

# Software Security

Kryptographie und IT Sicherheit SS 2018

---

Dimitrii ,Manuel Klappacher

Universität Salzburg

1. Einleitung
2. Remote and Lokale Gefahren
3. Exploits
4. Open Source und Propertäre Software
5. Firmware Security

# Einleitung

---

test frame

# Remote and Lokale Gefahren

---

# Exploits

---

Code Injection ist das ausnutzen von Bugs durch Eingabe von ungewollten Parametern, um dadurch die Ausführung zu verändern. Kann folgende Auswirkungen haben:

- Daten in SQL Tabellen verndern
- Installieren von Malware durch Server-Scripting Code zB. PHP
- Root Privilegien bekommen, durch Shell Injection oder Windows Service
- Angtiff auf Web User durch Cross-Site-Scripting in HTML/JS

Kann erschwert werden durch:

- API's benutzen, die sicher gegenüber allen Symbolen sind, indem der Eingabestring compiliert und gefiltert wird.
- Whitelisting von erwünschten Parametern



Ausnutzen von Sicherheitslücken in Zusammenhang mit SQL-Datenbanken. Ziele:

- Daten auszuspähen oder zu verändern
- Kontrolle über Server zu erhalten

# SQL Injection - Beispiel

Es wird zusätzlicher Code bei Aufruf eingeschleust, der die Bentzertabelle modifiziert.

Erwarteter Aufruf	
Aufruf	http://webserver/cgi-bin/find.cgi?ID=42
Erzeugtes SQL	SELECT author, subject, text FROM artikel WHERE ID=42;
SQL-Injection	
Aufruf	http://webserver/cgi-bin/find.cgi? ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=23
Erzeugtes SQL	SELECT author, subject, text FROM artikel WHERE ID=42;UPDATE USER SET TYPE="admin" WHERE ID=23;

test frame

# Cross Site Scripting - reflektierte Angriffe

Eine Benutzereingabe wird direkt vom Server wieder zurück gesendet. Wenn diese Eingabe Scriptcode enthält, die vom Browser des Nutzers interpretiert wird, kann dort Schadcode ausgeführt werden. Beispiel: Suchfunktion.

```
http://example.com/?suche=Suchbegriff  
http://example.com/?suche=<script type="text/javascript">alert("XSS")</script>  
<p>Sie suchten nach: <script type="text/javascript">alert("XSS")</script></p>
```

Ausgenutzt wird das dynamisch generierte Websites ihren Inhalt an übergebene Eingabewerte anpassen, durch HTTP-GET und HTTP-POST. Dieser Typ heisst auch nicht-persistent, da der Schadcode nur temporär bei der jeweiligen Generierung der Website eingeschleust wird.

Unterscheidet sich von reflektierenden Angriffen nur dadurch, dass der Schadcode auf dem Server gespeichert wird, wodurch er bei jeder Anfrage ausgeführt wird. Ist bei Webanwendungen möglich, die Benutzereingaben serverseitig ohne Prüfung speichern und diese später wieder ausliefert. Beispiel Posting auf Website:

```
Eine sehr gutes Produkt!<script type="text/javascript">alert("XSS")</script>
```

Webapplikation auf dem Server ist hier nicht beteiligt, wird auch lokales XSS genannt. Somit auch statische HTML Seiten mit JavaScript unterstützung anfällig für diesen Angriff.

- Anstatt Blacklist mit bösen Eingaben zu führen, besser Whitelist mit guten Eingaben. Da die Anzahl der Angriffsmethoden nicht bekannt ist.
- HTML-Metazeichen durch Zeichenreferenzen ersetzen, damit sie als normale Zeichen behandelt werden
- Sicher programmierte Anwendung sind Web Application Firewalls (WAF) vorzuziehen.

Ein HTTP Angriff, bei dem ein Angreifer zugriff auf gesperrte Verzeichnisse gewinnt und Code auserhalb des root Verzeichnisses ausführt.



test frame

test frame

test frame

# Buffer Overflows - Type-Safe Sprachen

Compiler stellt Typsicherheit her, indem Datentypen geprüft werden, damit keine Typverletzungen entstehen. Wenn Typverletzungen spätestens zur Laufzeit erkannt werden, spricht man von Typsicheren Programmiersprachen.

Beispiel String in Python, es reicht der Variable einen String zuzuweisen.

```
1 mystring = "This is my string"
```

```
2
```

Beispiel in C, es muss der Typ deklariert und auch der Speicher manuell reserviert werden.

```
1 char mystring[20] = "This is my string";
```

```
2
```

Wenn man in C nun einen 30 Byte String zuweist entsteht eine Overflow Situation.

- Type-Safe Programmiersprachen verwenden, welche Memory Management zB Java, Python, Ruby,...
- Überprüfen auf Overflows bei User Eingaben
- in C sichere Methoden verwenden, *get\_s* anstatt *get*.

# Open Source und Propertäre Software

---

# Firmware Security

---