Datenbanken-Praktikum Sommersemester 2006
Fallstudie 2:
Sicherheit

Um den Service zur Handhabung von Prüfungen und Notenvergaben zu implementieren, wurde zunächst einmal ein Datenbankserver auf dem Rechner SNAREDRUM aufgesetzt. Das Datenbankschema wurde angelegt. Datenbankseitig wurden ein Benutzer angelegt, mit dem der Server auf CONGA sich auf der Datenbank auf SNAREDRUM einloggt. Diesem Benutzer wurden uneingeschränkte Rechte auf allen Tabellen eingeräumt. Passwörter werden in der Tabelle *Person* MD5-verschlüsselt in der Spalte *passwort* gehalten. Ferner wurde dafür gesorgt, dass die Kommunikation von Client und Server nur über SSL, also verschlüsselt, stattfindet.

Nur ein DB-User mit uneingeschränkten Rechten! Besser: Mehrere DB-User für verschiedene Rollen.

Die Authentifizierung läuft folgendermassen ab:

Abhören durch SSL-Verschlüsselung schwierig, aber Man-in-the-Middle Attack theoretisch möglich, da keine Zertifizierung stattfindet

(a) Der Benutzer gibt auf dem Client TABLA seine Benutzer-ID und das Passwort ein

(b) Benutzer-ID und Passwort werden über den verschlüsselten Kanal an den Server geschickt

(c) Der Server CONGA loggt sich auf der Datenbank ein und setzt folgendes Statement ab:

Verbindung zwischen CONGA und SNAREDRUM unverschlüsselt

```
SELECT * FROM Person WHERE id='benutzer-id'
                       AND passwort='passwort'
```

'benutzer-id' und 'passwort' werden dabei durch die übermittelte ID und das Passwort ersetzt. Der so zusammengesetzte String wird an die Datenbank geschickt und von dieser ausgeführt. Liefert die Datenbank kein Ergebnis zurück, gilt die Authentifizierung als gescheitert und es wird eine Fehlermeldung an TABLA geliefert; andernfalls wird nun geprüft, ob es sich um einen Studenten oder Dozenten handelt:

SQL-Exploit mit „--" theoretisch möglich: dieser liefert immer ein Ergebnis. Hashwert des vom Client übertragenden Passworts muss dabei unterwegs geändert werden, um das Kommentarzeichen einzufügen; dies ist z.B. durch eine Man-in-the-Middle Attack möglich. Lösung: Prüfen auf „--", „;", etc.

```
SELECT id FROM Student WHERE id='benutzer-id'
```

In Verbindung mit obigem SQL-Exploit wird hier für nicht-existente Benutzernamen „Dozent" zurückgeliefert

Liefert diese Abfrage ein Ergebnis zurück, wird dem Client mitgeteilt, dass es ein Student ist; ansonsten wird dem Client gesagt, dass es sich um einen Dozenten handelt.

Nachdem der Benutzer erfolgreich authentifiziert wurde, holt sich der Client auf TABLA nun alle relevanten Prüfungen und Kurse vom Server.

- Ist der Benutzer Student, kann er alle Kurse und Prüfungen mit seiner Matr.-Nr. einsehen. Der Client ruft also auf dem Server eine Funktion auf, die die Benutzer-ID als Parameter hat; der Server verbindet sich zur Datenbank und setzt das Statement

  ```
  SELECT * FROM Prüfung WHERE matr_nr='benutzer-id'
  ```

  ab; das Resultat wird zum Client zurückgeliefert und dort dem Benutzer präsentiert. Ferner holt sich der Client noch alle Kurse mittels eines entsprechenden parameterlosen Funktionsaufrufs; der Server verbindet sich wieder zur Datenbank, führt ein `SELECT * FROM Kurs` aus und liefert das Ergebnis zurück. Der Client zeigt die Daten entsprechend an.

Siehe oben. SQL-Exploit mit„;" möglich; „--" hier harmlos, da nichts zurückgeliefert wird

- Ist der Benutzer Dozent, kann er alle seine Kurse sehen und die entsprechenden Prüfungen. Der Client auf TABLA ruft somit auf dem Server CONGA eine Funktion auf, die alle Kurse zu der übergebenen Dozent-ID holt. Die Menge der relevanten Kurse erstellt der Server durch das Statement

  ```
  SELECT * FROM kurs where Dozent_id='benutzer-id';
  ```

  Ferner holt der Server sich alle Prüfungen (`SELECT * FROM Prüfungen`) und lässt diejenigen Einträge fallen, die nicht vom Dozenten gehalten wurden. All diese Daten werden vom Server zum Client zurückgeschickt.

SQL-Exploit mit„;" möglich

Sobald eine Änderung vorgenommen wurde, wird im Client ein entsprechendes SQL-Statement generiert (`INSERT`, `UPDATE`, `DELETE`), und an den Server geschickt. Dieser verbindet sich zur Datenbank und führt das entsprechende Statement aus.

Die Betreiber des Systems argumentieren, dass ihr System sicher sei, da vor allem die Kommunikation sicher über SSL erfolgt und Passwörter nur verschlüsselt als Hashwert vorliegen. Ferner muss ein Benutzer erst authentifiziert sein muss, um überhaupt Daten einzusehen. Änderungen sind nur auf Basis der jeweiligen Rolle (Student/Dozent) möglich.

- Was haltet ihr von dieser Aussage? Welche Maßnahmen wurden unternommen, um das System sicher zu machen? Wo liegen mögliche Gefahren?
  Hinweis: Ist das System sicher vor Exploits? Sollten diese möglich sein, was könnten sie anrichten?
- Welche Änderungen würdet ihr vornehmen, um das System sicherer zu machen?