SQL-Injection im Unterricht

Von
Henning Ainödhofer (henning.ainoedhofer@gsma.schulbistum.de)
und Achim Kassemeier

Lizenz: CC-BY-NC-SA

Überblick **4**

Fachwissenschaftlicher Hintergrund **5**
Funktionsweise Client-Server-Klassen 7
Die Klasse Server 8
Die Klasse Client 8
Die Klasse Connection 9
Funktionsweise Datenbank-Klassen 9
Die Klasse DatabaseConnector 9
Die Klasse QueryResult 10

Einbettung in die Richtlinien des Landes NRW **10**
Prozessbezogene Kompetenzen 10
Inhaltsbezogene Kompetenzen 10
Medienkompetenzrahmen NRW 11

Methodische Ausrichtung der Reihe **11**

Verlaufsplanung der Reihe **12**
Fachliche Voraussetzungen 12
Reihenverlauf 12
Fokus Modellierung (M+K) 13
Fokus Implementierung (I+K) 14
Fokus Kommunikation (K) 15
Fokus Ganzheitlich (M+I+K) 15
Modellierung eines C-S-Zahlenratespiels 15
Lernziele 16
Problemorientierung 16
Stundenverlauf 17
Sicherung 17
Hilfen 17
Implementierung eines Client-Server-Ratespiels 18
Lernziele 18
Problemorientierung 18
Stundenverlauf 18
Sicherung 19
Testen des Programms mit verschiedenen Szenarien 19
Lernziele 20
Problemorientierung 20
Stundenverlauf 20
Sicherung 21
SQL-Injection 21
Lernziele 21

Problemorientierung 22
Stundenverlauf 22
Sicherung 23
Schließen der Sicherheitslücke 23
Lernziele 23
Problemorientierung 23
Stundenverlauf 23
Sicherung 24
Diskussion der Rechtlichen Rahmenbedingungen 24
Lernziele 24
Problemorientierung 25
Stundenverlauf 25
Sicherung 26

Lernstandsüberprüfung 26

Erfahrungen bei der Umsetzung 26

Anhang 27
AB Gesetz §202a 28
AB Datendiebstahl 29
AB Datenwert 30
AB Datenschutz 31
AB Szenarienkarten 32
TB Protokolllösung 33
TB Modellierungslösung 34
TB Softwareentwicklungsprozess 35
TB Nebenläufigkeit 36
TB Ergebnisse Testphase 37
TB Auswirkungen SQL-Injection 38
TB Sicherheitslücke schließen 39
TB Rechtliche Rahmenbedingungen 40
RS C-S-Kommunikation Ressourcenzugriff 41
AB Szenarienplanung 42
LÜ SQL-Injection 43
SQL-Injection – Die Technik 44
SQL-Injection – Rechtliches 45
SQL-Injection – Gegenmaßnahmen 46

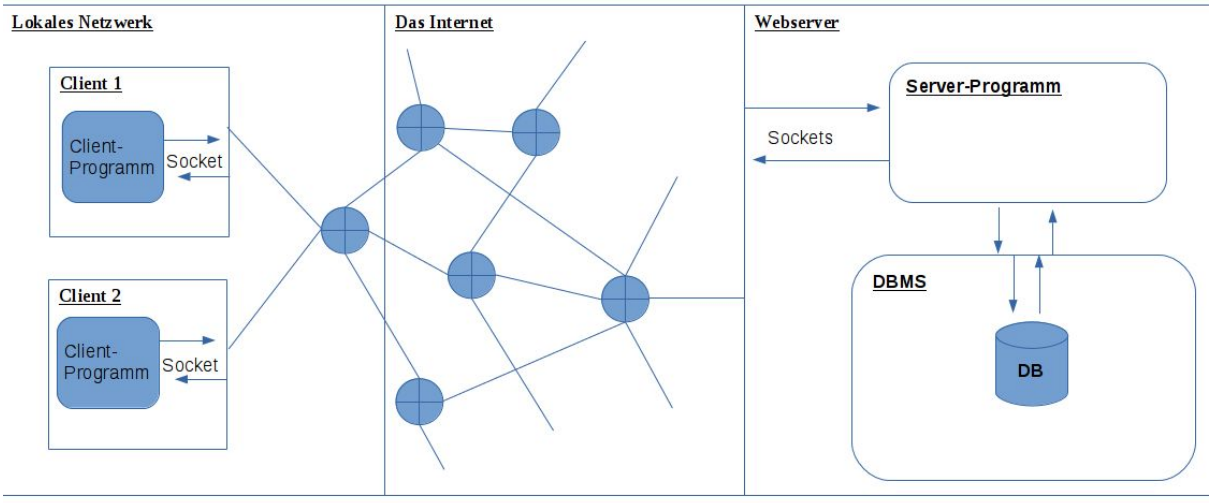
Überblick

Die vorliegende Reihe verknüpft viele verschiedene fachwissenschaftliche Themen der gymnasialen Oberstufe und bezieht dabei auch den Bereich “Informatik und Gesellschaft” mit ein. Als Kontext dient die Angriffstechnik der SQL-Injection auf Datenbanken. Auf der Grundlage der Themenkomplexe Datenbanken und Netzwerke, wird zunächst ein Online-Datenbank mit Hilfe des Client-Server-Modells umgesetzt, oder aber auch vorgegeben. Auf diese Online-Datenbank wird mittels SQL-Injection zugegriffen um Daten illegal zu verändern. Anhand dieses Beispiels wird ein Problemraum eröffnet, der sowohl bei den Datenbanken als auch beim Client-Server-Modell den Fokus auf Sicherheitsaspekte legt. Die Schülerinnen und Schüler werden mit der “grundlegenden Unsicherheit des Netzes” konfrontiert und müssen Strategien entwickeln und umsetzen, wie sie mit den bekannten Mitteln, solche SQL-Injections im gegebenen Fall verhindern oder zumindest sehr unwahrscheinlich werden lassen. Darüber hinaus werden auch gesellschaftliche, bzw. gesellschaftspolitische Themen angesprochen, die in Zusammenhang mit der Angriffsmethode und der Sicherheit von Online-Datenbanken stehen.

Die gesamte Reihe basiert dabei auf den Klassen der “Standardsicherung NRW” und bedient Schülerwünsche hinsichtlich des Themenbereichs “Hacking”. Die Reihe, wie hier beschrieben, ist für den Leistungskurs konzipiert, kann aber auch mit einigen Hilfen seitens der Lehrperson für den Grundkurs verwendet werden.

Fachwissenschaftlicher Hintergrund

Datenbanken an sich gibt es schon eine ganze Weile. Allerdings hat sich in der Konzeption im Laufe der Zeit einiges getan. Die ersten elektronischen Datenbanken wurden in den 1960ger Jahren entwickelt. Dabei handelte es sich um sogenannte hierarchische Datenmodelle. IMS von der Firma IBM war eines der ersten Systeme die eine Datenverwaltung mit Hilfe der Sprache DL/I ermöglichte. Dieses System war lokal begrenzt und wurde häufig in großen Firmen oder Banken genutzt. Parallel dazu wurde mit CODASYL auch ein System entwickelt, dass den Zugriff auf eine Datenbank über ein größeres Netzwerk ermöglichte. Hier waren schon viele Dinge enthalten, die später in Relationalen Datenbanken auch zu finden sind. Ab den 1990ger Jahren setzte sich mit der Entwicklung des Internets, das Relationale Datenbankmodell mehr und mehr durch.
Die Struktur des Internets fördert dabei das Anbieten einer Online-Datenbank, da die grundsätzliche Kommunikation über standardisierte Protokolle das Entwickeln von entsprechenden Schnittstellen vereinfacht.
So lassen sich weitere Anwendungen in das Client-Server-Modell integrieren. So kann ein Webserver beispielsweise ein DBMS betreiben. Mittlerweile beruhen große Teile des Internets und dessen Diensten auf der im folgenden vorgestellten Architektur. Das klassische Beispiel sind dabei dynamisch generierte Webseiten, die mit Hilfe von php-Scripts und auf der Grundlage einer Datenbank erzeugt und auf Anfrage zu einem Browser gesendet werden.



Der Socket der im Client-Programm geöffnet wird ist das Bindeglied zwischen der Anwendungsschicht und den darunter liegenden Schichten Internet und Transport. Das

Clientprogramm erzeugt über den Aufruf der Methode *connect()* mithilfe einer Portangabe und der Angabe der IP-Adresse des Servers ein neues Socket-Objekt. Über diesem Socket-Objekt werden nun zwei I/O-Objekte "gelegt". Ein Reader-Objekt für das Zwischenspeichern der eintreffenden Nachrichten vom Server und ein Writer-Objekt zum Senden von Nachrichten an den Server. Die Methode *recieve()* wird vom Client-Thread ständig, d.h. nebenläufig, abgefragt. Die Methode *send(nachricht)* schickt eine übergebene Nachricht an den Server.
Serverseitig gibt es eine ähnliche Implementation. Auch hier werden Socket-Objekte erzeugt und zwar für jeden angemeldeten Client. Stellt ein Client eine Anfrage über das Internet, so registriert der Server die Eingabe über ein zuvor festgelegtes Protokoll. Insofern eine Datenbank angebunden ist, wird der Auftrag entsprechend ausgeführt. Ändert sich dabei etwas an der Datenbank wird aufgrund des Protokolls entschieden welche angeschlossenen Clients darüber informiert werden müssen.

Der Einsatz eines Solchen Modells bringt viele Vorteile mit sich. Die Datenbank ist vom eigentlichen Programm abgekapselt und kann leicht gewartet oder ausgetauscht werden. Ebenso kann das Server-Programm und sein Protokoll ständig erweitert werden, ohne das die Funktionsfähigkeit der Clients eingeschränkt wäre. Updates lassen sich über entsprechende Protokollanweisungen übermitteln.
Darüber hinaus gibt es aber auch viele Nachteile. Da sich der Dienstanbieter nicht sicher sein kann, dass alle Clients mit "guten Absichten" benutzt werden, muss eine Menge Aufwand für die Sicherheit des Systems betrieben werden. Vor allem die Daten in der Datenbank müssen vor nicht gewünschten Zugriff geschützt werden. Sollten hier Lücken entstehen, können die Clients dazu genutzt werden, die Daten auszuspähen. Der Wert solcher Daten ist unterschiedlich, doch für etwa 1000 Facebook-Anmeldedaten bekommt man schätzungsweise 10€. Bedenkt man, dass man mit Datenbankangriffen weit mehr als 1000 Datensätze ausspähen kann, kommen dort größere Summen zusammen.[1]
Eine Angriffstechnik, die bei Datenbanken eingesetzt wird, ist die SQL-Injection. Diese Methode nutzt aus, dass Programme die mit Datenbanken zusammenarbeiten, auch Anfragen an diese stellen, die mit Benutzereingaben gefüllt sind. Beispielsweise bei der Anmeldung eines Nutzers, muss überprüft werden, ob dieser im System ist und Zugang zu eben diesem bekommt. Dazu setzt das Client-Programm eine Anfrage an den Server ab, in der dann der eingegebene Benutzername, nebst Passwort übertragen wird. Ein Angreifer kann diese Standardanfrage dazu nutzen, selbst ein SQL-Statement zu übermitteln. Dazu

[1] http://www.trendmicro.de/infografiken/wie-viel-sind-ihre-daten-wert/index.html

gibt er keinen Benutzernamen an, sondern ein SQL-Statement. Darüber hinaus maskiert er dieses so, dass das eigentliche Statement entweder mit ausgeführt oder gar nicht abgefragt wird. Der Server stellt dann die Angriffs-Anfrage des Benutzers an die Datenbank und gibt das Ergebnis zurück. Mit etwas Kenntnis zu den beteiligten Systemen kann man so, auch ohne Systemzugang, an jede Menge Daten kommen.

Vor diesem Hintergrund gibt es in nahezu jeder Programmiersprache, die die Server-Client-Architektur in Zusammenhang mit einer Datenbank ermöglicht, vorgefertigte Strukturen und/oder Herangehensweisen, die nicht gewünschte Zugriffe unmöglich machen oder zumindest höchst unwahrscheinlich werden lassen.

Das Bundesamt für Sicherheit in der Informationstechnik hat im Allgemeinen Dinge zusammengestellt, die gegen die genannte Angriffstechnik der SQL-Injection helfen.[2]

Außerdem muss man sich darüber im Klaren sein, dass der “Nichtschutz” von Daten nicht strafbar ist, das Ausspähen von Daten schon. Laut Paragraph §202 a, der eine Erweiterung des Briefgeheimnisses darstellt, macht sich strafbar, wer “unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.”.

Trotzdem ist die allgemeine Kenntnis solcher Angriffstechniken, aber auch der Gegenmaßnahmen wichtig, um zu verstehen, in welchem gesellschaftlichen Kontext Informationstechnologie und hier im Besonderen Dienste die über das Internet angeboten werden, stehen. Die Meldungen der großen Medienanstalten der letzten Jahre zeigen nämlich, dass auch große Unternehmen, mit Millionen von Daten in ihren Datenbanken, nicht immer ausreichend auf den Schutz eben dieser Daten geachtet haben.

Als Gesellschaft sollte man sich also fragen, inwieweit beispielsweise auch nicht oder nicht gut gesicherte Datenbanken, strafrechtlich relevant sein könnten. Außerdem sollte sich die Gesellschaft fragen, ob wie viele und vor allem welche Daten man bei wie vielen verschiedenen Diensten abgeben sollte, obwohl diese nur versprechen die Daten entsprechend zu sichern. Darüber hinaus lassen sich bestimmt noch weitere Fragestellungen diskutieren.

Funktionsweise Client-Server-Klassen

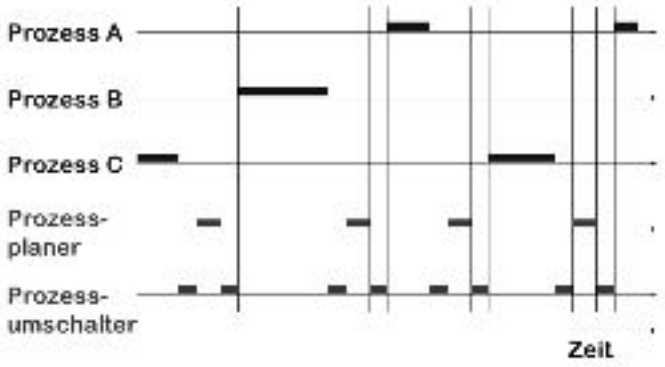
Die vorhandene Netzarchitektur bedingt eine Unterteilung der Teilnehmer in Anbieter (engl. Server) und Teilnehmer (engl. Client). Der Server dient dabei als Verteiler und Vermittler von Diensten an dem sich die Clients anmelden können, um die Dienste in Anspruch zu

[2] https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/ _content/m/m02/m02363.html

nehmen. Über eine Verbindung (engl. Connection) können Server und Client Daten untereinander austauschen. Dies geschieht über festgelegte Protokolle, die gemeinsam das OSI-Schichtenmodell kennzeichnen. Im folgenden werden die drei Komponenten näher betrachtet.

Die Klasse Server

Über die Klasse Server ist es möglich, Serverdienste anzubieten, so dass Clients Verbindungen gemäß dem TCP/IP-Protokoll aufbauen können (siehe Connection). Nachrichten werden grundsätzlich zeilenweise verarbeitet, d.h., beim Senden einer Zeichenkette wird ein Zeilentrenner ergänzt und beim Empfangen wird er entfernt. Der Verbindungsaufbau, Nachrichtenempfang und Verbindungsende geschehen nebenläufig, was das Anmelden beliebig vieler Clients ermöglicht. Durch das Überschreiben der entsprechenden Methoden in einer ableitenden Serverklasse, kann diese auf bestimmte Ereignisse reagieren.



Der Begriff der Nebenläufigkeit ist im Zusammenhang mit Serverdiensten wichtig, da die Dienste in Echtzeit von verschiedenen Clients angefragt werden können. Die Nebenläufigkeit ist allerdings häufig nur Softwareseitig und über sogenannte Threads gelöst. Threads sind dabei Objekte, denen von einem Monitorobjekt, Rechenzeit zugewiesen wird.
Auf Mehrkernprozessoren kann es dabei auch zu echter Nebenläufigkeit kommen, da bspw. Java diese schon länger unterstützt. Bei der Klasser Server läuft ab Erzeugung eines Serverobjektes ein ServerSchleifen-Objekt, das von der Klasse Thread abgeleitet ist. Hier werden alle Verbindungen zu den Clients angenommen und als neue Verbindung gespeichert. Ist eine solche Verbindung aufgebaut wartet der Server über diesem Objekt auf Nachrichten vom Client um diese zu verarbeiten.

Die Klasse Client

Über die Klasse Client werden Netzwerkverbindungen mit dem TCP/IP-Protokoll ermöglicht (siehe Connection). Es können nach Verbindungsaufbau zu einem Server Zeichenketten gesendet und empfangen werden, wobei der Empfang nebenläufig geschieht. Zur Vereinfachung geschieht dies zeilenweise, d.h., beim Senden einer Zeichenkette wird ein Zeilentrenner ergänzt und beim Empfangen wird er entfernt. Die empfangene Nachricht wird

durch die Ereignisbehandlungsmethode processMessage(String message) verarbeitet, die in Unterklassen überschrieben werden muss. Das auch der Client in einem eigenen Threadobjekt läuft, ist der Tatsache geschuldet, dass der Client ständig mit einer Nachricht des Servers rechnen muss. Um nicht dauerhaft den Prozessor mit dem Erwarten einer Nachricht vom Server zu blockieren, wird dem Threadobjekt von Zeit zu Zeit gestattet, Rechenzeit zu belegen.

Die Klasse Connection

Objekte der Klasse Connection ermöglichen eine Netzwerkverbindung über die Verwendung des TCP/IP-Protokolls. Wichtigster Bestandteil der Klasse Connection und Schnittstelle zwischen Client und Server bildet der Socket (engl. für Steckverbindung oder Steckdose). Der Socket ist das Bindeglied zwischen den Anwendungsschichten (OSI 5-7) und den darunter liegenden OSI-Schichten 3 (Internet) und 4 (Transport). Die Methode connect() nimmt dabei den eingegebenen Port und die IP-Adresse und erzeugt ein neues Socket-Objekt. Über diesem Socket-Objekt werden nun zwei I/O-Objekte gelegt. Ein Reader-Objekt für das Zwischenspeichern der eintreffenden Nachrichten vom Server und ein Writer-Objekt zum Senden von Nachrichten an den Server. Die Methode recieve() wird vom Client-Thread „ständig“ abgefragt. Die Methode send(String nachricht) schickt eine übergebene Nachricht an den Server.

Funktionsweise Datenbank-Klassen

Die Datenbankklassen umfassen eine Klasse für den Verbindungsaufbau und die Kommunikation zum DBMS. Eine weitere für die Kapselung der Anfrageergebnisse und die Datenstruktur Queue, die intern zur Verwaltung der Anfrageergebnisse verwendet wird. Im Folgenden wird die Funktionsweise der beiden speziellen Datenbankklassen näher erläutert.

Die Klasse DatabaseConnector

Die Klasse DatabaseConnector benötigt für die Erzeugung einer eigenen Instanz die Parameter IP, Port, Datenbankname, Benutzername und Passwort. Da in diesem Falle mit SQLite bearbeitet wird, werden laut Dokumentation die Parameter IP, Port sowie Benutzername und Passwort ignoriert, da die Datenbank immer lokal existiert und eine Benutzerverwaltung in SQLite nicht implementiert ist. Somit bleibt nur die Angabe des Datenbanknamens. Der Treiber für die SQLite Datenbank wird zur Herstellung der Datenbankverbindung geladen (org.sqlite.JDBC) und so die Verbindung aufgebaut. Wichtig

zu beachten ist, dass wenn keine Datenbank mit dem angegebenen Namen existiert, die Klasse eine neue Datenbank erzeugt.
Von nun an können SQL-Statements über die Methode executeStatement(statement) abgesetzt werden. Insofern eine Rückgabe erwartet wird, kann diese mit Hilfe der Methode getCurrentQueryResult() angefordert werden. Das zurückgegebene Objekt ist dann vom Typ QueryResult (siehe unten). Falls Fehler bei der Anfrage oder der Rückgabe aufgetreten sind können diese über getErrorMessage() abgefragt werden. Der Aufruf der Methode close() schließt die Verbindung. Auch wenn das Objekt dann noch in einer anderen Klasse existiert, besteht keine Verbindung mehr. Ein Methode “verbinde()” wird nicht angeboten.

Die Klasse QueryResult
Ein Objekt der Klasse QueryResult stellt die Ergebnistabelle einer Datenbankanfrage mit Hilfe der Klasse DatabaseConnector dar. Die Repräsentation der Daten ist ein zweidimensionales Array, wobei in dem Array ausschließlich die Daten enthalten sind. Die Attributnamen und Datentypen sind in separaten Arrays gespeichert. Über den Aufruf der Methode getData() wird das Array zurückgegeben. Außerdem lassen sich die Attributnamen und Datentypen ebenfalls über Methoden zurückgeben. Darüber hinaus kann man sich mithilfe der Methoden getRowCount() und getColumnCount() die Längen der Einträge ausgeben lassen. Row steht dabei für die Anzahl der Datensätze, Column für die Anzahl der Ergebnisspalten.

Einbettung in die Richtlinien des Landes NRW

Prozessbezogene Kompetenzen

In dieser Unterrichtsreihe werden übergeordnete Kompetenzerwartungen des Bereiches “Argumentieren” (vgl. [KLP 2014, S. 24.]) erworben und erweitert, indem die Schülerinnen und Schüler

- Aussagen über Informatiksysteme erläutern,
- im Problemlösungsprozess Alternativen aufzeigen und ihre Auswahlentscheidung begründen,
- Informatiksysteme hinsichtlich ihrer Möglichkeiten, Grenzen und Auswirkungen beurteilen.

Inhaltsbezogene Kompetenzen

Die Unterrichtsreihe fokussiert sich vor allem auf Kompetenzen in den Inhaltsfeldern “Informatik, Mensch und Gesellschaft” (vgl. [KLP 2014, S. 35f.]) und “Informatiksysteme” (vgl. [KLP 2014, S. 34f.]) . Die Schülerinnen und Schüler erwerben und erweitern diese Kompetenzen, indem sie

- anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystem und des Datenschutzes untersuchen (A),
- Problemlagen untersuchen und bewerten, die sich aus dem Einsatz von Informatiksystemen ergeben hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenslagen (A),
- Protokolle zur Kommunikation in einem Client-Server-Netzwerk entwickeln und erweitern (M),
- die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung erläutern (A).

Medienkompetenzrahmen NRW

Der Medienkompetenzrahmen NRW weist digitale Kompetenzen aus, die die Schülerinnen und Schüler bis zur Jahrgangsstufe 10 erreicht haben sollen. In der Oberstufe werden diese dann im Unterricht weiter vertieft und angewendet. In Bezug auf die vorliegende Reihe, werden vor allem die Kompetenzen im Bereich Problemlösen und Modellieren (Bereich 6) verwendet. Auch im Bereich 3 und 4, Kommunizieren und Kooperieren, sowie Produzieren und Präsentieren, werden Kompetenzen zum Thema Rechtliche Grundlagen (3.3, 3.4, 4.4) geschult. Ansonsten werden im Unterricht natürlich immer wieder Medien eingesetzt, die die ausgewiesenen Kompetenzen des Medienkompetenzrahmens aufgreifen und weiterentwickeln.

Methodische Ausrichtung der Reihe

Methodisch ist der exemplarisch beschriebene Reihenverlauf darauf ausgerichtet, dass die Schülerinnen und Schüler sich immer in der Rolle des Softwareentwicklers wieder finden. Dies bedeutet, dass zwischen Phasen am Rechner in denen Implementiert wird auch

Phasen der Kommunikation eingebaut sind. Dies soll den hohen Stellenwert der Kommunikation in Informatikberufen hervorheben. Ebenso soll der Reihenverlauf von den Schülerinnen und Schülern mitbestimmt werden. So werden immer Wieder Situationen geschaffen, in denen sich die Schülerinnen und Schüler entscheiden können, welche Bereich des jeweiligen Themengebietes bearbeitet werden, bzw. wie es im Unterrichtsverlauf weiter gehen soll. Zu diesem Zweck werden Problemsituationen geschaffen, die die Schülerinnen und Schüler lösen müssen.

Verlaufsplanung der Reihe

Fachliche Voraussetzungen

Die Reihe setzt eine weite Bandbreite an bereits erlerntem Fachwissen und anwendbaren Techniken voraus. Die Schülerinnen und Schüler müssen sowohl in der Modellierung als auch in der Programmierung schon mehrere einfache Projekte objektorientiert umgesetzt haben. In Zuge dessen sollten auch die linearen Abstrakten Datenstrukturen ein Begriff sein und zur Problemlösung angewendet werden können. Die Themenkomplexe Datenbanken und Netzwerke sollten Themenspezifisch behandelt worden sein, so dass die Schülerinnen und Schüler SQL-Befehle absetzen und die Rückgaben verarbeiten können, auch mit Hilfe der Java-Schnittstelle. In Bezug auf das Thema Netzwerke sollten vor allem die grundlegenden Inhalte wie die Funktionsweise und der Aufbau von Netzwerken und die Modellierung und Implementierung von einfachen Client-Server-Strukturen mit ihren Protokollen behandelt worden sein.
Das Grundspiel “Zahlenraten” sollte bereits in vorangegangenen Sequenzen als Beispiel benutzt worden sein, damit die Einarbeitung nicht allzu lange dauert. Hier bieten sich mehrere Stellen in der Oberstufen an:

- Modellierung und Implementierung von Programmen (EF oder Wdh. Beginn Q1)
- Datenbankanbindung von Programmen (Datenbanken Q1, Highscore speichern)

In Bezug auf die Anwendung im Grundkurs, in der die Datenbankanbindung an Programme nicht zwingend erforderlich ist, sollte man mit vorgefertigten Programmen arbeiten, bei denen lediglich die SQL-Statements eingebunden werden müssen. Wichtig ist, dass die SuS auch im Grundkurs in die Entwicklerrolle kommen. Dies gelingt aber nur, wenn sie das Gefühl haben, das Spiel weitgehend selbst modelliert und implementiert zu haben.

Reihenverlauf

Je nach Kursvoraussetzungen und Vorlieben, sowie Zeitvorgaben und Kursstufe, sind die folgenden Verläufe denkbar. In Bezug auf die Zielsetzung der Reihen ist immer darauf zu achten, dass in jedem Fall die Kommunikation mit eingebaut wird, da das Thema auch rechtliche Komponenten beinhaltet, welche in jedem Fall angesprochen werden müssen!

Fokus Modellierung (M+K)

Beim Fokus auf der Modellierung und der Kommunikation stehen zwei Fragestellungen im Vordergrund. Zum einen muss man sich die Frage stellen, wie man Systeme, die mit einer Online-Datenbank arbeiten, gegenüber SQL-Injections sicher macht. Zum anderen sollte über rechtliche Grundlage von Systemen und Diensten, sowie über Auswirkungen auf die Gesellschaft und ihre Prozesse diskutiert werden.

Zu Beginn der Reihe wird also ein Szenario vorgestellt, welches zunächst einen Datenbankbasierte Webanwendung zeigt, die einen bestimmten Dienst anbietet. Im vorliegenden Fall handelt es sich um ein einfaches Ratespiel. Die Funktionalität sollte im Kursverband vorgestellt werden.

Es schließen sich mehrere Möglichkeiten an, die SQL-Injection einzuführen, bzw. vorzuführen.

Szenario 1:

Die SQL-Injection wird von der Lehrperson selbst vorgeführt und erläutert.

Szenario 2:

Ein Wettbewerb wird ausgerufen um einen Belohnung, für denjenigen, der die wenigsten Versuche beim Ratespiel erreicht. Ein zuvor Eingeweihter (Lehrer, Schüler) setzt eine SQL-Injection ab, die einen unmöglich erreichbaren Wert in der der Datenbank abspeichert (0 Versuche) und einem gleichzeitig den Preis sichert.

Szenario 3:

Die Methode der SQL-Injection wird von den Schülerinnen und Schülern selber erarbeitet. Die Schülerinnen und Schüler versuchen über ein einleitendes Arbeitsblatt[3] und mit ihren SQL-Kenntnissen, die Datenbank des Dienstes zu ihren Gunsten zu manipulieren.

Szenario 4:

[3] siehe Anhang AB SQL-Injection

Eine Kombination aus Szenario 2, 3 und 4, bei der ein SQL-Statement gezeigt wird und die Schülerinnen und Schüler weitere Möglichkeiten der SQL-Injection anhand eines Arbeitsblattes ausprobieren.

Jedes Szenario ist anwendbar und muss je nach äußeren Bedingungen (Zeitrahmen, Leistungsfähigkeit, Vorwissen) ausgewählt werden.
Die eigentliche Modellierung fängt erst im Anschluss an. Hier sollen nämlich mögliche Gegenmaßnahmen von den Schülerinnen und Schülern entwickelt und diskutiert werden. Über das zuvor ausgeteilte Arbeitsblatt wird dabei nicht nur die SQL-Injection erläutert, sondern auch, wann SQL-Befehle ausgeführt werden und wann nicht. Die Modellierung beschränkt sich hierbei auf SQL soll keine weitergehenden Überlegungen zur Implementierung in Java hervorbringen.
Aufgrund des Vorwissens und der Informationen auf dem Arbeitsblatt sollten die Schülerinnen und Schüler dazu in der Lage sein zumindest eine der folgenden Möglichkeiten zu entwickeln:

- Es muss verhindert werden, dass mit Variablen SQL-Schlüsselwörter übergeben werden können
- Die Aneinanderreihung von SQL-Befehlen muss verhindert werden
- Schlüsselwörter wie DROP sollten nicht zur Verfügung stehen

Im Anschluss an die Modellierung des Angriffs und die Abwehr, erarbeiten die Schülerinnen und Schüler den rechtlichen Rahmen der SQL-Injections. Hierbei ist es wichtig die entsprechenden Gesetzestexte[4] zu bearbeiten und die Auswirkungen zu diskutieren. Als letzten Schritt sollen sich die Schülerinnen und Schüler damit auseinandersetzen, dass es wohl ein Gesetz gibt, welches das Ausspähen von Daten unter Strafe stellt, das “nicht sichern” von Daten aber nicht. Letztlich soll dabei ein Gesetzesentwurf herauskommen, der das “nicht sichern” von Daten nach einem bestimmten Standard unter Strafe stellt.

Fokus Implementierung (I+K)

Bei diesem Fokus wird die Methode der SQL-Injection von der Lehrperson im Kontext des Ratespiels vorgestellt und es werden die Gegenmaßnahmen, die vom BSI und anderen Einrichtungen empfohlen werden erläutert. Dies bietet eine Grundlage für die Implementierung von Gegenmaßnahmen. Diese sind, in Methoden gekapselt, an passender Stelle in einer weiteren Version des Ratespiels angegeben. Die Schülerinnen und Schüler

[4] siehe Anhang AB Gesetz §202 a

sollen nun anhand der Informationen die Methoden mit den ihnen bekannten Mitteln implementieren. Im Zuge dessen werden vor allem String-Methoden wiederholt oder eben eingeführt. Die Methoden befinden sich alle in der speziellen Klasse Server:

- boolean checkVariableSQLKeyword(String var)
  Überprüft, ob sich in der Variablen SQL-Schlüsselwörter verbergen. Ist dies der Fall wird true zurückgegeben.
- boolean checkVariableSQLConcat(String var)
  Überprüft, ob sich in der Variablen eine auffällige Abfolge von Zeichen befindet. Beispielsweise sollte der doppelte Slash (//) oder das Semicolon (;) nicht in der Variable zu finden sein. Ist dies dennoch der Fall wird true zurückgegeben.

Im Anschluss an die Implementierung der Abwehr, erarbeiten die Schülerinnen und Schüler den rechtlichen Rahmen der SQL-Injections. Hierbei ist es wichtig die entsprechenden Gesetzestexte zu bearbeiten und die Auswirkungen zu diskutieren. Als letzten Schritt sollen sich die Schülerinnen und Schüler damit auseinandersetzen, dass es wohl ein Gesetz gibt, welches das Ausspähen von Daten unter Strafe stellt, das “nicht sichern” von Daten aber nicht. Letztlich soll dabei ein Gesetzesentwurf herauskommen, der das “nicht sichern” von Daten nach einem bestimmten Standard unter Strafe stellt.

Fokus Kommunikation (K)

Bei diesem Fokus wird die Methode der SQL-Injection von der Lehrperson im Kontext des Ratespiels vorgestellt und es werden die Gegenmaßnahmen, die vom BSI und anderen Einrichtungen empfohlen werden, erläutert. Die Umsetzung als Quellcode wird ebenfalls gezeigt und eine erneute Prüfung durch SQL-Injection durchgeführt.
Anschließend werden den Schülerinnen und Schülern mehrere Beispiele für Datendiebstahl[5] , Datenwert[6] und [7]Datenschutz vorgelegt. Diese drei Themen sollen arbeitsteilig erarbeitet und gegenseitig präsentiert werden. Dies bietet dann die Grundlage um sich mit dem Fakt auseinander zu setzen, dass es wohl ein Gesetz gibt, welches das Ausspähen von Daten unter Strafe stellt, das “nicht sichern” von Daten aber nicht. Letztlich soll dabei ein Gesetzesentwurf herauskommen, der das “nicht sichern” von Daten nach einem bestimmten Standard unter Strafe stellt.

[5] siehe Anhang AB Datendiebstahl
[6] siehe Anhang AB Datenwert
[7] siehe Anhang AB Datenschutz

Fokus Ganzheitlich (M+I+K)

Exemplarisch wird der Reihenverlauf *Fokus Ganzheitlich (M+I+K)* nun näher erläutert, da dieser für den Leistungskurs und die vorhandene Zeit am besten erscheint. Er stellt eine Kombination aus den zuvor vorgestellten Vorgehen dar und verwendet unter anderem die selben Arbeitsmaterialien. Die Stunden sind als Doppelstunden zu verstehen.

Modellierung eines C-S-Zahlenratespiels

Die erste Stunde in der Reihe dient dem Anknüpfen an das bereits bekannte Ratespiel aus der EF (Implementierung Grundlagen), bzw. aus der Q1 (Datenbanken und Datenbankanbindung in Java). Dazu wird die Implementierung des Spiels und die Spiellogik noch einmal ausprobiert. Anhand dieser Thematisierung werden dann die einzelnen Komponenten für die Modellierung identifiziert und ein entsprechendes UML-Diagramm entwickelt. Eine getrennte Modellierung für den Client und den Server sind nicht vorgesehen.

Lernziele

Die Schülerinnen und Schüler ...

- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)
- analysieren und erläutern informatische Modelle (A)
- beurteilen die Angemessenheit von Modellierungen und Implementationen (A)
- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M)
- modifizieren und erweitern informatische Modelle (M)
- wenden im Modellierungsprozess geeignete Lösungsstrategien an (M)
- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K)
- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen, Grafiken und Formalismen da (D)
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K)
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)
- modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M)

- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M)
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M)
- stellen die Kommunikation zwischen Objekten grafisch dar (D)
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D)
- dokumentieren Klassen (D)
- analysieren und erläutern objektorientierte Modellierungen (A)
- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M)

Problemorientierung

Auf der Grundlage der Wiederholung des Ratespiels und der vorangegangenen Thematisierung der Client-Server-Architektur liegt die Problematisierung in der Umsetzung des bekannten Spiels als Online-Multiuser-Spiels.

Stundenverlauf

Einstieg:
Wiederholung des Ratespiels (mit Datenbankanbindung) und der Modellierung
Aufgabenbeschreibung/Problem I:
Welche Schritte sind notwendig um aus dem Spiel ein Online-Spiel zu machen?
Erarbeitung I (Plenum, Einzelarbeit => Partnerarbeit):
Im Plenum werden die Modellierungsschritte festgesetzt, die notwendig sind, um das Spiel anzupassen. Dies sind insbesondere die Entwicklung eines geeigneten Protokolls und die Umsetzung von eben diesem im angepassten Modell.
Es folgt die Erarbeitung eines Protokolls für das Spiel in Einzelarbeit mit einer anschließenden Diskussionsrunde in Partnerarbeit. Die Ergebnisse werden präsentiert und an der Tafel in einem gemeinsamen Protokoll festgehalten (siehe TB “Protokolllösung” im Anhang).
Aufgabenbeschreibung/Problem II:
Wie muss die Modellierung erweitert werden, um das Protokoll umzusetzen?
Erarbeitung II (Plenum oder Kleingruppen):
Die SuS besprechen die Anpassungen der Modellierung im Bezug zur Einführung der Client-Serve-Architektur anhand des entwickelten Protokoll in Kleingruppen. Die Gruppen werden gemischt und es findet ein Austausch der Ideen zwischen den Gruppen statt. Die

Zusammenführung der Ideen geschieht im Plenum mit der Sicherung eines Implementatonsdiagramms an der Tafel (siehe TB “Modellierungslösung” im Anhang).

Sicherung

Als Sicherung für die Stunde gelten das Protokoll und das Implementationsdiagramm. Das weitere Vorgehen wird mit den Schülerinnen und Schülern abgesprochen, so dass der Entwicklungsprozess in der Hand der Schülerinnen und Schüler bleibt. Als Hausaufgabe bietet sich an, das Modell fertig zu stellen, es zu dokumentieren oder aber auch einzelne Komponenten und deren Wahl zu begründen.

Hilfen

Vor allem bei der Modellierung werden erfahrungsgemäß einige Dinge nicht beachtet, die für die Modellierung notwendig sind. Im Zuge dessen sollten in der zweiten Erarbeitungsphase nach Bedarf die Szenarienkarten (AB “Szenarienkarten” im Anhang) an die Gruppen ausgeteilt werden.

Implementierung eines Client-Server-Ratespiels

In dieser Stunde werden die Ergebnisse der Stunde 1 aufgegriffen indem das Implementationsdiagramm umgesetzt wird. Eine Aufteilung in Client- (leicht) und Server (mittel) -Implementierung kann als Differenzierung genutzt werden. Je nach Leistungsstand und Arbeitsstand der Schülerinnen und Schüler müssen ggf. auch vorgefertigte Klassen als Arbeitsgrundlage von der Lehrperson zur Verfügung gestellt werden. Vor allem die Implementierungen zur Datenbank können hier evtl. wieder benutzt werden. Eine Thematisierung der Wiederverwendbarkeit von Code (Design Pattern) ist an dieser Stelle möglich und sinnvoll.
Am Ende sollte idealerweise eine Implementierung des Programms vorliegen.

Lernziele

Die Schülerinnen und Schüler …

- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I)
- implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I)

- entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I)
- erläutern das Prinzip der Nebenläufigkeit (A)

Problemorientierung

Die Problematisierung ergibt sich aus der Implementierung. Darüber hinaus lässt sich an dieser Stelle die Nebenläufigkeit von Prozessen und die Ressourcenverwaltung thematisieren, die alle Clients auf eine Liste von Spielen zugreifen müssen. Diese muss vor doppeltem Zugriff geschützt werden.

Stundenverlauf

Einstieg:
Auf der Grundlage der Thematisierung des Implementationsdiagramms aus der letzten Stunde werden die Implementierungsaufgaben festgelegt und evtl. kurz besprochen.
Aufgabenbeschreibung/Problem:
Wie müssen die modellierten Methoden programmiert werden?
Erarbeitung:
Die Schülerinnen und Schüler implementieren die modellierten Methoden und stellen diese im Anschluss daran im Plenum vor.
Sicherung:
Projekte der Schülerinnen und Schüler, die die besprochenen Methoden ggf. verbessern oder von den Mitschülern übernehmen (evtl. GitHub)

Einstieg:
Ein Rollenspiel zur Kommunikation von drei Clients mit dem Server (RS C-S-Kommunikation Ressourcenzuriff im Anhang)
Aufgabenbeschreibung/Problem:
Die Zugriffe auf der die Ressource List im Server können problematisch werden. Wie kann man das Verhindern?
Lehrervortrag:
Die Lehrperson stellt die Nebenläufigkeit und die Maßnahmen in Java in Form eines Lehrervortrags und eines Tafelbildes (TB Nebenläufigkeit im Anhang) vor.
Erarbeitung:
Integration der individuellen in eine gemeinsame Lösung am Rechner.
Sicherung:

Projekte der Schülerinnen und Schüler, die die besprochenen Methoden ggf. verbessern oder von den Mitschülern übernehmen (evtl. GitHub)

Sicherung

Die Schülerinnen und Schüler reflektieren das Gelernte und Erarbeitete und planen die nächsten Schritte. Diese wären nach der erfolgreichen Implementierung die Vorbereitung der Testphase. Dazu sollten die Schülerinnen und Schüler Szenarien festlegen, die, angelehnt an das Protokoll, die Funktionalität des Programms testen. Insofern die fertige Implementierung noch nicht vorliegt, kann diese auch als Hausaufgabe aufgegeben werden.

Testen des Programms mit verschiedenen Szenarien

In dieser Stunde sollen die Schülerinnen und Schüler Szenarien entwickeln, um das fertige Programm und dessen Funktionalität zu testen. Dazu erstellen die Schülerinnen und Schüler alle möglichen Szenarien, die beim Einsatz der Software geschehen können und protokollieren die Ergebnisse. Die Ergebnisse werden besprochen und führen zu Anpassungen/Verbesserungen des Codes.

Lernziele

- testen Programme schrittweise anhand von Beispielen (I)
- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)
- testen und korrigieren Computerprogramme(I)
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K)
- präsentieren Arbeitsabläufe und -ergebnisse (K)

Problemorientierung

Die Problemorientierung ergibt sich aus dem entwickeln der Szenarien und den gewonnenen Ergebnissen. Kommt es hier zu Fehlern oder unerwünschten Verhalten des Programms muss dies zu einer Korrektur führen.

Stundenverlauf

Einstieg:
Die Ergebnisse der letzten Stunden werden nochmals mündliche besprochen und der Softwareentwicklungsprozess noch einmal verdeutlicht (TB Sotwareentwicklungsprozess im Anhang).

Aufgabenbeschreibung/Problem:
An der aktuellen Stelle muss die Funktionsfähigkeit des Programms getestet werden. Welche Szenarien bieten sich dafür an, bzw. sind notwendig dafür?
Erarbeitung:
Die Schülerinnen und Schüler erarbeiten in Partnerarbeit mindestens fünf Szenarien, mit denen das System getestet werden soll. Sie orientieren sich bei der Planung der Szenarien an einer Vorgabe (AB Szenarienplanung im Anhang). Die fertigen Szenarien werden kurz vorgestellt. Insofern dopplungen aufgetreten sind, sollen die Schülerinnen und Schüler diese in einer Austauschphase zu jeweils einer Lösung zusammenführen. Es ergeben sich so die Szenarien die anschließend für die Testphase verwendet werden.
Sicherung:
Die Szenarien, die am Ende der Erarbeitungsphase vorhanden sind.

Einstieg:
Der Softwareentwicklungsprozess wird nochmals gezeigt. Außerdem wird besprochen, wie die zu erwartenden Ergebnissen dokumentiert werden soll (AB Dokumentation Testphase).
Aufgabenbeschreibung/Problem:
Durchführen der Szenarien und entwickeln erster Problemlösungen für evtl. auftretende Probleme.
Erarbeitung:
Den Schülerinnen und Schülern werden die selbst erstellten Szenarien zugeordnet. Diese stellen die beschriebenen Testparameter her und führen das Szenario durch. Die Ergebnisse halten Sie auf einem Arbeitsblatt fest. Anschließend werden die Ergebnisse im Plenum besprochen.
Sicherung:
An der Tafel werden aufgedeckte Probleme festgehalten und Lösungsansätze kurz besprochen und notiert (TB Ergebnisse Testphase im Anhang).

Sicherung

Am Ende sollten die entwickelten Szenarien sowie die Dokumentation der Durchführung und die Testergebnisse gesichert sein. Insofern noch Probleme aufgetreten sind sollten diese als Hausaufgabe gelöst werden.

SQL-Injection

Für diese Stunde muss ein lauffähiges Programm vorhanden sein. Insofern dies schon aus der letzten Stunde bekannt gewesen ist, kann man die Schülerversion verwenden. Ansonsten müsste eine Lehrerversion zur Verfügung gestellt werden. Auf dieser Grundlage werden die Schülerinnen und Schüler dazu animiert, das selbst programmierte Spiel nun auch auszuprobieren. Die Schülerinnen und Schüler spielen also und füllen mit ihren Namen die Highscore Datenbank. Es folgt die von der Lehrperson durchgeführte SQL-Injection. Die Schülerinnen und Schüler erarbeiten im Anschluss auf verschiedenen Gebieten das Thema SQL-Injection und stellen diese vor.

Lernziele

Die Schülerinnen und Schüler ...

- untersuchen anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystem und des Datenschutzes (A)
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenslagen (A)
- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk entwickeln und erweitern (M)
- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A)

Problemorientierung

Aufgrund der Durchführung der SQL-Injection wird den Schülerinnen und Schülern vor Augen geführt, dass sie einen Aspekt der Softwareentwicklung nicht berücksichtigt haben, nämliche die Sicherheit. Die Fragen, die die Schülerinnen und Schüler entsprechend stellen sind:

- Was war das für ein Angriff?
- Wie können wir das Programm davor schützen?
- Welche rechtlichen Konsequenzen ergeben sich für Entwickler und Angreifer?

Stundenverlauf

Einstieg:

Die Schülerinnen und Schüler spielen das selbst entwickelte Spiel und füllen zunächst die Highscoreliste mit normalen Ergebnissen. Im Anschluss daran begibt sich die Lehrperson in die Rolle des Angreifers (mit Verkleidung: Kapuzenpullover, Sonnenbrille) und führt die SQL-Injection durch.

Aufgabenbeschreibung/Problem:

Was für ein Angriff war das? => AB SQL-Injection Technik

Wie können wir das Programm vor dem Angriff schützen? => AB SQL-Injection Gegenmaßnahmen

Warum ist unser System angreifbar? => AB AB SQL-Injection Codeanalyse

Erarbeitung:

Die Schülerinnen und Schüler bilden Kleingruppen. Innerhalb dieser dürfen sie sich je nach Neigung einer der aufgeworfenen Fragen widmen, so dass innerhalb einer Gruppe alle Themen abgedeckt sind. Die Ergebnisse werden innerhalb der Gruppe ausgetauscht. In der anschließenden Plenumsphase wird zusammenfassend erörtert, welche Auswirkungen die Thematik auf die Bereiche Softwareentwicklung, Rechtsprechung und Gesellschaft hat. Hierzu entsteht ein entsprechendes Tafelbild (TB Auswirkungen SQL-Injection im Anhang).

Sicherung:

Als Sicherung dienen die eigenen Notizen, sowie das Tafelbild.

Sicherung

Die Schülerinnen und Schüler sollen über die Kenntnis der Angriffstechnik einen Einstieg in die Diskussion rund um das Thema Datenschutz und Datensicherheit bekommen. Die Diskussion darüber, welche Auswirkungen solche Techniken auf verschiedene Bereiche hat, soll im Fokus stehen.

Schließen der Sicherheitslücke

Lernziele

Die Schülerinnen und Schüler ...

- analysieren und erläutern Algorithmen und Programme (A)
- modifizieren Algorithmen und Programme (I)
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D)
- entwickeln iterative und rekursive Algorithmen unter Verwendung von dynamischen Datenstrukturen (I)
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I)

Problemorientierung

Die Problemorientierung entsteht aus dem Aufdecken der Sicherheitslücke und der Rolle der Schülerinnen und Schüler als Entwickler.

Stundenverlauf

Einstieg:
Die erarbeiteten Inhalte aus der letzten Stunde dienen als Grundlage für den Einstieg. Versetzt man sich in die Sicht eines Softwareentwicklers, muss die Sicherheitslücke geschlossen werden.
Aufgabenbeschreibung/Problem:
Wie kann die Sicherheitslücke geschlossen werden?
Erarbeitung:
Die Schülerinnen und Schüler wiederholen die Inhalte aus der letzten Stunde zum Thema “Gegenmaßnahmen” und erarbeiten in Kleingruppen Lösungsansätze, die anschließend im Plenum vorgestellt und diskutiert werden.
Sicherung:
An der Tafel wird eine Möglichkeit festgehalten (TB Sicherheitslücke schließen im Anhang), wie die Sicherheitslücke zu schließen ist. Eine entsprechende Lösungsstrategie in Umgangssprache wird notiert.

Einstieg:
Die Lösungsstrategie dient als Grundlage für die Algorithmenentwicklung.
Aufgabenbeschreibung/Problem:
Wie sieht der Algorithmus aus, der die Benutzereingaben überprüft?
An welchen Stellen im Programm muss die Funktion eingebaut werden?
Erarbeitung:
Die Schülerinnen und Schüler entwickeln in Einzelarbeit den zuvor besprochenen Algorithmus und setzen ihn an den entsprechenden Stellen im Programm ein.
Sicherung:
Die Schülerinnen und Schüler stellen ihre Lösunge im Plenum vor.

Sicherung

Die Projekte der Schülerinnen und Schüler sind die Sicherung. Darüber hinaus sollte zum Abschluss der Sequenz/Reihe noch einmal darüber gesprochen werden, welche

Auswirkungen die gelernten Inhalte auf die Schülerinnen und Schüler selbst haben. Da die allerdings individuelle Sichtweisen sind, bedarf es keiner weiteren schriftlichen Sicherung.

Diskussion der Rechtlichen Rahmenbedingungen

Lernziele

Die Schülerinnen und Schüler ...

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K)
- beurteilen Informatiksysteme hinsichtlich ihrer Möglichkeiten, Grenzen und Auswirkungen ()
- untersuchen anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystem und des Datenschutzes (A),
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenslagen (A)

Problemorientierung

Welche gesetzlichen Regelungen gibt es in Bezug auf SQL-Injections?

Stundenverlauf

Einstieg:
Auf der Grundlage der bisherigen Ergebnisse wird nach den gesetzlichen Rahmenbedingungen gefragt. Es wird also den Fragen nachgegangen, ob sich der “Hacker” mit seinem Angriff strafbar gemacht hat und was Daten, die mit SQL-Injections abgegriffen werden können, überhaupt Wert sind.
Aufgabenbeschreibung/Problem:
Ist eine SQL-Injection strafbar? (AB Datendiebstahl)
Welchen Wert haben Daten überhaupt? (AB Datenwert)
Sind wir für den “Nicht-Schutz” von Daten/Systemen haftbar? (AB Datenschutz)
Erarbeitung:
Die Schülerinnen und Schüler bearbeiten arbeitsteilig (TPS) die ABs Datenschutz, Datenwert und Datendiebstahl. Anschließend stellen sich die Schülerinnen und Schüler die Ergebnisse gegenseitig vor.

Sicherung:

Die Schülerinnen und Schüler erläutern im Plenum die Ergebnisse und erste Fragestellungen vom Anfang der Stunde werden beantwortet und festgehalten (TB Rechtliche Rahmenbedingungen).

Einstieg:

Aufgrund der Zwischenergebnisse zeigt sich, dass noch nicht klar ist, nach welchen Gesetzen Hacker für ihre Vergehen bestraft werden können.

Aufgabenbeschreibung/Problem:

Ist eine SQL-Injection strafbar?

Erarbeitung:

In den zuvor gebildeten Gruppen wird das AB Gestz $202a bearbeitet. Hierbei sollen die Schülerinnen und Schüler zunächst ein Urteil für den Hacker diskutieren und begründen. Anschließend soll diskutiert werden, inwieweit der Nicht-Datenschutz und der Datendiebstahl/-manipulation unter Strafe steht. Abschließend erarbeiten die Schülerinnen und Schüler ein Gesetz zum Datenschutz.

Sicherung:

Der Gesetzestext wird im Tafelbild (TB Rechtliche Rahmenbedingungen) festgehalten.

Sicherung

In diesen Stunden steht vor allem der Austausch zwischen den Schülerinnen und Schülern im Vordergrund. Sie sollen fundiert und auf der Grundlage von rechtlichen Gesetzen und Verordnungen die Situation bewerten und die Ergebnisse darstellen. Die Sicherung gelingt über die eigentlichen Diskussionen während der Arbeitsphase und im Tafelbild.

Lernstandsüberprüfung

Die Lernstandsüberprüfung am Ende der Reihe muss angepasst werden an das Vorgehen in der Reihe. Trotzdem lassen sich viele Elemente aus allen Vorgehen heraus identifizieren, die in jedem Fall von den Schülerinnen und Schülern gekonnt werden sollten. Außerdem sollten die schon genannten Kompetenzen in der Überprüfung zumindest zum Teil abgebildet sein. Eine mögliche Überprüfung ist im Anhang zu finden (LÜ SQL-Injection).

Erfahrungen bei der Umsetzung

Bei der Umsetzung der Reihe im GK und LK fielen folgende Zusammenhänge und Besonderheiten auf:

- Der LK war aufgrund der vorherigen Implementierung wesentlich interessierter an der Verbesserung des Codes.

- Der GK dachte, man hätte eine Methode extra für den Angriff geschrieben. Sie haben den Live-Hack als solchen, trotz Rollenspiel, nicht erkannt.
- Der Begriff des Hacking ist im inhaltslosen Raum stark motivierend - Sobald die SuS mehr darüber erfahren und ggf. die Methoden selbst anwenden ist der “Zauber” verflogen.
- Anwendbarkeit begrenzt, aber die SuS probieren die Angriffe “überall” aus. Hinweise auf die Strafbarkeit sind wichtig.
- Es besteht die Möglichkeit einen eigenen Webshop auf einem virtuellen Server zu hosten, der zu Testzwecken per SQL-Injection „gehackt“ werden kann. => Internetseite hackTheBox (noch nicht ausprobiert)

Anhang

AB Gesetz §202a

Im Strafgesetzbuch ist in Abschnitt 15 mit dem Titel “Verletzung des persönlichen Lebens- und Geheimbereichs” geregelt, wie Verstöße zu handhaben sind, die einen Eingriff in den persönlichen Lebens- und Geheimnisbereich bedeuten.
Der Paragraph §202a ist auch als “Hackerparagraph” bekannt und umfasst die folgenden Absätze:

Ausspähen von Daten

(1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

Arbeitsaufträge:

1. Wenden Sie das Gesetz auf den “Hacker” ihres Systems an. Fällen Sie ein Urteil und begründen Sie es entsprechend.
2. Diskutieren Sie die Ausgewogenheit der beiden rechtlichen Rahmen §202a und DSGVO in Bezug auf die Bestrafung von unzureichendem Datenschutz auf der einen und Datendiebstahl/-manipulation auf der anderen Seite.
3. Formulieren Sie gemeinsam ein Gesetz zum Datenschutz.

AB Datendiebstahl

1. Hotel: Türen zu: Hacker legen das Schließsystem des Hotels Jägerwirt in Kärnten lahm, so dass im voll ausgebuchten Vier-Sterne-Haus die Türen zu den Hotelzimmern nicht mehr zu öffnen sind.
Ein klassischer Internet-of-Things-Hack: Hier wurden technische Geräte manipuliert, die über ein zentrales Steuerungssystem verfügen und für den Betriebsablauf eines Hotels von elementarer Bedeutung sind. Durch eine Hintertür im System konnten die Angreifer insgesamt drei Mal das System manipulieren und den Betreiber erpressen

2. Clinton: E-Mails geklaut: Private E-Mails von Hillary Clinton werden von Hackern gestohlen und gelangen so an die Öffentlichkeit.
Dieser Hack des Clinton-E-Mail-Servers während des US-Wahlkampfes erregte viel Aufsehen. Die von Hackern veröffentlichten E-Mails stammten aus der Zeit als Hillary
Clinton Außenministerin war und ihre dienstliche Kommunikation über private Server und E-Mail-Adressen führte. Dabei waren Gesprächsprotokolle, die sie nach den Regularien zum Umgang mit geheimen Informationen nicht mehr hätte besitzen dürfen.

3. Yahoo: Daten weg: Unbekannte Hacker stehlen persönliche Daten von mehr als einer Milliarde Yahoo-Nutzern.
Im Jahr 2014 wurde von Yahoo eine Datenbank mit 500 Millionen Kundendatensätzen gestohlen. Der Hack wurde jedoch erst 2016 bekannt. Wenige Monate später wurde ein weiterer Datendiebstahl mit mehr als einer Milliarde betroffener Kunden aus dem Jahr 2013 bekannt. Die Sicherheitslücke war vermutlich eine falsche Verifikation der in Cookies gespeicherten Authentifizierungsdaten.

4. DNS-Server: Mirai greift an
Der Angriff richtete sich zum ersten Mal nicht gegen einzelne Dienste, sondern gegen die zentralen Verwaltungselemente des Internets, die DNS Server. Eine weitere Besonderheit an Mirai ist, dass es noch nie ein Botnetz in dieser Größe gab und es bis zum Zeitpunkt des Angriffs vollkommen unentdeckt blieb. Die meisten Zombies in dem Botnetz waren ungesicherte IoT-Geräte.

(Quelle: https://berlinvalley.com/cyberangriffe-beispiele)

Arbeitsaufträge:

1. Beschreiben Sie das Vorgehen und das Ziel der Hacker in den einzelnen Beispielen.
2. Ordnen Sie die Beispiele für Datendiebstahl in Kategorien ein.
3. Erläutern Sie, wieso immer wieder Daten Ziel von Angriffen sind.

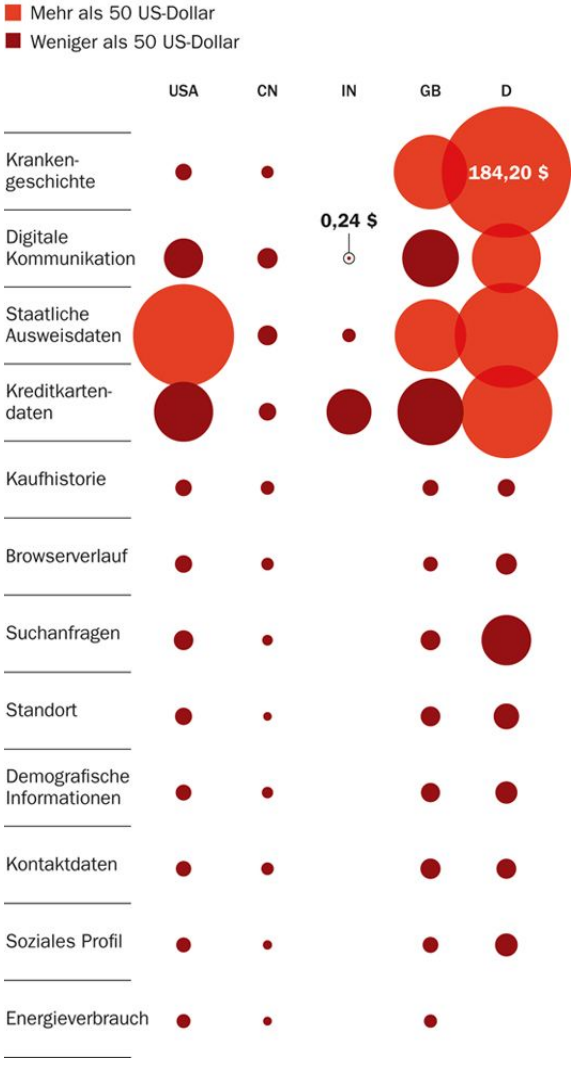
AB Datenwert

Persönliche Daten werden weltweit höchst unterschiedlich bewertet. Die persönliche Krankengeschichte, Ausweis- und Kreditkartendaten oder der Browserverlauf: Ihr Preis schwankt zwischen wenigen Cent und dreistelligen Dollarbeträgen.

Umfragen unter Verbrauchern in den USA, in China, Indien, Großbritannien und Deutschland zeigen, dass die Menschen bestimmten Informationen einen deutlichen höheren Wert beimessen als anderen. Die folgende Grafik zeigt die Unterschiede.

Der Preis der Daten

Zwar sorgen sich Verbraucher darum, wie Unternehmen ihre persönlichen Daten speichern und nutzen. Sie wissen aber erstaunlich wenig darüber, welche Informationen sie im Internet offenlegen - und die meisten Unternehmen klären sie auch nicht darüber auf. Das untergräbt das Vertrauen der Kunden und deren Bereitschaft, Informationen preiszugeben.



Die Branche der Datenhändler in Deutschland sitzt trotzdem auf wahren Schätzen. Auch wenn es pro Person um kleine Beträge geht: Eine Adresse, angereichert mit verschiedenen Merkmalen vom Wohnort, über Leseinteressen bis zur Mobilität, ist schon für 25 Cent zu haben. Peanuts könnte man meinen! Aber mit den Gigabytes an Daten ist Gewinn zu machen. Das Bertelsmann Unternehmen AZ Direct verfügt über eine Datenbank mit 600 Merkmalen für 70 Millionen Konsumenten.

<u>Arbeitsaufträge:</u>

1. Vergleichen Sie die Beträge, die Befragte aus unterschiedlichen Länder bereit sind zu zahlen, um bestimmte Daten zu schützen.

2. Erläutern Sie, wieso Daten in unserer heutigen Gesellschaft etwas wert sind.

3. Stellen Sie Vermutungen auf, welche Unternehmen Interesse am Datenkauf haben.

(Quelle: http://www.harvardbusinessmanager.de/blogs/was-sind-ihnen-ihre-daten-wert-a-1061019.html) Ungefährer Beitrag in US-Dollar, den die Befragten zahlen würden, um bestimmte Daten zu schützen (verglichen nach Kaufkraftparität).

AB Datenschutz

Hinsichtlich der Datensicherheit gibt es neue Vorgaben durch die DSGVO. Die DSGVO sieht vor, dass Unternehmen künftig bei risikobehafteten Datenverarbeitungen eine „Datenschutz-Folgenabschätzung“ nach bestimmten Kriterien durchzuführen haben. Doch nicht nur das: Falls sich aus der Folgenabschätzung ergibt, dass ein Risiko für die Betroffenen besteht, müssen Unternehmen oder Behörden die für sie zuständige Aufsichtsbehörde darüber informieren. Diese soll dann binnen acht Wochen schriftliche Empfehlungen an das Unternehmen zur Risikominimierung geben; sie darf die Datenverarbeitung aber auch untersagen.
Diese in Artikel 34 geregelte „vorherige Konsultation“ ist eines der folgenreichsten „Institute“ des neuen EU-Datenschutzrechts. Kritiker bemängeln sich hieraus ergebende Innovationshemmnisse. Denn Unternehmen können in Zeiten von agiler Softwareentwicklung häufig nicht die Entscheidung von Aufsichtsbehörden abwarten, zumal deren Stellungnahmefrist von acht Wochen auch noch verlängert werden kann.

Teure Verstöße

Das neu geschaffene europäische Datenschutzrecht ändert nicht alles – aber doch vieles. Es erweitert in vielen Punkten die Rechte der Verbraucher. Für Unternehmen entstehen dagegen eine ganze Reihe von neuen Pflichten. Herauszuheben ist hier vor allem die Pflicht zur Schaffung eines Datenschutzmanagements, die viele betroffene Mittelstandsunternehmen oder Selbstständige vor erhebliche Herausforderungen stellen wird. Auch wenn das Inkrafttreten noch weit entfernt klingt, sollten sich Unternehmen rechtzeitig mit der Umsetzung beschäftigen und sich auf die anstehenden Änderungen vorbereiten. Noch stehen die Datenschutz-Regeln in Deutschland im Ruf, ein weitgehend zahnloser Tiger zu sein. Über die im BDSG festgelegten Sanktionen schmunzelten viele Unternehmer. Das maximale Bußgeld in Höhe von 300 000 Euro pro Einzelfall wurde nur sehr selten abgerufen. Und gerade bei den US-Konzernen, die das deutsche Recht oft bewusst ignorieren, läuft eine solche Summe eher in der Kategorie „Portokasse“. mBald drohen auf Basis der DSGVO ganz andere Hausnummern, was zweifellos die Akzeptanz des Datenschutzes und dessen Bedeutung im Compliance-Bereich erhöhen wird. Damit die drohenden Geldbußen wie in Artikel 79 verlangt auch „in jedem Einzelfall wirksam, verhältnismäßig und abschreckend“ sein können, sind nun Strafzahlungen bis zu 20 Millionen Euro oder bis zu 4 Prozent des gesamten weltweit erzielten Jahresumsatzes eines Konzerns möglich. Beendet hat die EU übrigens quasi en passant auch einen uralten Streit unter Juristen, nämlich die Frage, ob es sich bei IP-Adressen um personenbezogene Daten handelt, die dem Datenschutz unterfallen. In den Erwägungsgründen der Verordnung wird klargestellt, dass sich Nutzer beziehungsweise Anschlussinhaber anhand dieser Informationen identifizieren lassen. Es kommt nicht darauf an, ob dies beispielsweise nur für den Provider möglich ist, es reicht vielmehr die abstrakte Möglichkeit einer Identifikation.

(Quelle: Christiane Schulzki-Haddouti, Datenschutz europaweit, Was sich mit der EU-Grundverordnung ändern wird, c’t 03/16, S. 86.)

Arbeitsaufträge:

1. Beschreiben Sie die Veränderungen des Datenschutzes durch die DSGVO.
2. Erläutern Sie die Folgen, dass eine IP-Adresse als personenbezogene Daten zählt.
3. Beurteilen Sie, ob die DSGVO zur Datensicherheit der Verbraucher beiträgt.

AB Szenarienkarten

Szenario 1

1. Voraussetzung: Server läuft
2. Durchführung: Ein Client meldet sich am Server an
3. Ergebnis: Der Client ist am Server angemeldet und bekommt eine entsprechende Rückmeldung

Szenario 2

1. Voraussetzung: Server läuft, Ein Client ist bereits angemeldet
2. Durchführung: Ein weiterer Client meldet sich am Server an
3. Ergebnis: Der Client ist am Server angemeldet und bekommt eine entsprechende Rückmeldung

Szenario 3

1. Voraussetzung: Server läuft, Zwei Clients sind bereits angemeldet
2. Durchführung: Ein Client beginnt ein Spiel. Während Client 1 spielt, beginnt auch der zweite Client mit dem Spiel. Beide beenden das Spiel. Client 2 fragt den Spielstand beim Server an.
3. Ergebnis: Das Spiel beider Clients befindet sich in der Datenbank

Szenario 4

1. Voraussetzung: Server läuft, Ein Client ist bereits angemeldet
2. Durchführung: Ein Client beginnt ein Spiel. Während des Spiels schickt der Client einen falschen Wert (Zeichen) an den Server
3. Ergebnis: Der Client bekommt einen Versuch hinzu und der Server fragt nach einer weiteren Zahl

Szenario 5

1. Voraussetzung: Server läuft, Ein Client ist bereits angemeldet
2. Durchführung: Ein Client beginnt ein Spiel. Während des Spiels beendet der Client die Verbindung
3. Ergebnis: Das Spiel des Clients wird nicht gewertet. Der Client wird aus der Spielliste entfernt.

Arbeitsaufträg:

1. Bearbeiten Sie drei der vorgeschlagenen Szenarien so, dass diese wirklich vollständig sind.
2. Bearbeiten Sie das AB Szenarienplanung.

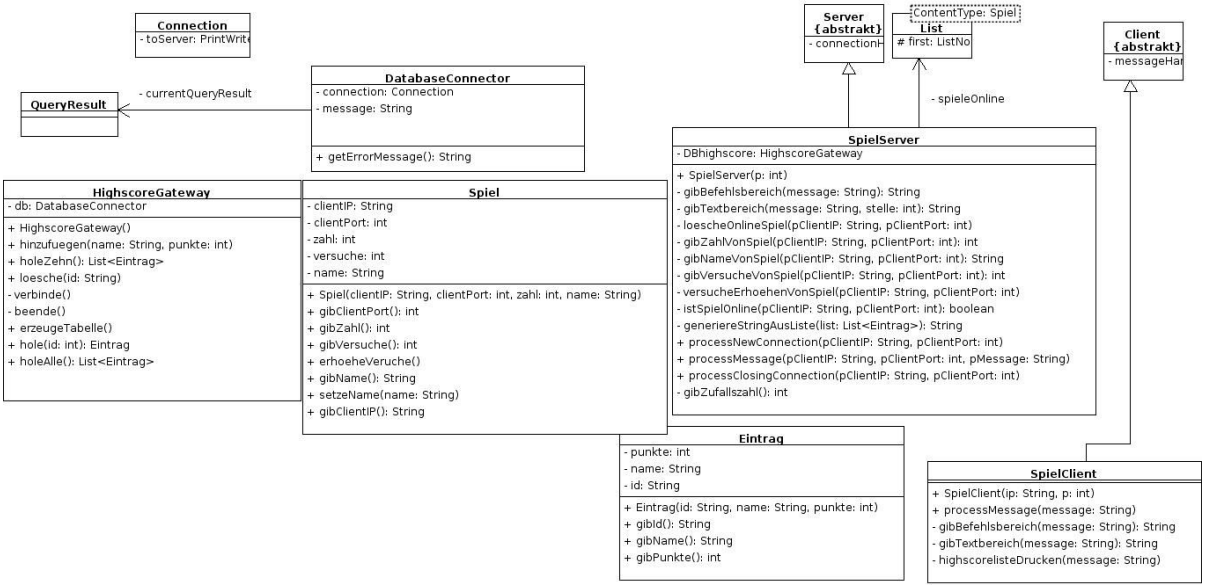
TB Protokolllösung

Sequenzdiagramm (Mitte) und Implementationshinweise (Außen)

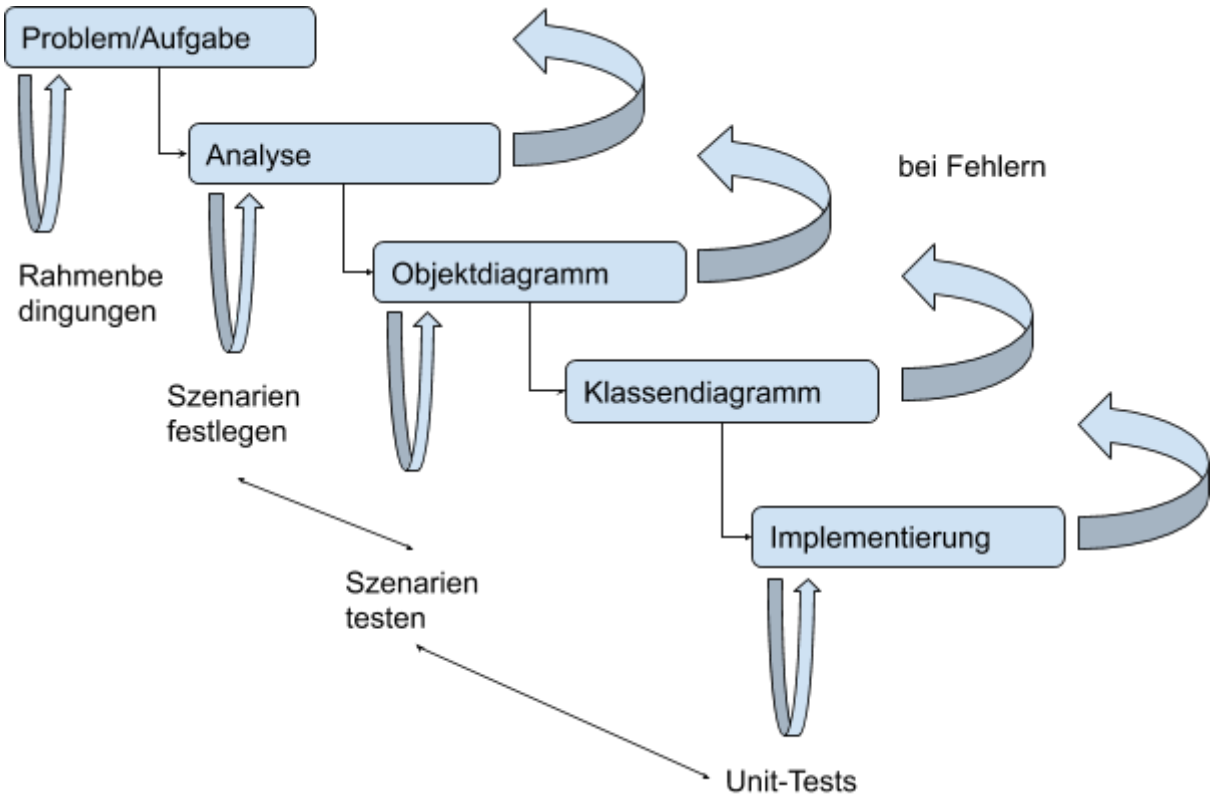
Implementierung im Server	Client zum Server	Server zum Client	Implementierung im Client
Client meldet sich am Server an. Client wird in die Liste der angeschlossenen Clients aufgenommen => ProcessNewConnection	Am Server Anmelden	OK	Nachricht vom Server ausgeben => ProcessMessage
Neues Spiel anlegen mit Zufallszahl => ProcessMessage	PLY <Name>	PLY Spiel gestartet	Nachricht vom Server ausgeben => ProcessMessage
Highscore aus der Datenbank holen und an den Client senden => ProcessMessage	HSC	HSC 01.Spielername | Punkte, 02. Spielername | Punkte, ...	Nachricht vom Server ausgeben => ProcessMessage
Client wird nach dem Senden der Nachricht aus der Clientliste entfernt => ProcessClosing Connection	END	END Auf Wiedersehen!	Nachricht vom Server ausgeben => ProcessMessage
Server überprüft die gesendete Zahl mit der im Spiel gespeicherten Zufallszahl und schickt eine entsprechende Nachricht zurück => ProcessMessage	SND<Zahl>	NUM Wenn Zahl falsch => NEG Leider falsch! Rate noch einmal, die Zahl war größer/kleiner Wenn Zahl richtig => POS Richtig geraten, du hast x Versuche dafür gebraucht!	Nachricht vom Server ausgeben => ProcessMessage

Das Protokoll muss nun in der Methode processMessage beim Server und beim Client implementiert werden. Beim Server müssen auch die Methoden processNewConnection und processClosingConnection nach dem Protokoll implementiert werden.

TB Modellierungslösung



TB Softwareentwicklungsprozess



TB Nebenläufigkeit

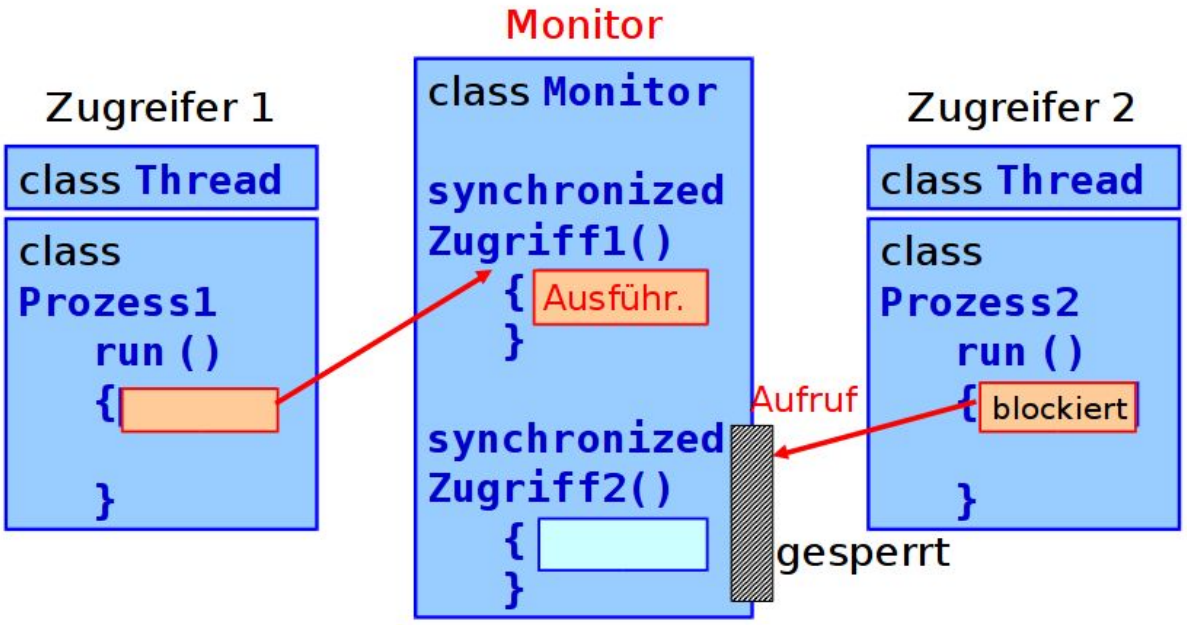


Abbildung aus der Vorlesung zum Nebenläufigen Rechnen, Prof. Echtle, Uni Duisburg-Essen

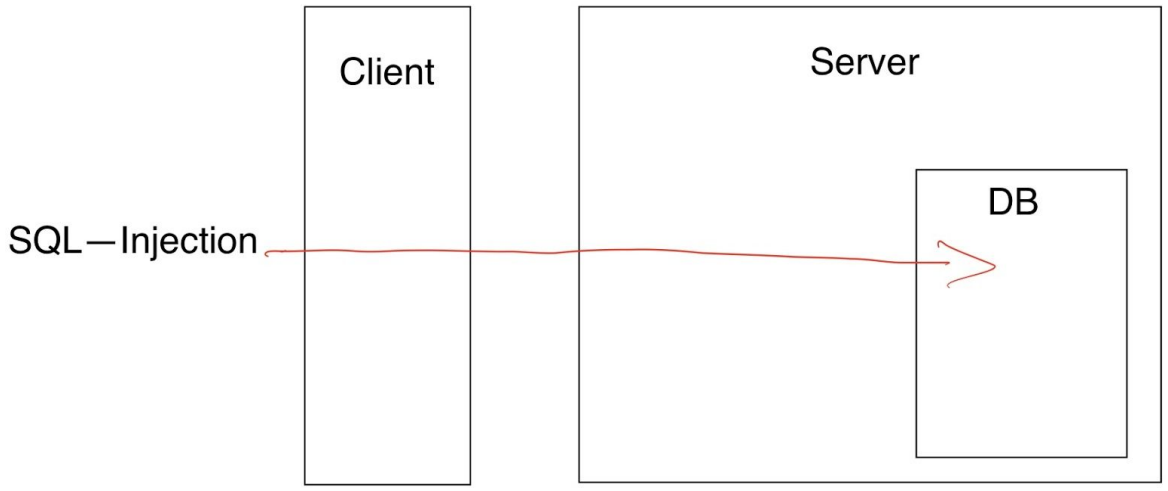
Da der Server intern sowohl die neuen Verbindungen, als auch ankommende Nachrichten als Threads anlegt, lassen sich auch interne Ressourcen wie Datenspeicher oder Datenbankzugriffe exklusiv vergeben.

TB Ergebnisse Testphase

Gefundene Fehler	Mögliche Lösungsansätze
Anmeldung Funktioniert nicht	• Server nicht verfügbar. Verfügbarkeit prüfen
Es kann immer nur ein Spieler spielen	• Die Liste im Server läuft nicht korrekt. Methoden überprüfen
...	• ...

=> Verteilen Sie die Aufgaben und beheben Sie die Fehler. Testen Sie anschließend erneut.

TB Auswirkungen SQL-Injection



Die SQL-Injection kann direkt vom Client ungefiltert auf den Server gelangen. Selbst der Zugang zur Datenbank ist ungeschützt. Auf der Datenbank wird der Befehl dann ausgeführt. Ein unerlaubter Zugriff auf die Datenbank ist möglich!

TB Sicherheitslücke schließen

Beobachtung: Der Angriff wurde durch die Verwendung von Zeichen möglich, die das normale SQL-Statement teilweise auskommentieren und eigene Eingaben berücksichtigen.

Gegenmaßnahme: Ein SQL-Statement darf nicht ungeprüft auf der Datenbank ausgeführt werden. Es müssen „Filter“ eingebaut werden, die ein SQL-Statement daraufhin überprüfen, ob bestimmte Zeichen vorhanden sind.

Ort der Gegenmaßnahme: Am sinnvollsten erscheint eine Überprüfung im Server, da der Client ebenfalls manipuliert werden könnte. In ProcessMessage oder spätestens in den Gateway-Klassen muss eine Überprüfung erfolgen.

TB Rechtliche Rahmenbedingungen

Der Datenschutz ist in mehreren Gesetzen, hier vor allem StGB §202 und Verordnungen geregelt. Eine gesetzliche Pflicht Daten zu schützen gibt es nicht. Der Einsatz von Techniken, die zu unerlaubten Datenbesitzt führen, sind strafbar!

RS C-S-Kommunikation Ressourcenzugriff

Das Rollenspiel soll verdeutlichen, welche Komponenten die beteiligten Klassen benötigen. Dazu werden drei Spieler, ein Server, ein Prozessmanager und viele Nachrichtenüberbringer benötigt. Auf der Grundlage des zuvor fertiggestellten Protokolls wird dann ein normaler Ablauf mit drei Clients simuliert.
Wichtig hierbei ist, dass der Server über die Ressourcen Datenbank und Spielliste verfügen kann. Aufgrund der Implementierungen muss der Server nun die Befehle ausführen, die der Prozessmamager bekommt. An dieser Stelle muss darauf geachtet werden, dass der Code genau ausgeführt wird, da dies zu Problemen führt, die thematisiert werden sollen.
Beobachtungen:
Der Server muss Anfragen eigentlich „parallel“ verarbeiten, was zu Problemen führt.

- Der Pozessmanager wechselt immer wieder zwischen den Prozessen um eine möglichst schnelle Bearbeitung zu gewährleisten.
- Zugriff auf Ressourcen (Spieleliste, Datenbank) problematisch.

=> Forderung: Ressourcen müssen exklusiv verwendet werden!

AB Szenarienplanung

Die Funktionsfähigkeit eines informatischen Systems sollte vor der Einführung ausführliche getestet werden, um Fehler zu beheben und bestimmte Features zu testen. Zu diesem Zweck werden sogenannte “Unit-Tests” im Softwareentwicklungsprozess beschrieben, die die einzelnen Teile des Systems testen.
Darüber hinaus sollte aber auch das Gesamtsystem mit Hilfe definierter Szenarien getestet werden. Vor allem bei der Server-Client-Architektur ist dies essentiell, da somit die Korrektheit der Protokollimplementierung gewährleistet werden kann.

Arbeitsaufträge:

1. Erarbeiten Sie in Partnerarbeit mindestens 5 Szenarien, mit denen das System getestet werden kann.
    a. Beschreiben Sie dazu das Szenario so genau wie möglich (Voraussetzung, Durchführung, erwartetes Ergebnis)
2. Führen Sie ihre Szenarien der Reihe nach durch
3. Dokumentieren Sie die Ergebnisse und Diskutieren Sie auftretende Probleme, so dass Sie in der Lage sind, die Probleme zusammen mit einer möglichen Fehlerquelle und einer ersten Lösung vorzustellen

LÜ SQL-Injection

1. Erläutern Sie die Angriffsmethode der SQL-Injection.
2. Geben Sie eine Stelle in der Server-Client-Architektur an, an der verhindert werden kann, dass eine SQL-Injection erfolgen kann. Erläutern Sie.
3. Implementieren Sie eine Methode pruefeSQL(String sql), die die Aufgabe der Überprüfung erfüllt.
4. Stellen sie dar, welche gesetzlichen Rahmenbedingungen in Bezug auf die Bestrafung einer SQL-Injection gelten.

Mögliche Lösungen

1. Die SQL-Injection benötigt eine Eingabe seitens des Clients, die Auswirkungen auf die Datenbank hat. In dieser Eingabe gibt man nun mit Hilfe von bestimmten Zeichen einen weiteren SQL-Befehl ein. Das DBMS führt immer alle SQL-Befehle aus, die an das System delegiert wurden. Somit gelangen Anweisungen in das System, die eigentlich nicht vorgesehen sind.
2. Innerhalb der Server-Client-Architektur kann auf der Ebene des Servers eine Überprüfung des SQL-Statements erfolgen. Dies ist sinnvoll, da es sich um eine Zwischenebene handelt. Würde die Überprüfung beim Client stattfinden, könnte diese manipuliert werden.
3. Lösung siehe BlueJ-Projekt
4. Eine SQL-Injection ist laut StGB §202a strafbar und kann mit einer Freiheitsstrafe bestraft werden.

SQL-Injection – Die Technik

Allgemein:
Greift eine Anwendung auf Daten einer SQL-Datenbank zu, so werden Befehle in Form von SQL-Anweisungen an die Datenbank übermittelt. Ist die Anwendung anfällig für SQL-Injection, kann ein Angreifer durch Manipulation der Eingabedaten geänderte oder zusätzliche SQL-Anweisungen „injizieren“, die von der Anwendung an die Datenbank weitergeleitet und dort bearbeitet werden. Auf diese Weise können wie bei einem direkten Datenbankzugriff beliebige SQL-Anweisungen ausgeführt werden und so Sicherheitsmechanismen der Anwendung beim Datenzugriff umgangen werden.

Funktionsweise:
SQL-Injections nutzen eine Sicherheitslücke beim Ausführen von SQL-Anweisungen aus. Diese können nämlich auch aneinandergereiht durch ein Semikolon getrennt an den Datenbanktreiber gesendet werden, der dann alle Anweisungen ausführt.
Normale SQL-Anweisung wäre beispielsweise ein *SELECT * FROM highscore WHERE id = 1;* Da der Wert 1 in der Anweisung normalerweise vom Benutzer übermittelt wird, kann der Benutzer auch den Befehl ändern, indem er beispielsweise anstelle des normalen Wertes 1 einen weiteren SQL-Befehl eingibt.

Angriff:
Relationale Datenbanken sind oft nach dem selben Prinzip aufgebaut und enthalten oft auch die selben Tabellen. Bei einer Onlineplattform mit Anmeldeformular sind in der Datenbank beispielsweise Benutzername und Passwort gespeichert. Die Datenbankanfrage sieht wahrscheinlich wie folgt aus:
*SELECT * FROM user WHERE Name = 'test' AND Passwort = '1234'*
Da die Anwendung nach der Abfrage häufig nur noch überprüft, ob die Abfrage ein positives Ergebnis liefert, kann der Benutzer über das Eingabeformular nun folgendes tun. Er gibt beim Passwortfeld den Befehl ' OR 1=1;// ein.
Setzt die Anwendung nun die Eingabe in das Statement ein, kommt die folgende SQL-Anweisung heraus:
*SELECT * FROM user WHERE Name = 'test' AND Passwort = '' OR 1=1;//'*
Wertet der Datenbanktreiber den Ausdruck aus, so überprüft er, ob es einen user mit dem Namen test und keinem Passwort gibt. Hier wird mit Sicherheit *false* heraus kommen. Da anschließend aber ein logisches ODER verwendet wurde und 1=1 true ausgibt, wird insgesamt *true* zurück gegeben.
Die Folge ist, dass der Benutzer auch ohne Zugangsdaten Zugriff auf das System bekommt!

<u>Aufgaben:</u>

1. Führen Sie eine SQL-Injection auf dem Papier durch auf der Grundlage des oben vorgestellten Beispiels. Gehen Sie davon aus, dass das folgende SQL-Statement mit Hilfe einer Benutzereingabe für den Namen ausgeführt wird: SELECT Name, Adresse, TelNr FROM Benutzer WHERE Name = 'Maier';
    a. Löschen Sie die Datenbank! (DROP Benutzer)
    b. Legen Sie einen neuen Benutzer an! (INSERT INTO Benutzer (Name, Adresse, TelNr) VALUES ('Mueller', 'Steinstraße 35', '023823456'))
2. Analysieren Sie die im Unterricht vorgestellte Injection Anonymous',0);// in Bezug auf die Methode *public void hinzufuegen(String name, int punkte)* in der Gatewayklasse und erläutern Sie, warum das Statement so funktioniert.

SQL-Injection – Rechtliches

Allgemeines:
Datenschutz ist seit einigen Jahrzehnten auch gesetzlich geregelt. Als Ankerpunkt dient der §202 zur Verletzung des Briefgeheimnisses im Strafgesetzbuch. Daran angegliedert sind die Paragraphen §202a bis §202d, der zusammengefasst als sogenannter *Hackerparagraph* bekannt ist. Dieser umfasst die beiden folgenden Sätze:

(1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
(2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

Beispiel (aus den USA):
Weil Cody Kretsinger im Mai 2011 Sony „gehackt" hat, wurde er zu einem Jahr und einem Tag Gefängnis verurteilt. Außerdem bekam er ein Jahr Hausarrest, 1000 Sozialstunden und eine „Wiedergutmachungszahlung" über 600.000 US-Dollar aufgebrummt. So hart es auch klingen mag – er hatte Glück. Ursprünglich hatten ihm bis zu 15 Jahre Gefängnis gedroht. Der Rest der Gruppe wartet noch auf ein Urteil, die meisten Verfahren laufen bereits. So wird die Verhandlung gegen Raynaldo Rivera vor der gleichen Kammer stattfinden, ein Urteil soll es am 16. Mai geben. Auch er hatte sich nach seiner Festnahme im August 2012 im Oktober schuldig bekannt.

Foren-Beitrag:
Gast 2008-03-10, 15:26:41
Hallo, ich spiel aus Spaß an der Freude an manchen Seiten öfters mal SQL-Injections durch. Ich muss dazu sagen, dass ich
- nie etwas manipulieren, höchstens die Möglichkeiten dazu auslote
- die gewonnenen Daten nicht weitergebe/verkaufe oder sonst wie missbrauch.
Ich schreib die Seiteninhaber danach immer an, und fordere sie auf das ganze in einem Monat zu beheben. Normalerweise wird das dann auch immer gemacht, und meistens bekommt man sogar eine nette Antwortmail mit einem Dank. Einer meiner aktuellen Fälle, hat aber auf meine Mail, in der ich auf mehrere Lücken hinwies (SQL-Injections zum auslesen von sensiblen Kundendaten, Kontoinfos etc, sogar die Möglichkeit der Manipulation der Datenbank) geantwortet, sie werden bei weiteren Mails einen Anwalt einschalten (und auch ansonsten war die Mail sehr unfreundlich). Es handelt sich um eine Firma in der Schweiz (ich bin in DE). Nach nun über 2 Monaten haben sie das immer noch nicht gefixt. Inwiefern ist es gefährlich für mich nun irgendwelche [anonymisierten] Ausschnitte der DB ins Internet zu stellen, damit die mal was machen? Immerhin könnten längst 100 anderen die Daten verkaufen und missbrauchen. Gruß[8]

<u>Aufgaben:</u>

1. In der IT-Szene war der Aufschrei zum Urteil gegen die Gruppe um Cody Kretsinger groß, da die allermeisten die Firma Sony als Hauptverantwortlichen sahen. Äußern Sie Vermutungen dazu, warum dies so gesehen werden könnte.
2. Beurteilen Sie den Gast-Post hinsichtlich der Anwendbarkeit des §202a. Versetzen Sie sich in die Lage des Richters und fällen Sie ein schriftlich begründetes Urteil!

[8] https://www.forum-3dcenter.org/vbulletin/archive/index.php/t-407388.html

SQL-Injection – Gegenmaßnahmen

Da SQL-Injections durch Benutzereingaben ermöglicht werden, müssen eben diese besonders geprüft oder aber die Eingabe unabhängig vom SQL-Statement behandelt werden.

1. Möglichkeit: Benutzereingaben prüfen

Wird dem Benutzer die Möglichkeit einer textuellen Eingabe ermöglicht und wird diese Eingabe zur Kommunikation mit einer Datenbank verwendet, besteht immer die Gefahr, dass der Benutzer die mögliche Hintereinanderreihung von SQL-Befehlen ausnutzt um auf die Datenbank in nicht vorhergesehener Art und Weise zuzugreifen. Wird beispielsweise dem Benutzer die Möglichkeit gegeben sich einen speziellen Platz in der Highscoreliste anzuzeigen, dann gelingt dies über das Statement *SELECT * FROM highscore WHERE punkte = 1;*

Die Zahl 1 ist dabei die Benutzereingabe. Wird anstelle der 1 nun aber der SQL-Befehl *1; DROP highscore;* übergeben, so führt das Statement dazu, dass die Tabelle komplett gelöscht wird. Folglich muss die Eingabe vom Benutzer vor dem Senden an die Datenbank überprüft werden. Gefiltert werden müssen:

' // ; SQL Befehle wie (SELECT, FROM, WHERE, DROP, INSERT, …)

Allerdings muss man bei der Filterung auch genau aufpassen, dass man nicht Benutzerdaten filtert, die eigentlich in Ordnung sind. Beispielsweise bei einer Anmeldung von Thomas Fromm, müsste der Filter das FROM im Nachnamen nicht als Angriff werten.

2. Möglichkeit: SQL-Statement und Benutzereingaben trennen

Hierbei werden die Eingabedaten und das SQL-Statement getrennt an die Datenbank gesendet und erst vom Datenbanksystem zusammengesetzt. Dabei übernimmt die Datenbank selbst die Filterung der Eingabedaten und maskiert diese mit entsprechenden Zeichen. Die Daten werden dann nicht mehr als Befehle interpretiert. Darüber hinaus ist eine nicht vorgesehene Aneinanderreihung von Befehlen so nicht mehr möglich, da nur noch die Parameter der Abfrage gefüllt werden. Die Methode ist unter dem Stichwort „Prepared Statements“ bekannt.

Weitere Möglichkeiten: Updates und übersichtliche Strukturen

Darüber hinaus sollten alle am Dienst beteiligten Strukturen dem aktuellen Standard in Programmierung und Strukturierung entsprechen, so dass Sicherheitslücken schnell entdeckt und geschlossen werden können. „unaufgeräumte“ Server und Datenbanksysteme bieten häufig Möglichkeiten für Angreifer. Des Weiteren sollten immer alle relevanten Patches auf den beteiligten Systemen aufgespielt werden um Sicherheitslücken der verwendeten Komponenten (Java, SQL, PHP, Server-Betriebssystem, etc.) auszuschließen oder aber zumindest schnell zu schließen.

<u>Aufgaben:</u>

1. Erläutern Sie den Unterschied zwischen den beiden vorgestellten Methoden und beurteilen Sie diese hinsichtlich ihrer Sicherheit.
2. Geben Sie die Stellen im Programmcode an, an denen die Benutzereingaben für das Ratespiel geprüft werden müssten und erläutern Sie ihre Entscheidungen kurz.
3. Modellieren Sie eine Java-Methode, die die Filterung der Eingabe übernimmt. Recherchieren Sie dazu kurz die von Java bereitgestellten Methoden charAt(), contains(), toLowerCase(), toUpperCase().