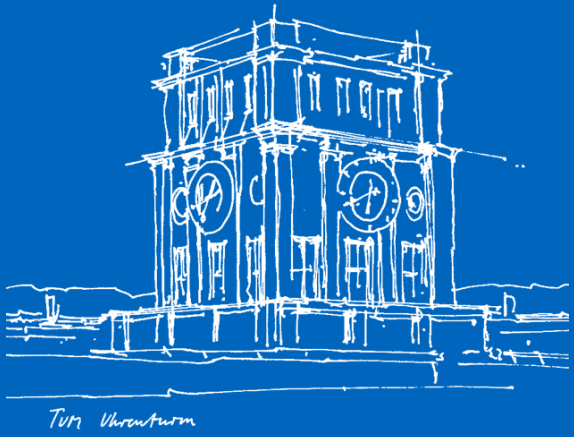
IT-Sicherheit Übung 1 - Web Application Security

Johann Höpfner

Chair of IT Security
Department of Computer Science
Technische Universität München

20. Oktober 2025



Intro

Wichtig! *Bitte setzt die von uns vermittelten Angriffstechniken nur bei euren* ***eigenen*** *Systemen oder bei Systemen für die Ihr* ***eine Erlaubnis des Betreibers*** *erhalten habt ein. Das unerlaubte Eindringen in IT-Systeme ist nach § 202a StGB ein* ***Straftatbestand****. Auch sind zivilrechtliche Schadenersatzansprüche möglich. Solltet Ihr über eine Sicherheitslücke in einem IT-System stolpern, so gebietet die Ethik einen Coordinated Disclosure Prozess mit dem Betreiber des Systems!*

1) System Setup

`https://scoreboard.sec.in.tum.de`

Sandkasten

SSH-Public Key hochladen und mit darunter angezeigtem Befehl auf Sandkasten anmelden

```
$ ssh-keygen -t ed25519
```

(Der Public Key liegt beispielsweise unter `~/.ssh/id_ed25519.pub`).

Dateitransfer mit `scp` / `rsync`

Lokale Installation

Python installieren

2) Begrifflichkeiten

a) Grenzen Sie kurz die Begriffe *Safety* und *Security* voneinander ab. Worin unterscheiden sie sich?

2) Begrifflichkeiten

a) Grenzen Sie kurz die Begriffe *Safety* und *Security* voneinander ab. Worin unterscheiden sie sich?

Safety: Funktions- und Betriebssicherheit

Erkennen und Abwehr von Störungen, die die korrekte Funktionalität (Betriebssicherheit) beinträchtigen

Störungen kommen **von Innen**

2) Begrifflichkeiten

a) Grenzen Sie kurz die Begriffe *Safety* und *Security* voneinander ab. Worin unterscheiden sie sich?

Safety: Funktions- und Betriebssicherheit

Erkennen und Abwehr von Störungen, die die korrekte Funktionalität (Betriebssicherheit) beinträchtigen

Störungen kommen **von Innen**

Security: Daten- und Informationssicherheit

Verwundbarkeit von zu schützenden Werten systematisch reduzieren

Bewahren eines Systems vor Beeinträchtigung und Missbrauch durch Angriffe

2) Begrifflichkeiten

a) Grenzen Sie kurz die Begriffe *Safety* und *Security* voneinander ab. Worin unterscheiden sie sich?

Safety: Funktions- und Betriebssicherheit

Erkennen und Abwehr von Störungen, die die korrekte Funktionalität (Betriebssicherheit) beinträchtigen

Störungen kommen **von Innen**

Security: Daten- und Informationssicherheit

Verwundbarkeit von zu schützenden Werten systematisch reduzieren

Bewahren eines Systems vor Beeinträchtigung und Missbrauch durch Angriffe

Es gibt Wechselwirkungen zwischen Security und Safety

Aufwändige Zugriffskontrollen (für Security) können im Notfall die Safety beeinträchtigen

Redundanz (erhöht Safety) bringt zusätzliche Angriffsmöglichkeiten

2) Begrifflichkeiten

b) Beschreiben Sie die Begriffe **Schwachstelle**, **Bedrohung** und **Angriffsvektor**

2) Begrifflichkeiten

b) Beschreiben Sie die Begriffe **Schwachstelle**, **Bedrohung** und **Angriffsvektor**

Schwachstelle: ermöglicht es, dass die Sicherheitskontrollen des Systems umgangen oder getäuscht werden können

Beispiel: das System erlaubt schwache Passwörter

2) Begrifflichkeiten

b) Beschreiben Sie die Begriffe **Schwachstelle**, **Bedrohung** und **Angriffsvektor**

Schwachstelle: ermöglicht es, dass die Sicherheitskontrollen des Systems umgangen oder getäuscht werden können

Beispiel: das System erlaubt schwache Passwörter

Bedrohung: Ein Umstand oder Ereignis mit dem **Potenzial**, ein System durch unbefugten Zugriff, Zerstörung, Offenlegung, etc. zu beeinträchtigen.

Beispiel: DDoS-Attack

2) Begrifflichkeiten

b) Beschreiben Sie die Begriffe **Schwachstelle**, **Bedrohung** und **Angriffsvektor**

Schwachstelle: ermöglicht es, dass die Sicherheitskontrollen des Systems umgangen oder getäuscht werden können

Beispiel: das System erlaubt schwache Passwörter

Bedrohung: Ein Umstand oder Ereignis mit dem **Potenzial**, ein System durch unbefugten Zugriff, Zerstörung, Offenlegung, etc. zu beeinträchtigen.

Beispiel: DDoS-Attack

Angriffsvektor: Ein **konkreter** Angriffsweg, der eine oder mehrere Schwachstellen ausnutzt, um die Sicherheit eines Systems zu gefährden

Beispiel: Eve snifft Alice' Passwort (Schwachstelle: unverschlüsselte Übertragung) und gibt sich dann als Alice aus

3) SQL-Injections

a) Erläutern Sie möglichst allgemein, was eine Injection Vulnerability ist, und nennen Sie Beispiele für Stellen, an welchen eine solche Schwachstelle vorkommen kann!

3) SQL-Injections

a) Erläutern Sie möglichst allgemein, was eine Injection Vulnerability ist, und nennen Sie Beispiele für Stellen, an welchen eine solche Schwachstelle vorkommen kann!

Nicht validierte Daten werden von einem Interpreter als Bestandteil einer Anfrage verarbeitet, um z.B. **Kommandos auszuführen** oder die **Semantik** zu verändern.

3) SQL-Injections

a) Erläutern Sie möglichst allgemein, was eine Injection Vulnerability ist, und nennen Sie Beispiele für Stellen, an welchen eine solche Schwachstelle vorkommen kann!

Nicht validierte Daten werden von einem Interpreter als Bestandteil einer Anfrage verarbeitet, um z.B. **Kommandos auszuführen** oder die **Semantik** zu verändern.

Beispiele: SQL Injection in SQL-Abfrage, XSS (unsicherer Webserver erlaubt Einfügen von Schadcode in Website)

3) SQL-Injections

b) Schauen Sie sich das folgende Beispiel zu SQL-Injections an. Gehen Sie davon aus, dass es aus einem fiktiven Nutzerverwaltungsprogramm stammt. Wie geht der Angreifer vor und was ist im gegebenen Beispiel sein Ziel? SQL-Query zur Suche von Nutzern:

```
name  = input()
query = "SELECT * FROM users WHERE vorname='" + name +   "' AND id > 1000"
```

Angriff: `' OR 1=1 --`

id	vorname	nachname	email	pwhash
1	Fabian	Franzen	franzen@sec.in.tum.de	$2b$12$5xuB/x9oU...
2	Claudia	Eckert	claudia.eckert@in.tum.de	$2b$12$MqIWEYxOm...
...	...	...	...	...

3) SQL-Injections

c) In der Datenbanklösung SQLite3 existiert eine Tabelle `sqlite_master`, welche Informationen über die existierenden Tabellen der Datenbank enthält. Dies kann nützlich sein, wenn Ihnen nicht bekannt ist, welche Datenbanktabellen eine Anwendung angelegt hat (z.B. weil Ihnen dessen Implementierung unbekannt ist).
Wie würden Sie die SQL-Injection aus der vorherigen Aufgabe verändern, um diese Informationen auszulesen?

3) SQL-Injections

c) In der Datenbanklösung SQLite3 existiert eine Tabelle `sqlite_master`, welche Informationen über die existierenden Tabellen der Datenbank enthält. Dies kann nützlich sein, wenn Ihnen nicht bekannt ist, welche Datenbanktabellen eine Anwendung angelegt hat (z.B. weil Ihnen dessen Implementierung unbekannt ist).
Wie würden Sie die SQL-Injection aus der vorherigen Aufgabe verändern, um diese Informationen auszulesen?

Mittels einer SQL-Injection kann via `UNION` der Inhalt der Metatabelle `sqlite_master` zusätzlich zu dem Inhalt von `users` ausgegeben werden.

3) SQL-Injections

d) In einigen Fällen werden die Ergebnisse einer verwundbaren Datenbankabfrage überhaupt nicht ausgegeben, sondern man erhält von der verwundbaren Webseite nur eine *richtig* oder *falsch* Meldung. Entwickeln Sie eine Methode, um trotzdem Informationen zu extrahieren.

3) SQL-Injections

d) In einigen Fällen werden die Ergebnisse einer verwundbaren Datenbankabfrage überhaupt nicht ausgegeben, sondern man erhält von der verwundbaren Webseite nur eine *richtig* oder *falsch* Meldung. Entwickeln Sie eine Methode, um trotzdem Informationen zu extrahieren.

Eine Extraction ist mittels Blind-SQL-Injection möglich, indem der gesuchte Wert zeichenweise per Brute-Force bestimmt wird. Ein Beispiel bei der folgende verwundbare Abfrage gegeben ist `SELECT * FROM users WHERE id='INJECTION'`:

Injection	**Ergebnis**
`1' AND 'A'=SUBSTR(vorname, 1, 1) --`	Falsch
⋮	⋮
`1' AND 'F'=SUBSTR(vorname, 1, 1) --`	Wahr
`1' AND 'FA'=SUBSTR(vorname, 1, 2) --`	Wahr
⋮	⋮