

## HELPDESK

# Webapps vor SQL-Injection geschützt

Jede Woche beantworten Sicherheitsexperten Leserfragen und geben Ratschläge, wie sich die Sicherheit in einem Unternehmen erhöhen lässt.

**Frage:** Ich habe den Auftrag erhalten, eine Webapplikation vor «SQL-Injection» zu schützen. Wie funktioniert dieser Angriff und wie kann ich meine Applikation davor bewahren?

SQL-Injection ist ein Angriff auf Datenbank-basierte Applikationen. Ähnlich wie beim «Buffer Overflow» wird eine Schwachstelle im Code ausgenutzt, welche es erlaubt, Benutzereingaben in Befehle zu verwandeln. Bei der SQL-Injection werden dabei die SQL Befehle manipuliert, welche die Applikation an die Datenbank sendet.

## Beispiel

Oft sind Webapplikationen von SQL-Injection betroffen, da diese öffentlich zugänglich sind. Eine geschlossene Webapplikation verlangt einen Benutzernamen und ein Passwort, um den Benutzer zu authentifizieren. Dieser gibt in den beiden Eingabefeldern seine Daten ein, etwa 'jdoe' und 'topsecret'. Daraus erstellt die Anwendung eine SQL-Abfrage, um festzustellen, ob die Eingaben korrekt sind. Diese wird an die Datenbank geschickt und sieht in der Regel etwa folgendermassen aus: «SELECT id FROM users WHE-

RE username = 'jdoe' and password = 'topsecret'». Gibt die Datenbank nun einen gültigen Eintrag zurück, weiss die Applikation, dass der Benutzer existiert und das Passwort korrekt eingegeben wurde. Selbstverständlich sollten Passwörter in Datenbanken verschlüsselt abgelegt werden, worauf aus praktischen Gründen in diesem Beispiel verzichtet wird.

Was passiert nun, wenn der Benutzer statt 'jdoe' und 'topsecret' folgendes eingibt:

«' OR '='»? Falls der Programmierer die SQL-Befehle durch Strings zusammensetzt und die Applikation anfällig ist,

**«Im schlimmsten Fall ist es gar möglich, die gesamte Datenbank zu löschen.»**

wird der finale SQL-Befehl wie folgt aussehen: «SELECT id FROM users WHERE username = ' OR '=' AND password = ' OR '='».

Damit wird eine Anfrage an die Datenbank geschickt, welche sämtliche Benutzer-IDs retourniert. Mit grosser Wahrscheinlichkeit wird die erste ID sogar die eines privilegierten Benutzers sein. Da die Applikation wahrscheinlich nur eine oder keine ID erwartet, bleiben

die restlichen Einträge unberührt und der Benutzer wird authentifiziert.

## Massnahmen

Dem SQL-kundigen lässt dieses klassische Beispiel unbeschränkte Möglichkeiten erahnen. Im schlimmsten Fall ist es gar möglich, Befehle einzuschleusen, welche die gesamte Datenbank löschen. Es ist leider nicht immer einfach, herauszufinden, ob eine Applikation tatsächlich verwundbar ist. Zeigt sie Datenbank-Fehlermeldungen an, erleichtert dies das Testen – aber auch den Angriff. Aus diesem Grund macht es Sinn, derartige Fehlermeldungen ausschliesslich für den internen Gebrauch frei zu schalten. Um sich vor SQL-Injection zu schützen, sollte der Entwickler in einem ersten

Schritt auf einen Whitelist-Filter für gültige Zeichen zurück greifen. Es kann aber sein, dass bei einigen Feldern einfache Anführungszeichen zulässig sind, wenn man Menschen mit Namen wie O'Brian nicht diskriminieren möchte. Die SQL-Injection ist nicht auf Login-Felder beschränkt: sie kann theoretisch bei jedem Eingabefeld, das mit der Datenbank in Verbindung steht, statt finden. Auch die ver-

schiedenen Codierungsvarianten der Zeichen (z.B. Unicode) wollen bei der Filterung bedacht sein.

Der wesentliche Punkt liegt darin, wie der Entwickler die SQL-Befehle zusammen setzt. Das zusammenkopieren von Strings sollte er vermeiden. Es existieren Standard-Bibliotheken, so genannte SQL Constructors. Hier wird zuerst das komplette SQL Statement mit Platzhaltern erstellt und anschliessend in einem zweiten Schritt durch die bereinigten Benutzereingaben abgefüllt. Dieses Vorgehen bietet wesentlich mehr Sicherheit. Schadensbegrenzung findet statt, wenn sich die Applikation mit einem eigenen Benutzerkonto und möglichst wenig Rechten mit der Datenbank verbindet. ■



**Der Autor**  
Simon Wepfer ist Consultant bei der Sicherheitsberaterin Oneconsult, Thalwil. [www.oneconsult.com](http://www.oneconsult.com)

**Unsere Experten beantworten Ihre Fragen.** Schreiben Sie uns: [it-security@computerworld.ch](mailto:it-security@computerworld.ch)

**Ein Archiv der Helpdeskartikel finden Sie im Internet:** [www.computerworld.ch](http://www.computerworld.ch)

ILLUSTRATION: THU

