Kurs: CY-B-3 Sichere Programmierung

Dozent: Michael Heigl

**Absichern eines terminalbasierten Online-Schachservers (Chess Online)**

Autoren:

Sandro Schamberger (22102471) [33%]

Florian Hagengruber (22101608) [33%]

Christian Joiko (22111097) [33%]

**Inhaltsverzeichnis**

**1. Einleitung**

Die Digitalisierung betrifft momentan fast jeden Bereich unseres Lebens. Sei es nun am Arbeitsplatz, im Auto oder im eigenen zuhause, die Digitalisierung ist mittlerweile omnipräsent. Doch so viele Vorteile der digitale Wandel auch bringt, so wird sie jedoch stets von einem schwerwiegenden Problem begleitet, dass leider noch nicht dieselbe Aufmerksamkeit wie die Digitalisierung erhält wie die Digitalisierung selbst. Bei dieser bösartigen Nebenwirkung handelt es sich um die Cyber Sicherheit oder besser gesagt der Mangel davon. Mit jeder digitalen Neuerung und jedem neuen Gerät bilden wir eine weitere Angriffsfläche, auf der wir von Cyberkriminellen attackiert werden können, weshalb diese Dienste bestmöglich geschützt werden müssen. Viele denken hierbei wahrscheinlich an kritische Infrastruktur wie Krankenhäuser oder Stromversorgung, wobei jedoch alle Aspekte hinreichend abgesichert werden müssen, auch wenn sie noch so unscheinbar sind.

Ein Beispiel hierfür kommt aus dem Bereich der Freizeitgestaltung, die Videospielindustrie. Doch warum ist Informationssicherheit hier so wichtig? Dank der Digitalisierung ist diese Freizeitaktivität mittlerweile weit verbreitet, sodass im Jahr 2021 bereits jeder Zweite in Deutschland ab 16. Jahren gelegentlich Videospiele spielt [1]. Allein 2021 wurden 6,17 Milliarden Euro für Videospielsoftware ausgegeben [1]. Somit entsteht ein sehr großes Risiko für Firmen, da Vorfälle in der IT-Sicherheit einen großen Reputationsschaden mit sich führen können, wodurch der Anteil der Firma an dieser Geldsumme drastisch sinken kann. Doch nicht nur die Anbieter und Produzenten selbst sind von Cyberangriffen betroffen, sondern auch die Konsumenten, welche ihre persönliche Daten, ihre Zahlungsinformationen und ihr investiertes Vermögen in die Hände der Firmen geben. Werden Nutzerdaten nicht ausreichend geschützt, so könnten diese gestohlen werden und zu Schäden beim Nutzer führen.

Doch wie schwer ist es eigentlich, ein Videospiel mit all seinen verknüpften Diensten und Nutzerdaten gegen Cyberkriminelle und Hacker zu schützen? Um diese Frage zu beantworten, wird in diesem Projekt ein Server, mit dem Nutzer gegeneinander Schach spielen können, genauer unter die Lupe genommen. Dabei werden zuerst mögliche Schwachstellen analysiert und aufgrund ihres Gefahrenpotentials für Nutzer und Firma bewertet. Anschließend werden die erkannten Schwachstellen repariert, um eine möglichst sichere Version zu erstellen. Durch den Vergleich der beiden Versionen und die Aufzählung der möglichen Angriffsvektoren wird versucht, diese Frage zu beantworten.

**2. Funktionsweise der schwachen Version**

**2.1 Login und Menü**

Bevor ein Benutzer per „Chess Online“ Schach spielen kann, muss er mit dem Server eine Verbindung aufbauen. Dafür wird das Protokoll „Telnet“ verwendet, womit sich ein Client unter Eingabe der IP-Adresse des Servers und der Portnummer 8080 verbinden kann. Nachdem die Verbindung erfolgreich hergestellt wurde, wird dem Benutzer das Menü angezeigt, wo er aufgefordert wird, sich anzumelden oder sich zu registrieren, falls er noch keinen Account besitzt (siehe Abb. 1).

*Abbildung 1: Benutzer wird im Menü zum Login aufgefordert*

Für den Fall, dass der Benutzer noch kein Konto besitzt, kann er den Menüpunkt „Registration“ auswählen, um sich eines anzulegen. Für die Registrierung wird zuerst eine gültige Mailadresse eines Studenten oder Mitarbeiters der Technischen Hochschule Deggendorf benötigt. Die eingegebene E-Mail wird zur Kontrolle auf ihren Aufbau hin überprüft, ob sie eine Mailadresse der THD sein kann. Ist dies der Fall, so wird der Benutzer nach einem Passwort gefragt. Schlussendlich wird dem Nutzer dann mitgeteilt, dass ihm per E-Mail ein Aktivierungscode, bestehend aus fünf Ziffern, gesendet wurde (siehe Abb. 2). Die Mailadresse, das Passwort und der Aktivierungscode werden außerdem zusammen mit einem automatisch generierten Nutzernamen in der Tabelle „Spieler“ der Datenbank abgespeichert (siehe Abb. 3).

*Abbildung 2: Registrierung des Accounts (+ Hackermail abweisung)*

*Abbildung 3: Design der Datenbank*

Sollte der Nutzer bereits ein Konto besitzen, so kann er das Menü „Login“ benutzen, um sich anzumelden. Dafür wird seine Mailadresse und sein Passwort benötigt. Ist dies seine erste Anmeldung, so wird er zudem noch nach seinem Aktivierungscode gefragt (siehe Abb. 4). Wurde dieser richtig eingegeben, so ist der Benutzer eingeloggt und kann nun die Funktionen des Servers benutzen. Zu diesen zählen das Abrufen der Statistik eines anderen Spielers mittels einer Datenbankabfrage mit dem Befehl **--stats**, dem Ausloggen durch den Menüpunkt „Logout“**,** das Laden oder Starten eines Spiels gegen eine KI, dem Beitreten eines Online-Matches und dem Verlassen der Applikation. Im Zuge dieser Arbeit wird hauptsächlich der Vorgang eines Online-Matches betrachtet, da dieser Aspekt die meisten Risiken birgt.

*Abbildung 4: Anmeldung des Nutzers mit Abfrage des Aktivierungscodes*

**2.2 Während eines Online-Matches**

Entschließt sich ein eingeloggter Nutzer, dass er online eine Partie gegen einen anderen Spieler austragen möchte, so betritt er durch Eingabe des dazugehörigen Menüzeichens eine Wartschlange. Betritt eine weiterer Spiele die Warteschlange, so werden die Daten der beiden Kontrahenten in einem Array gespeichert und ein Spiel wird gestartet. Nun können die Nutzer abwechselnd durch die Eingabe einer Zeichenkette, bestehend aus dem Anfangsfeld und dem Zielfeld, ihre Figuren bewegen (e.g. „G1E1“). Neben diesen Aktionen können die Spieler Befehle wie **--stats**, **--surrender** und **--remis** verwenden.

Eine Runde kann aus verschiedenen Gründen enden. Die verschiedenen Möglichkeiten hierfür sind das manuelle Beenden mittels den oben genannten Befehlen **--surrender** und **--remis** (nur wenn sich beide Spieler für ein Remis einigen), das Besiegen des gegnerischen Königs oder das Verlassen der Runde beziehungsweise das Abbrechen der Verbindung. Letzteres führt hierbei zu einer automatischen Niederlage für den Spieler, welcher die Runde verlassen hat.

Am Ende der Runde wird das Ergebnis außerdem in der Datenbank vermerkt. Dazu zählt das Vermerken des Endergebnisses in der Tabelle „Spiele“, sowie das Abändern der Statistiken der Spieler. Dabei wird den beiden Spielern je nach Situation der Wert des Attributs „siege“, „niederlagen“ oder „remis“ um eins erhöht und die ELO-Bewertung wird für beide Spieler neu kalkuliert (siehe Abb. 5).

*Abbildung 5:Berechnung der ELO-Änderung*

**3. Analyse der Schwachstellen**

**3.1 Registrierung und Login**

Bereits bei der Erstellung oder Anmeldung eines Kontos ermöglichen eine Vielzahl an Schwachstellen dem Angreifer unterschiedliche Angriffsmethoden. Da bei diesen Aktionen viel Kommunikation mit der Datenbank stattfindet, wirken sich die Schäden durch einen Angriff hauptsächlich dort aus.

Der erste Angriffsvektor ist eine typische SQL-Injection, welche sowohl beim Registrieren als auch beim Login möglich ist. Dies ist aufgrund der fehlenden Input-Validation und der Verwendung von „%s-Platzhaltern“ anstatt „?-Platzhaltern“ möglich. Lediglich die Eingabe der Mailadresse während der Registrierung ist durch Verifizierung als Hochschuladresse vor einer Injection geschützt. Somit ist bei beiden Prozessen die Passworteingabe und beim Anmelden zusätzlich die Eingabe der Mailadresse verwundbar (siehe Abb. 6 + 7).

*Abbildung 6: Löschen der Tabelle „Spieler“ während der Registrierung*

*Abbildung 7: Löschen der Tabelle „Spiele“ während der Anmeldung*

*b', 'a', 'd'); DROP TABLE Spieler;--*