

Übungsblatt 1 Gruppe 2

Stand: 16. April 2018 Version: 1.0

1 Footprinting über das Internet

Ausgewählte Hochschule: FH-Bielefeld - University of Applied Sciences

Webserver: Apache/2.4.10 (Linux/SUSE)

Bekannte Schwachstelle: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3523>

Listing 1: Ermittlen des Webservers

```
#!/bin/bash
curl -I -s https://www.fh-bielefeld.de | grep Server
```

2 Netzwerkdaten analysieren

1. Username: gurpartap@patriots.in
2. Passwort: punjab@123
3. Betreff: SMTP
4. Sender: gurpartap@patriots.in
5. Empfänger: raj_deol2002in@yahoo.co.in
6. Body*:

Hello

I send u smtp pcap file

Find the attachment

GPS

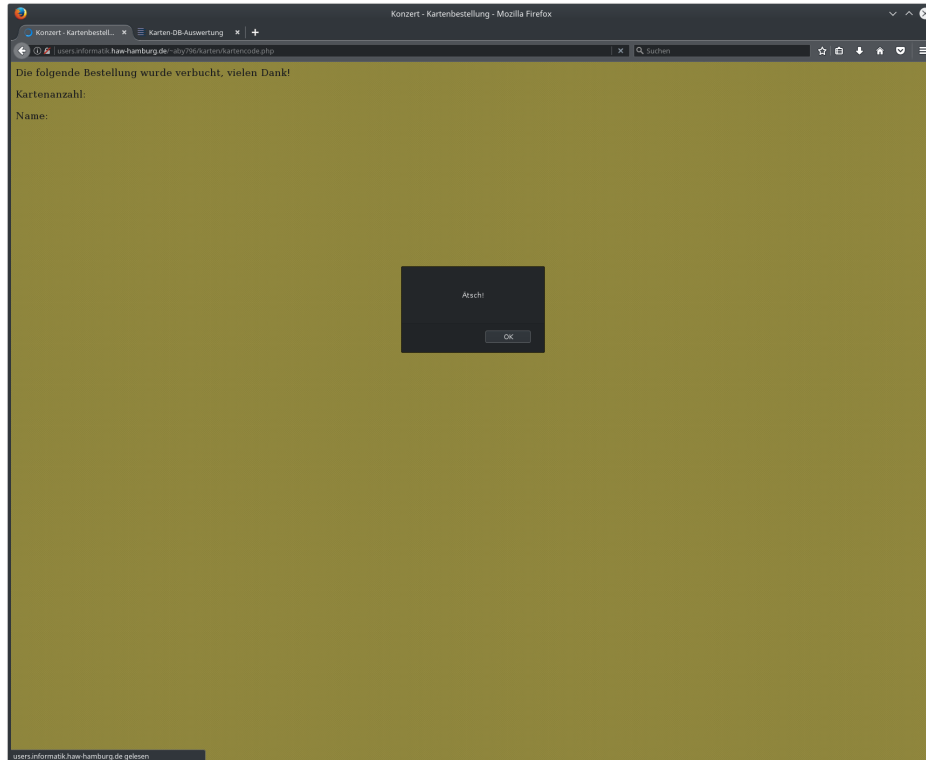
*Unnötige Leerzeilen entfernt

3 XSS / SQL-Injection

3.1 XSS - Ausführung

Folgendes Code-Snippet in einer der Input-Boxen einfügen:

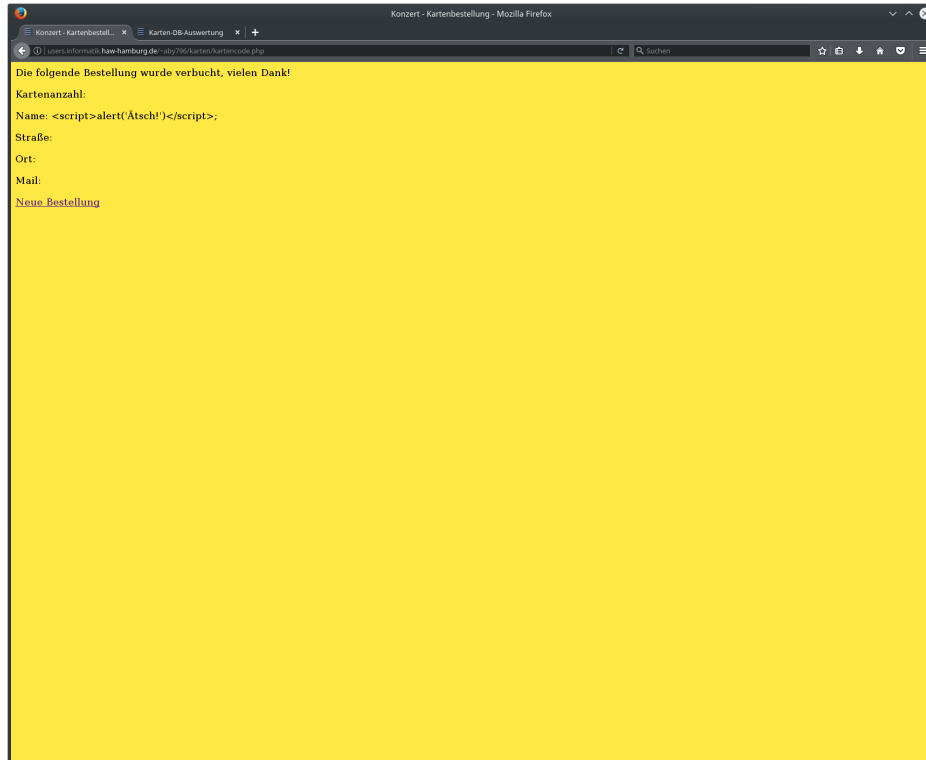
```
<script>alert('Ätsch!')</script>
```



3.2 XSS - Fix

XSS-Fix durch Umwandeln von besonderen Zeichen der Eingabe:

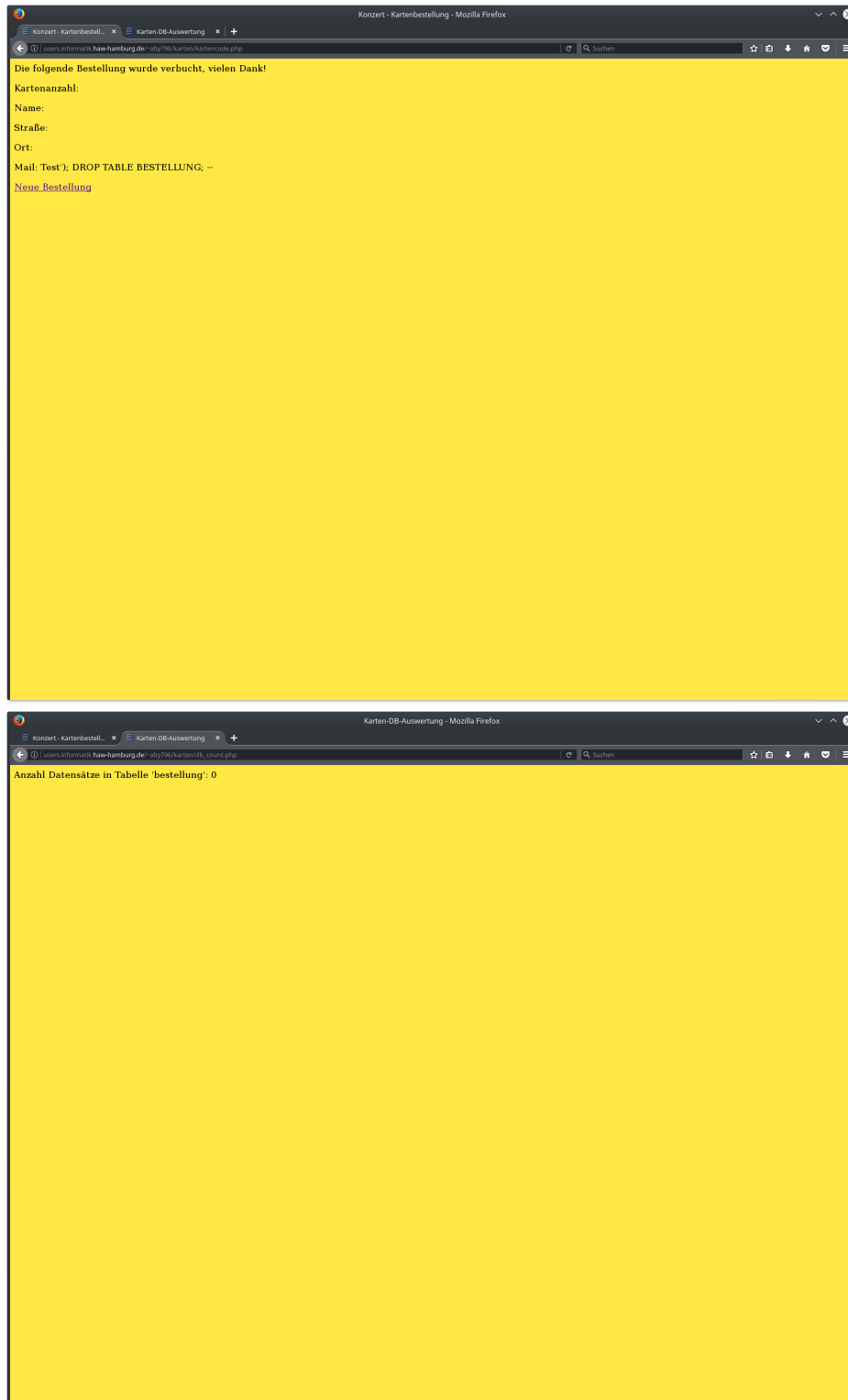
```
$Kartenzahl = htmlspecialchars($_POST['Kartenzahl'],  
    ↪ ENT_QUOTES, 'UTF-8');  
$Name = htmlspecialchars($_POST['Name'], ENT_QUOTES, 'UTF-  
    ↪ 8');  
$Strasse = htmlspecialchars($_POST['Strasse'], ENT_QUOTES  
    ↪ , 'UTF-8');  
$Ort = htmlspecialchars($_POST['Ort'], ENT_QUOTES, 'UTF-8  
    ↪ ');  
$Mail = htmlspecialchars($_POST['Mail'], ENT_QUOTES, 'UTF  
    ↪ 8');
```



3.3 SQL-Injection - Ausführung

In der Input-Box der Mail-Adresse folgendes Code-Snippet eingeben:

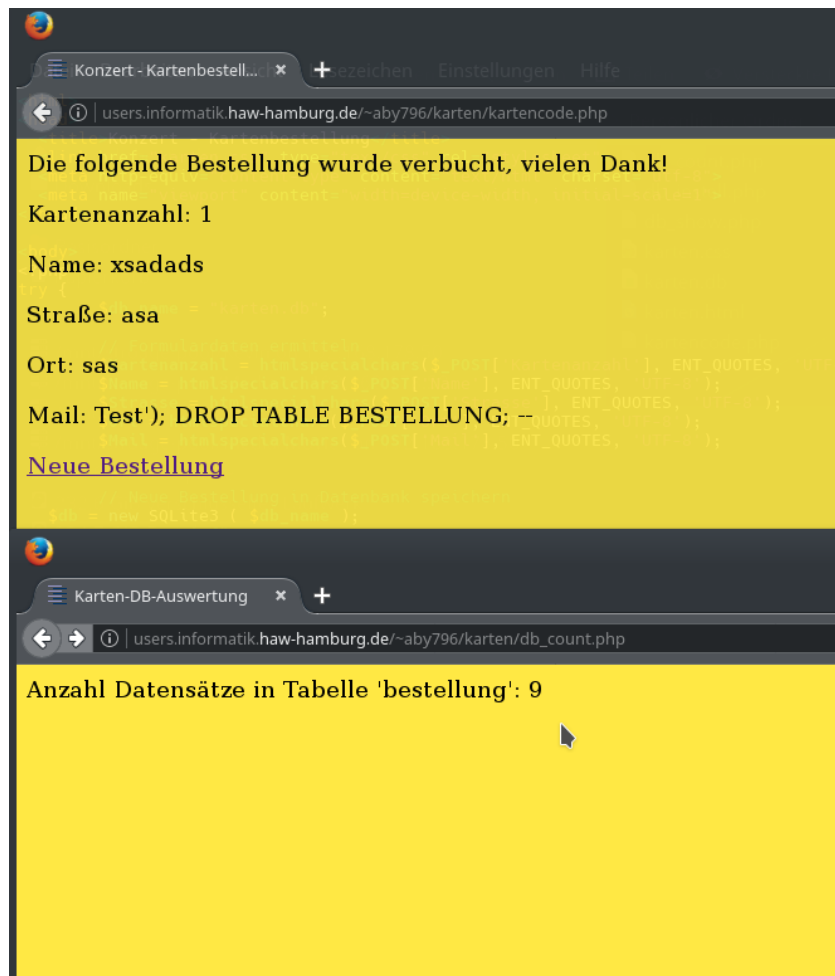
```
Test'); DROP TABLE bestellung; --
```



3.4 SQL-Injection - Fix

SQL-Injection-Fix durch Anwenden eines Prepared-Statements:

```
$sql = $db->prepare("INSERT INTO _bestellung_ (anzahl, _name  
→ , _strasse, _ort, _mail) VALUES (?,?,?,?,?)");  
$sql->bindParam(1, $Kartenzahl);  
$sql->bindParam(2, $Name);  
$sql->bindParam(3, $Strasse);  
$sql->bindParam(4, $Ort);  
$sql->bindParam(5, $Mail);  
  
$sql->execute();
```



4 Passwort - Cracking

4.1 Drei unterschiedliche Passwörter - Länge 3

Passwort1: 1a2 → Zeit: 7 Sekunden

Passwort2: ggj → Zeit: 20 Sekunden

Passwort3: 123 → Zeit: <0.1 Sekunden (Sofort)

Grund für die Zeitunterschiede:

Manche Passwörter sind in einer Passwortliste gespeichert, welche iterativ von oben nach unten durchprobiert wird. Falls kein Passwort passt wird das Passwort gebruteforced.

4.2 Passwort mit 8 Ziffern

Passwort mit 8 Ziffern: 96526738

```
./makepasswd 96526738 > digits.txt  
john -i=digits digits.txt
```

Zeit für das Cracken: 7 Sekunden

4.3 Passwort < 10 Sekunden - Länge 6

Passwort: abcdef → Zeit: <0.1 Sekunden (Sofort)

4.4 Komplexes Passwort - Länge 8

Passwort: XUXa71Aj → Zeit: <0.1 Sekunden (Sofort)