16]

> Cryptographers have long suspected that the agency planted vulnerabilities in a standard adopted in 2006 by the National Institute of Standards and Technology and later by the International Organization for Standardization, which has 163 countries as members.
> Classified N.S.A. memos appear to confirm that the fatal weakness, discovered by two Microsoft cryptographers in 2007, was engineered by the agency. The N.S.A. wrote the standard and aggressively pushed it on the international group, privately calling the effort "a challenge in finesse."

Abb. 9: Die entsprechende Veröffentlichung hat den bemerkenswerten Titel: Dual EC: A Standardized Back Door [BLN16] Rechte s. Anhang.

Problem 6: Falscher öffentlicher Schlüssel benutzt (1)

Ziel:

- Alice verschlüsselt an Bob und nutzt dazu den öffentlichen Schlüssel von Bob.
- Nur Bob kann lesen: Nur Bob hat den passenden privaten Schlüssel.

Situation:

- Alice nutzt zur Verschlüsselung versehentlich den öffentlichen Schlüssel von Mallory.
- Nur Mallory kann lesen: Nur Mallory hat den passenden privaten Schlüssel.
- Mallory kann nun die Nachricht für Bob verschlüsseln.
- Alice und Bob sehen, daß verschlüsselt wurde.
- Mallory kann aber mitlesen und sogar verändern.

Angriff:

- Angreifer unterschiebt Alice den eigenen öffentlichen Schlüssel als jenen von Bob.

Problem 6: Falscher öffentlicher Schlüssel benutzt (2)

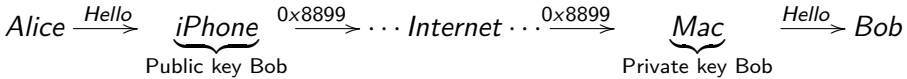


Abb. 10: Was Alice **beabsichtigt:** Die Verschlüsselung einer Nachricht an Bob. Dazu muß sie mit dem public key von Bob verschlüsseln. Bob entschlüsselt dann mit seinem private key. Rechte s. Anhang.

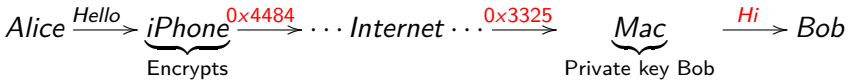


Abb. 11: Was **tatsächlich** passiert: Alice verschlüsselt mit einem Schlüssel, den sie **fälschlicherweise** für Bobs public key hält, Bob entschlüsselt mit dem seinem private key. Rechte s. Anhang.

Problem 6: Falscher öffentlicher Schlüssel benutzt (3)

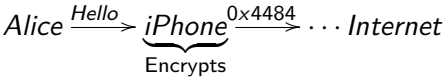


Abb. 12: Was Alice sieht: Eine verschlüsselte Nachricht verläßt ihren Rechner. **Sie hat aber nicht geprüft, ob der richtige public key zur Anwendung kam.** Das kann sie aber auch nicht so ohne weiteres. Insbesondere kann sie nicht versuchen, den Text zu entschlüsseln, denn dazu bräuchte sie den privaten Schlüssel von Bob. Rechte s. Anhang.



Abb. 13: Was Bob sieht: Er erhält eine verschlüsselte Nachricht und kann sie mit seinem private key in einen sinnvollen deutschen Text entschlüsseln. Rechte s. Anhang.

Problem 6: Falscher öffentlicher Schlüssel benutzt (4)

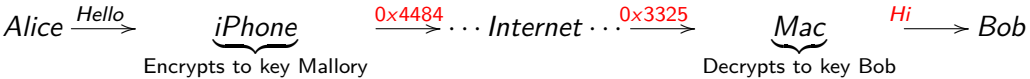


Abb. 14: Was wir aus einer angenommenen Super-Perspektive beobachten. Rechte s. Anhang.

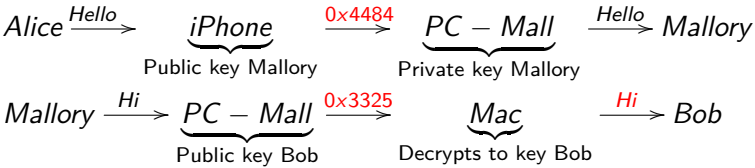


Abb. 15: Was im Detail geschieht: Alice verschlüsselt versehentlich an Mallory. Mallory liest die Nachricht, verändert sie, und sendet sie danach weiter an Bob, dabei verschlüsselt Mallory nun korrekt an Bob. Rechte s. Anhang.

Problem 6: Falscher öffentlicher Schlüssel benutzt (5)

Zusammenfassung

Man-in-the-Middle Angriff (MITM)

Alice hält den public key von Mallory fälschlicherweise für den public key von Bob.
Mallory hat die Möglichkeit, die Kommunikation von Alice zu Bob zu interceptieren.

Was der Angreifer kann

Alle Nachrichten mitlesen, verändern, unterdrücken oder neue einschleusen.

Die Personen der Kryptographie

Konvention

In der Kryptographie haben Beteiligten feste Namen und Rollen.

- **Alice:** Will eine Nachricht senden. Legitime Teilnehmerin.
- **Bob:** Soll die Nachricht empfangen. Legitimer Teilnehmer.
- **Carol:** Dritte legitime Teilnehmerin, falls noch benötigt.
- **Eve:** Eavesdropper, Lauscherin.
  Passive Angreiferin: Kann nur mithören.
- **Mallory:** Malicious, bösartig.
  Aktive Angreiferin: Kann Nachrichten verändern, unterdrücken, einspeisen.
- **Trent:** Trusted Third Party. Vertrauenswürdige dritte Instanz.

2. Tiefere Ursachen

2.1. Menschliches Fehlverhalten

2.2. Gestiegene Komplexität

2.3. Ebene und Meta-Ebene

Wo kommt die ganze Problematik eigentlich her?

1. Grundbegriffe

2. Tiefere Ursachen

3. Mindset

4. Rechtslage

Das eigentliche Problem

Das eigentliche Problem in der Sicherheit ist der **Mensch**.

Menschliches Fehlverhalten
Das Problem sind die Handlungen, die Menschen aus **Unkenntnis** setzen.

Gestiegene Komplexität
Das Problem ist die stark ansteigende Komplexität von Systemen
verbunden mit mangelnder Präzision in Design und Implementierung.
Wir wünschen uns immer mehr (oft unnötige) *features*
und beherrschen dann die damit entstehende Komplexität nicht (mehr).

Ebene und Meta-Ebene
Turing-vollständige Systeme trennen Ebene und Meta-Ebene nicht strikt.
Die Vermischung der Ebenen erzeugt Probleme.

Social Engineering

Social Engineering

Mißbrauch Gruppen-spezifischer Verhaltensweisen um den Nutzer zu einer Handlung zu bewegen die er unter Kenntnis der damit verbundenen Folgen so nicht setzen würde.

- **Phishing:** Abgreifen vertraulicher Daten durch Vorspiegelung vertrauenswürdiger Umgebung.
- **Spear-Phishing:** Auf bestimmte Person spezifisch zugeschnittene Phishing-Mail.
- **CEO-Fraud:** Angreifer gibt sich als Geschäftsführer und fordert Transaktion an.
- **Vishing:** Phishing per Telefon. Bsp: Anruf eines Bankmitarbeiters.
- **Smishing:** Phishing per SMS oder Messenger. Bsp: Paketbenachrichtigung.
- **Pretexting:** Aufbauen glaubwürdiger Legendem. Bsp: Wirtschaftsprüfer, neuer Kollege.
- **Baiting:** Präparierte USB-Sticks auf Firmenparkplatz.
- **Tailgating:** Einem berechtigten Mitarbeiter durch gesicherte Tür folgen.
- **Quid pro quo:** Eine Gegenleistung anbieten für Zugangsdaten.
- **Watering-Hole:** Detaillierte Beschreibung
- **Fake-Support:** Browser behauptet Problem am Rechner, Hotline sei anzurufen, SW zu installieren.
- **Deepfake:** KI geklonte Stimme täuscht Notlage vor und bittet um Geld.
- **Usw usw usw...**

Beispiel einer Phishing Mail

Sehr geehrter Kunde.

Ihr Konto wurde vorübergehend eingeschränkt,
da wir ungewöhnliche Aktivitäten bei ihrem Zahlungsmittel
festgestellt haben.

Um die Sicherheit und Funktionsfähigkeit Ihres Kontos
wieder herzustellen, müssen Sie sich innerhalb
der nächsten 24 Stunden unter diesem Link einloggen.

Mit freundlichen Grüßen
Hans Müller, Kundenbetreuer.

Phishing Mails

Phishing Mail

Betrügerische E-Mail, mit der Kriminelle versuchen, an vertrauliche Daten zu gelangen. Dabei geben sie sich als vertrauenswürdige Institutionen aus.

Typische Merkmale:

- **Künstlicher Druck:** Drohende Kontosperrung, kurze Fristen..
- **Unpersönliche Anrede:** Allgemeine Formulierungen statt des echten Namens.
- **Falscher Absender:** Abweichung von der echten Unternehmensdomain.
- **Links / Anhänge:** Aufforderungen zum Klicken externer Links / Öffnen infizierter Dateien.

Richtige Reaktion:

- **Nicht klicken/laden:** Keine Links klicken, keine Anhänge herunterladen.
- **Keine Daten** eingeben in ein unbekanntes Ziel.
- **Einloggen:** Nur über die offizielle Website oder App, niemals über den Link.
- **Meldung:** Löschen oder weiterleiten. Siehe: BSI oder Verbraucherzentrale

Identitätsdiebstahl

Identitätsdiebstahl

ist der unbefugte Mißbrauch persönlicher, finanzieller oder sensitiver Daten durch Kriminelle, um im Namen der fremden Identität Straftaten oder betrügerische Geschäfte zu begehen.

Bsp: Name, Geburtsdatum, Adresse, Personalausweisnummer, Kontodaten oder (insbesondere Email) Zugangsdaten

https://twbk.de/2ta6

Wurden Sie schon einmal Opfer eines Identitätsdiebstahls?

- Nein
- Ja: Einmal
- Ja: Mehrmals

Dumpster Diving

Dumpster Diving

bedeutet, im Müll von Personen nach Daten suchen.

- Dokumente mit vertraulichen Kundendaten.
- Alte Festplatten mit unverschlüsselten Daten. Beispiel
- USB-Sticks und Speicherkarten auf ebay.
- Entsorgte Smartphones und Laptops.
- Firmenausweise und Zugangskarten.
- Rechnungen mit Bankverbindungen.
- Personalakten und Bewerbungsunterlagen.
- Netzwerkpläne und IT-Dokumentationen.
- Notizzettel mit PINs oder Passwörtern.
- Entsorgte Backup-Medien.
- Ausdrucke mit internen E-Mails.
- Versandetiketten mit personenbezogenen Daten.
- Alte Router mit gespeicherten Konfigurationen.
- Kopierer mit alten Scan-Daten: Beispiel

Wichtig daher:

- **Encryption at Rest:** Auch gespeicherte Daten müss(t)en verschlüsselt sein.
- **Sichere Entsorgung** und Vernichtung von Datenträgern und IT-Systemen.

Encryption in transit wäre dann die Verschlüsselung bei der Übertragung.

CVEs: Common Vulnerabilities and Exposures

cve.org

cve.org: Öffentliche Datenbank bekannter Sicherheitslücken in Software und Hardware mit eindeutiger Kennung zur zweifelsfreien Identifikation.

- **Betrieb:** 1999 von MITRE ins Leben gerufen, von CISA finanziert.
- **Vergabe:** Durch die *CVE Numbering Authorities* (CNAs)
  Bsp: Software-Hersteller wie Google oder Red Hat.
- **Abgrenzung:** Bewertung des Schweregrads über den CVSS-Score.

Beispiel einer CVE: php2shell (1)

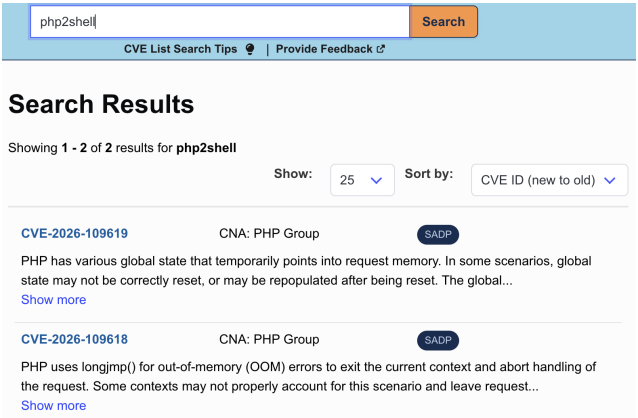
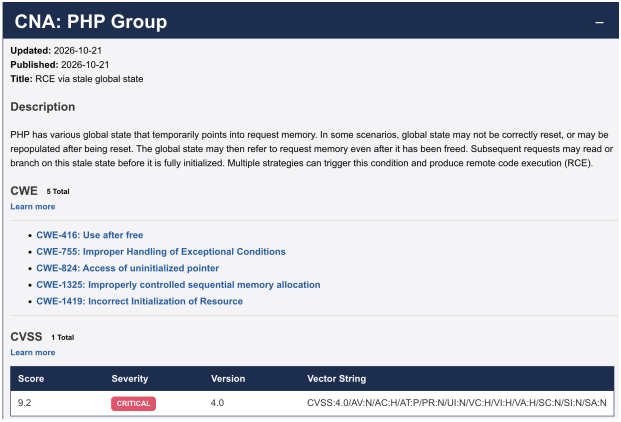


Abb. 16: php2shell in `cve.org` Rechte s. Anhang.

Beispiel einer CVE: php2shell (2)



CNA: PHP Group

Updated: 2026-10-21
Published: 2026-10-21
Title: RCE via stale global state

Description

PHP has various global state that temporarily points into request memory. In some scenarios, global state may not be correctly reset, or may be repopulated after being reset. The global state may then refer to request memory even after it has been freed. Subsequent requests may read or branch on this stale state before it is fully initialized. Multiple strategies can trigger this condition and produce remote code execution (RCE).

CWE 5 Total
Learn more

- CWE-416: Use after free
- CWE-755: Improper Handling of Exceptional Conditions
- CWE-824: Access of uninitialized pointer
- CWE-1325: Improperly controlled sequential memory allocation
- CWE-1419: Incorrect Initialization of Resource

CVSS 1 Total
Learn more

Score	Severity	Version	Vector String
9.2	CRITICAL	4.0	CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Abb. 17: Details zu php2shell Rechte s. Anhang.

CVEs im Zeitverlauf

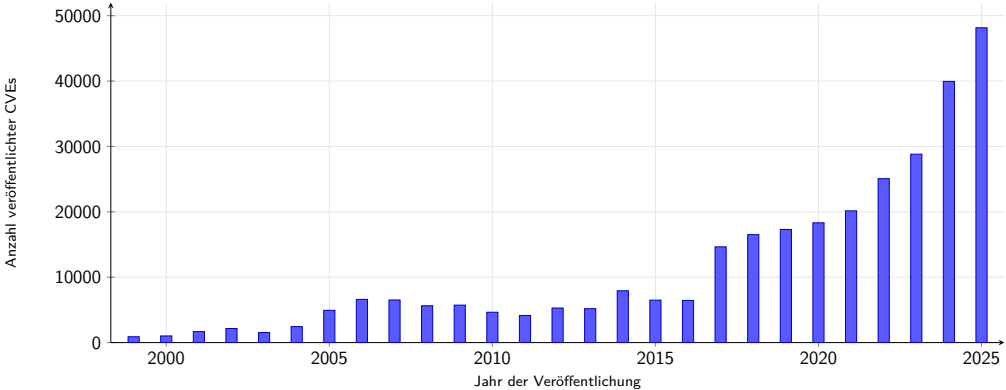


Abb. 18: Anzahl jährlich veröffentlichter CVE-Einträge. Quelle: CVE.ICU, `https://cve.icu/years.html`, Datenstand 30.10.2026. **Ursachen: (1) Zunehmende Software-Komplexität**, **(2) Verfügbarkeit von Angriffs-Werkzeugen**, speziell KI, **(3) zunehmende Vernetzung aller Dinge.** Rechte s. Anhang.

Anstieg der Lines of Code im Linux Kern

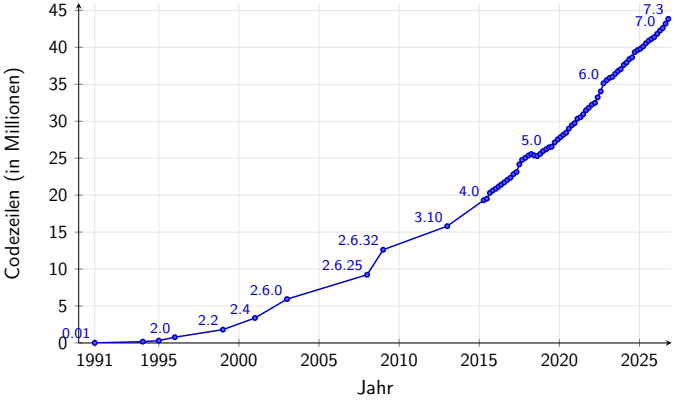


Abb. 19: Lines of Code (LoC) im Linux Kern von 0.01 (1991) bis 7.3 (Oktober 2026). Ab 4.0 (2015) ist jede Hauptversion eingetragen, gezählt im offiziellen Release-Archiv von `https://kernel.org` durch Anthropic/Opus 5.5. Die wachsende Komplexität erzeugt auch eine Mehrzahl von Fehlern. Rechte s. Anhang.

Homeoffice (1)



Abb. 20: Home Office ist nicht in allen Berufen möglich... Rechte s. Anhang.

Homeoffice (2)



Abb. 21: ...aber in der Cyber-Kriminalität ist es definitiv eine Option. Praktisch alle Ziele von Cyber-Kriminalität sind über das Netz erreichbar. Das gilt insbesondere für das **IoT** (**I**nternet **o**f **T**hings). Rechte s. Anhang.

Angriffe auf das IoT: Beispiel Kühlschränke



50 Grad im Kühlschrank: Cyberkriminelle greifen Supermärkte in Belgien und Schweden an

Nach dem Cyberangriff auf über 500 Geschäfte in Frankreich vor drei Wochen trifft es nun weitere EU-Länder. Erneut kapern Hacker die Steuerung von Kühlsystemen.

02.11.2026, 16.38 Uhr 3 Leseminuten

Abb. 22: Cyberangriffe auf Kühlschränke zeigen das Problem der Datensicherheit im **IoT** auf. Hier eine Schlagzeile der Neuen Zürcher Zeitung. Rechte s. Anhang.

Beispiel für SQL-Injection

```
1 $query = SELECT * FROM USERS WHERE user = '$uname';
```

Src. 1: Was beabsichtigt diese Datenbank-Abfrage?

```
1 Username:  '; DROP TABLE USERS - -
```

Src. 2: Dateneingabe des Benutzers in das Web-Formular.

```
1 SELECT * FROM USERS WHERE user=''; DROP TABLE USERS - - '
```

Src. 3: Die sich durch String-Ersetzung ergebende Query. Welche Effekte löst diese Query auf dem DB-Server aus?

Beispiel für HTML-Injection

```
1 <html><body><table>
2 <tr><td>User:</td><td>$username</td></tr>
3 </table></body></html>
```

Src. 4: Ausgabe des Benutzernamens als Teil eines Web-Formulars.

```
1 Hal<br>lo <script>alert("HUHU!");</script></table></body></html>
```

Src. 5: Was passiert, wenn der Benutzer diese String in das Formular als `$username` eingibt?

Problematik

Viele Systeme erzeugen dynamisch Web-Seiten oder Programm-Code.
Sie erhalten dazu Daten vom Client, die sie dann einbauen in:

- SQL Queries
- HTML Seiten
- PHP Programme
- Javascript Programme
- Templates

Problem:

- Programmierer machen implizite Annahmen über Semantik und Syntax.
- Wenn diese verletzt werden entstehen unerwünschte Effekte.
- **Angriff 1:** Unerwartete Eingabe in ein Formular durch User.
- **Angriff 2:** Direktes Erzeugen von Anfragen (URL, Query, header, body)

Vermeidung von HTML-Injection

```
1 var ele = document.getElementById ("ele");
2 ele.innerHTML = "das ist <b>fett</b>";               // zeigt fett an
3 ele.innerHTML = "das ist &lt;b&gt;fett&lt;/b&gt;";  // zeigt HTML Text an
```

Src. 6: HTML-Entities stellen Zeichen durch alternative Techniken dar.

```
1 function addTextNode(text) {
2   var newtext = document.createTextNode(text),
3   p1 = document.getElementById("p1");
4   p1.appendChild(newtext);
5 }
```

Src. 7: DOM-Funktionen erzeugen ausdrücklich nicht-interpretierte Elemente eines Douments.

Vermeiden von SQL-Injection

Lange Liste von Techniken:

- Escapen von Quotes (DB-abhängig).
- Nutzung von Query-Buildern.
- Vermeiden von String Interpolation.
- Typ-Prüfung (String, Name, PLZ)
- Eingabe-Validierung.
- Prepared Statements verwenden.
- PHP Data Objects (PDO) nutzen.
- Separate DB-user für Wartung und Produktion verwenden.
- SQL-seitig reduzierte Rechte für den DB-user vorsehen.

```
1 $stmt = $dbh->prepare ("INSERT INTO user (first, last) VALUES (:fn, :ln)");
2 $stmt -> bindParam(':fn', 'John');  // Korrekte Behandlung von Quotes im Wert
3 $stmt -> bindParam(':ln', 'Doe');   // ist in bindParam gesichert.
4 $stmt -> execute();
```

Src. 8: Prepared Statements.

SQL-Injection nach xkcd

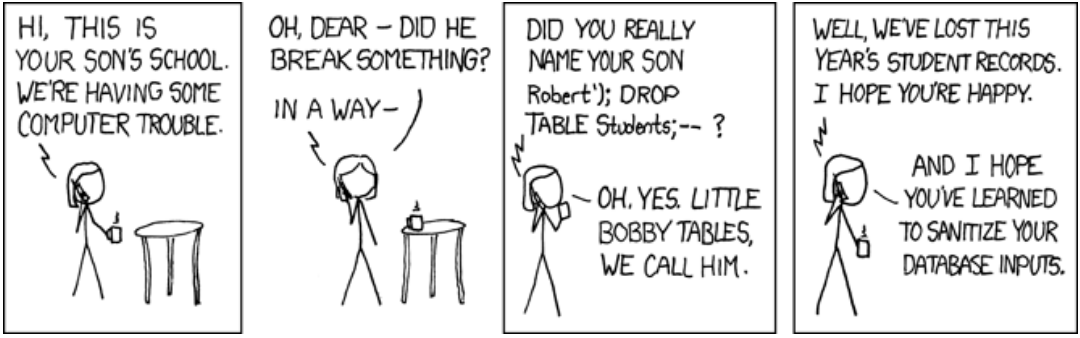


Abb. 23: xkcd327 zu SQL-Injection. Rechte s. Anhang.

SQL-Injection

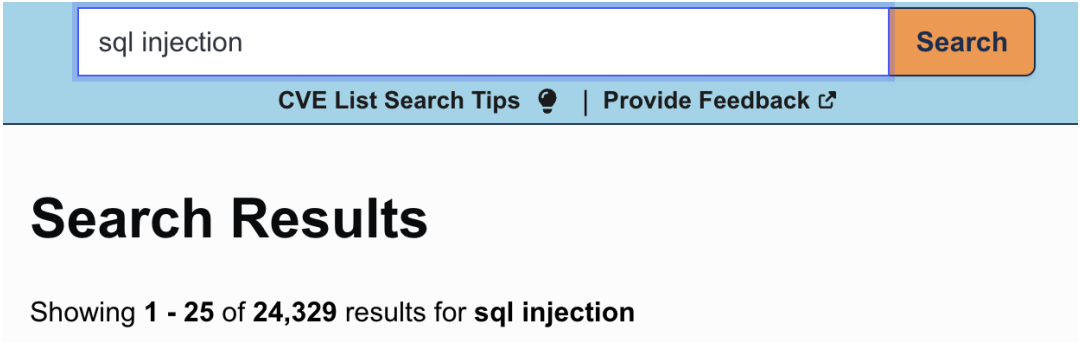


Abb. 24: Die SQL-Injection ist nach wie vor eines der häufigsten Probleme in Web-basierter Software. Rechte s. Anhang.

3. Mindset

Das richtige mentale Modell ist wichtig.

1. Grundbegriffe

2. Tiefere Ursachen

3. Mindset

4. Rechtslage

Was will der Angreifer?

1. Gewinn erzielen
2. Schaden anrichten
3. Rache nehmen
4. Informationen erlangen
5. Aufmerksamkeit bekommen
6. Verhalten provozieren
7. Nichts. Er handelt ziellos.

Beispiel: Bombendrohung in Kleinstadt (1)

Bei einer Bank in Kleinstadt geht ein Anruf ein.
Alle raus – in 10 Minuten geht eine Bombe hoch.

Beispiel: Bombendrohung in Kleinstadt (2)

Bei einer Bank in Kleinstadt geht ein Anruf ein.
Alle raus – in 10 Minuten geht eine Bombe hoch.

Kleinstadt mobilisiert alle Kräfte, die sie haben.

Beispiel: Bombendrohung in Kleinstadt (3)

Bei einer Bank in Kleinstadt geht ein Anruf ein.
Alle raus – in 10 Minuten geht eine Bombe hoch.

Kleinstadt mobilisiert alle Kräfte, die sie haben.

60 Minuten gespanntes Warten.

Beispiel: Bombendrohung in Kleinstadt (4)

Bei einer Bank in Kleinstadt geht ein Anruf ein.
Alle raus – in 10 Minuten geht eine Bombe hoch.

Kleinstadt mobilisiert alle Kräfte, die sie haben.

60 Minuten gespanntes Warten.

Dann der erlösende Anruf: Keine Bombe.

Beispiel: Bombendrohung in Kleinstadt (5)

Bei einer Bank in Kleinstadt geht ein Anruf ein.
Alle raus – in 10 Minuten geht eine Bombe hoch.

Kleinstadt mobilisiert alle Kräfte, die sie haben.

60 Minuten gespanntes Warten.

Dann der erlösende Anruf: Keine Bombe.

und danke, daß wir in Ruhe den Juweliersladen ausräumen konnten.

Das Ziel des Angreifers war die Provokation einer vorhersehbaren Reaktion.

James Bond und die zwei iPhones (1)



Abb. 25: Ein iPhone.

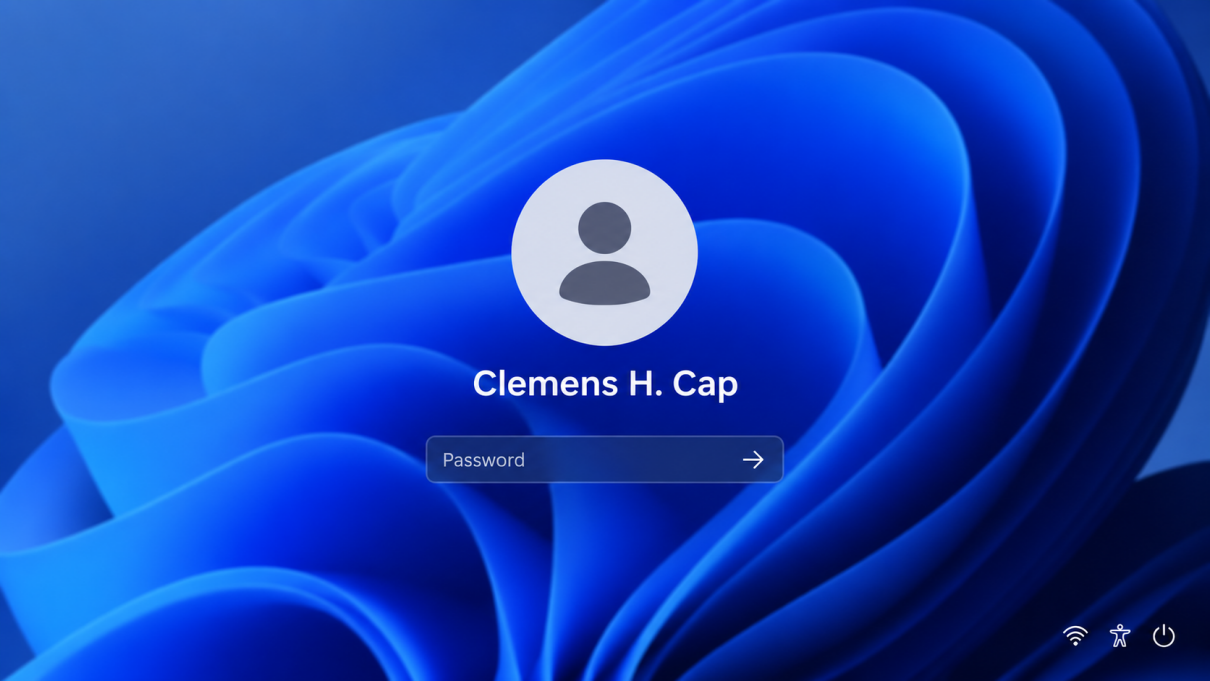
James Bond und die zwei iPhones (2)



Abb. 26: Zwei iPhones. Die volle Anekdote dazu wurde in der Vorlesung erzählt.

Jetzt bitte Tweedback starten...

https://twbk.de/2ta6


Clemens H. Cap
Password

Noch ein Quiz als Fazit

Vertrauen in die Ein / Ausgabe meines Rechners erfordert Vertrauen in

1. das Anzeige-Programm
2. das Betriebssystem
3. den Bildschirm-Treiber
4. das System-Update
5. jedes jemals installierte Programm
6. die Firmware des Motherboards



https://twbk.de/2ta6

Ein Beispiel: Cursor Redressing

Cursor Redressing

Cursor Jacking

Hinweis: Auch hier erfolgte die Vorführung live und die Erklärung dazu direkt in der Vorlesung.

Fazit über den ganzen Abschnitt

Sicherheit hat nicht nur mit Verschlüsselung allein zu tun.

Sicherheit hat viel zu tun mit

- Verhaltensweisen
- Modellvorstellungen
- Annahmen
- Identitäten
- Vertrauen

Vieles davon entzieht sich einer ingenieurwissenschaftlichen Analyse.

4. Rechtslage

Einige einschlägige deutsche Gesetze dazu.
Disclaimer: Keine Rechtsberatung.
Fazit: Nicht alles was spannend ist & geht ist rechtlich auch erlaubt.

1. Grundbegriffe
2. Tiefere Ursachen
3. Mindset
4. Rechtslage

Rechtliche Situation (1)

Strafgesetzbuch (StGB)
§ 202a Ausspähen von Daten

(1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

Strafgesetzbuch (StGB)
§ 202b Abfangen von Daten

Wer unbefugt sich oder einem anderen unter Anwendung von technischen Mitteln nicht für ihn bestimmte Daten (§ 202a Abs. 2) aus einer nichtöffentlichen Datenübermittlung oder aus der elektromagnetischen Abstrahlung einer Datenverarbeitungsanlage verschafft, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft, wenn die Tat nicht in anderen Vorschriften mit schwererer Strafe bedroht ist.

Abb. 27: Das deutsche Strafrecht kennt **einschlägige Tatbestände**. Wichtig: “Gesichert” bedeutet dem Juristen nicht das, was der Informatiker als “gesichert” bezeichnet. Ein untaugliches, schlechtes Türschloß darf man nicht aufbrechen, eine untaugliche, schlechte informatische Absicherung darf man ebenso nicht überwinden. Die Ausrede “war ja nicht *wirklich* gesichert” gilt nicht! Rechte s. Anhang.

Rechtliche Situation (2)

Strafgesetzbuch (StGB)
§ 202c Vorbereiten des Ausspähens und Abfangens von Daten

(1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er

1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist,

herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.

(2) § 149 Abs. 2 und 3 gilt entsprechend.

Abb. 28: Der sogenannte **Hacker-Paragraph** bestraft die Herstellung entsprechender Programme bei Beabsichtigung einer Straftat. Der Paragraph gilt als umstritten, da der Nachweis von Absicht juristisch oft heikel ist. Rechte s. Anhang.

Rechtliche Situation (3)

Strafgesetzbuch (StGB)
§ 202d Datenhehlerei

(1) Wer Daten (§ 202a Absatz 2), die nicht allgemein zugänglich sind und die ein anderer durch eine rechtswidrige Tat erlangt hat, sich oder einem anderen verschafft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, um sich oder einen Dritten zu bereichern oder einen anderen zu schädigen, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Die Strafe darf nicht schwerer sein als die für die Vortat angedrohte Strafe.

(3) Absatz 1 gilt nicht für Handlungen, die ausschließlich der Erfüllung rechtmäßiger dienstlicher oder beruflicher Pflichten dienen. Dazu gehören insbesondere

1. solche Handlungen von Amtsträgern oder deren Beauftragten, mit denen Daten ausschließlich der Verwertung in einem Besteuerungsverfahren, einem Strafverfahren oder einem Ordnungswidrigkeitenverfahren zugeführt werden sollen, sowie
2. solche beruflichen Handlungen der in § 53 Absatz 1 Satz 1 Nummer 5 der Strafprozessordnung genannten Personen, mit denen Daten entgegengenommen, ausgewertet oder veröffentlicht werden.

Abb. 29: Der sogenannte **Wikileaks-Paragraph** bestraft die Weitergabe widerrechtlich erlangter Daten bei Absicht der Bereicherung oder Schädigung. Rechte s. Anhang.

Appendix

Übersicht

Übersicht		70
Literaturverzeichnis		71
Programmquellenverzeichnis	Prog	72
Verzeichnis aller Abbildungen	Abb	73
Rechtsnachweise	©	76
Rechtliche Hinweise	§	78
Zitierweise dieses Dokuments	→	80
Verzeichnis aller Folien		81

Literaturverzeichnis

[BLN16] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. “Dual EC: A Standardized Back Door”. In: *The New Codebreakers*. Ed. by Peter Y. A. Ryan, David Naccache, and Jean-Jacques Quisquater. Vol. 9100. Lecture Notes in Computer Science. Springer, 2016, pp. 256–281. DOI: 10.1007/978-3-662-49301-4_17. URL: https://doi.org/10.1007/978-3-662-49301-4_17 (cit. on p. 22).

Programmquellenverzeichnis

1 SQL Injection (1) 44
2 SQL Injection (2) 44
3 SQL Injection (3) 44
4 HTML Injection (1) 45
5 HTML Injection (2) 45
6 Nutzung von HTML-Entities 47
7 Nutzung von DOM-Funktionen 47
8 Prepared Statements 48

Verzeichnis aller Abbildungen (1/3)

1 Linux Pinguin 8
2 Adobe 1 13
3 Adobe 2 14
4 XKCD 1553 15
5 XKCD 538 16
6 De Mail und Ende-zu-Ende Verschlüsselung 18
7 Nachschlüssel bei 3G und 4G 19
8 Zweite, unabhängige Quelle. 20
9 Dual EC: A Standardized Back Door 22

10 Falscher öffentlicher Schlüssel (1) 24
11 Falscher öffentlicher Schlüssel (2) 24
12 Falscher öffentlicher Schlüssel (3) 25
13 Falscher öffentlicher Schlüssel (4) 25
14 Falscher öffentlicher Schlüssel (5) 26
15 Falscher öffentlicher Schlüssel (6) 26
16 php2shell in cve.org 37
17 Details zu php2shell 38
18 CVEs im Zeitverlauf 39
19 LoC im Linux Kern 1991–2026 40
20 Home Office Straßenarbeiter 41

21 Home Office Cyber-Kriminalität 42
22 Cyberangriffe auf Kühlschränke 43
23 xkcd327 zu SQL-Injection. 49
24 Häufigkeit von SQL-Injection 50
25 Ein iPhone. 58
26 Zwei iPhones. Die volle Anekdote dazu wurde in der Vorlesung erzählt. 59
27 StGB Par 202a,b 66
28 StGB Par 202c: Hacker-Paragraph 67
29 StGB Par 202d: Wikileaks-Paragraph 68

Rechtsnachweise (1/2)

Abb. 1 Quelle: https://nds.wikipedia.org/wiki/Bild:Tux.svg Vom Autor zur Nutzung freigegeben.

Abb. 2 Quelle: https://x.com/jupenur/status/911286403434246144/photo/1

Abb. 3 Quelle: https://x.com/jupenur/status/911286403434246144/photo/2

Abb. 4 xkcd Lizenz

Abb. 5 xkcd Lizenz

Abb. 6 Eigener Screenshot von der Informationsseite der GI. https://gi.de/meldung/de-mail-fuer-schriftverkehr-mit-behoerden-muss-standardmaessig-ende-zu-ende-verschluesselt-werden

Abb. 7 Quelle: https://de.finance.yahoo.com/nachrichten/snowden-papier-nsa-und-gchq-stahlen-schl%C3%BCssel-zu-064512698--finance.html

Abb. 8 Quelle: https://time.com/3722150/nsa-sim-cards/

Abb. 9 Screenshot des Papiers, gedeckt durch Zitate-Recht

Abb. 10 Eigenes Diagram mit xymatrix

Abb. 11 Eigenes Diagram mit xymatrix

Abb. 12 Eigenes Diagram mit xymatrix

Abb. 13 Eigenes Diagram mit xymatrix

Abb. 14 Eigenes Diagram mit xymatrix

Abb. 15 Eigenes Diagram mit xymatrix

Abb. 16 Eigener Screenshot von cve.org

Abb. 17 Eigener Screenshot von cve.org

Abb. 18 Eigene Abbildung nach Daten aus dem Netz.

Abb. 19 Grafik erstellt durch Anthropic/Claude

Abb. 20 Eigene Karikatur, visualisiert mit

Abb. 21 Eigene Karikatur, visualisiert mitEigenes Bild, Visualisierung durch ChatGPT, AI

Abb. 22 Neue Zürcher Zeitung, 2. November 2026, Ausschnitt gemäß UrhG Par 51, Quelle

Abb. 23 Source: https://xkcd.com/327/ lizensiert nach http://creativecommons.org/licenses/by-nc/2.5/

Abb. 24 Eigener Screenshot

Abb. 27 Screenshot StGB

Abb. 28 Screenshot StGB

Abb. 29 Screenshot StGB

Rechtliche Hinweise (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das Zitatrecht in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.
Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise Par 60a UrhG ermöglichen.
Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.
Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.
Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions https://github.com/logos to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Rechtliche Hinweise (2)

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.
Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 (Pressemitteilung, Blog-Beitrag, Urteilstext). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.
Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben.
Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.
Weitere Hinweise finden Sie im Netz hier und hier oder hier.

Zitierweise dieses Dokuments

Wenn Sie Inhalte aus diesem Werk nutzen oder darauf verweisen wollen, zitieren Sie es bitte wie folgt:

Clemens H. Cap: Propädeutik der Datensicherheit. Electronic document, 9. 11. 2026
https://iuk.one/DS-PROP.pdf

Bibtex Information: https://iuk.one/DS-PROP.bib

```
@misc{doc:DS-PROP,
    author       = {Clemens H. Cap},
    title        = {Propädeutik der Datensicherheit},
    year         = {2026},
    month        = {11},
    howpublished = {Electronic document},
    url          = {https://iuk.one/DS-PROP.pdf }
}
```

Typographic Information:
Typeset on 2026-11-09 at 16:53
This is pdfTeX, Version 3.141592653-2.6-1.40.29 (TeX Live 2026) kpathsea version 6.4.2
This is pgf in version 3.1.11a
This is preamble-slides.tex ©C.H.Cap
Jobtype flag=foreground
Active modes (if any): slides, biblio
Keywords: keywords
Subject: subject

Verzeichnis aller Folien

Titelseite . . . 1
Scheinbar ganz einfach... . . . 2
Übersicht . . . 3

1. Grundbegriffe . . . 4

Grundbegriffe der Verschlüsselung . . . 5
Grundbegriffe der Digitalen Unterschrift . . . 6
Ein Beispiel . . . 7
◯ . . . 8
Symmetrische Verfahren . . . 9
Asymmetrische Verfahren . . . 10
Hybride Verfahren . . . 11
Ciphertext? . . . 12
Problem 1: Falscher Umgang mit dem Schlüssel (1) . . . 13
Problem 1: Falscher Umgang mit dem Schlüssel (2) . . . 14
Problem 1: Falscher Umgang mit dem Schlüssel (3) . . . 15
Problem 2: Falsches Angreifermodell . . . 16
Problem 3: Fehlende echte Ende-zu-Ende Verschlüsselung (1) . . . 17
Problem 3: Fehlende echte Ende-zu-Ende Verschlüsselung (2) . . . 18
Problem 4: Jemand hat einen Nachschlüssel (1) . . . 19
Problem 4: Jemand hat einen Nachschlüssel (2) . . . 20
Problem 4: Jemand hat einen Nachschlüssel (3) . . . 21
Problem 5: Der Schlüsselgenerator hat ein Problem . . . 22
Problem 6: Falscher öffentlicher Schlüssel benutzt (1) . . . 23
Problem 6: Falscher öffentlicher Schlüssel benutzt (2) . . . 24
Problem 6: Falscher öffentlicher Schlüssel benutzt (3) . . . 25
Problem 6: Falscher öffentlicher Schlüssel benutzt (4) . . . 26
Problem 6: Falscher öffentlicher Schlüssel benutzt (5) . . . 27
Die Personen der Kryptographie . . . 28

2. Tiefere Ursachen . . . 29

Das eigentliche Problem . . . 30

2.1. Menschliches Fehlverhalten

Social Engineering . . . 31
Beispiel einer Phishing Mail . . . 32
Phishing Mails . . . 33
Identitätsdiebstahl . . . 34
Dumpster Diving . . . 35

2.2. Gestiegene Komplexität

CVEs: Common Vulnerabilities and Exposures . . . 36
Beispiel einer CVE: php2shell (1) . . . 37
Beispiel einer CVE: php2shell (2) . . . 38
CVEs im Zeitverlauf . . . 39
Anstieg der Lines of Code im Linux Kern . . . 40
Homeoffice (1) . . . 41
Homeoffice (2) . . . 42
Angriffe auf das IoT: Beispiel Kühlschränke . . . 43

2.3. Ebene und Meta-Ebene
Beispiel für SQL-Injection ... 44
Beispiel für HTML-Injection ... 45
Problematik ... 46
Vermeidung von HTML-Injection ... 47
Vermeiden von SQL-Injection ... 48
SQL-Injection nach xkcd ... 49
SQL-Injection ... 50

3. Mindset ... 51
Was will der Angreifer? ... 52
Beispiel: Bombendrohung in Kleinstadt (1) ... 53
Beispiel: Bombendrohung in Kleinstadt (2) ... 54
Beispiel: Bombendrohung in Kleinstadt (3) ... 55
Beispiel: Bombendrohung in Kleinstadt (4) ... 56
Beispiel: Bombendrohung in Kleinstadt (5) ... 57
James Bond und die zwei iPhones (1) ... 58
James Bond und die zwei iPhones (2) ... 59
Jetzt bitte Tweedback starten... ... 60
... 61
Noch ein Quiz als Fazit ... 62
Ein Beispiel: Cursor Redressing ... 63
Fazit über den ganzen Abschnitt ... 64

4. Rechtslage ... 65
Rechtliche Situation (1) ... 66
Rechtliche Situation (2) ... 67
Rechtliche Situation (3) ... 68

Legende:
Fortsetzungsseite
Seite ohne Überschrift
Bildseite