Diese Arbeit wurde vorgelegt am Lehr- und Forschungsgebiet Informatik 9

The present work was submitted to Learning Technologies Research Group

Einstieg in das Hacking für die Sekundarstufe II – Erweiterung des HackIt!-Frameworks von Florian Kerber

First steps into hacking for seniors – an extension of Florian Kerber's HackIt!-framework

Bachelorarbeit

Bachelor-Thesis

von / presented by

Marnitz, Marcel

313998

Prof. Dr. Ulrik Schroeder

Prof. Dr. Horst Lichter

Aachen, 09. Juni 2015

Inhaltsverzeichnis

1. Einleitung **2**
1.1. Motivation . . . 2
1.2. Ausgangssituation . . . 3
1.2.1. Übersicht der Komponenten . . . 3
1.2.2. Schülersicht . . . 4
1.2.3. Lehrersicht . . . 4
1.3. Bezug zum Informatikunterricht . . . 6
1.4. Ziel . . . 8
1.5. Struktur . . . 8

2. Implementierung **9**
2.1. Probleme der ursprünglichen Version . . . 9
2.2. Weiterentwicklung des Grundsystems . . . 9
2.3. Neues Design . . . 10
2.3.1. Globale Veränderungen . . . 10
2.3.2. Anmelden & Registrieren . . . 11
2.3.3. Challengeübersicht für Schüler . . . 13
2.3.4. Ansicht der einzelnen Challenges . . . 14
2.3.5. Verwaltungsansichten . . . 16
2.4. Neue Funktionen . . . 16
2.4.1. Punktesystem . . . 16
2.4.2. Hinweissystem . . . 16
2.4.3. Pre- und Poststories . . . 17
2.4.4. Verwalten der Challenges . . . 18
2.4.5. Benutzerimport . . . 19
2.4.6. Fortschrittsanzeige für Lehrer . . . 19
2.4.7. Statistiken zu Hinweisen . . . 20
2.4.8. Hilfebereich . . . 21
2.5. Sicherheitsmaßnahmen . . . 22
2.5.1. Sicherheit der Challenges . . . 22
2.5.2. Sicherheit der Web-Anwendung . . . 22
2.6. Verbesserung der Datenbank . . . 23

3. Challenges **24**
3.1. Übersicht aller Challenges . . . 24
3.1.1. Nötige Vorkenntnisse . . . 25
3.1.2. Basis-Challenges . . . 25
3.1.3. Hash-Challenges . . . 26
3.1.4. HTTP-Challenges . . . 27

3.1.5. Code Execution 28
3.1.6. Parameter Injection-Challenge 29
3.1.7. User Agent-Challenge 29
3.1.8. SQL Injection 30
3.2. Erweiterungsoptionen 32
3.3. Einsatz im Unterricht 33

4. Fazit **34**
4.1. Zusammenfassung 34
4.2. Ausblick 34

Literaturverzeichnis **36**

Anhänge **38**

A. Übersicht der Challengesets **38**
A.1. Basic challenges 38
A.2. Advanced challenges 39

B. Anleitung zum Bereitstellen der Software (Lehrer) **40**

C. Anleitung zur Benutzung der Software (Schüler) **49**

D. DVD **56**

Vorbemerkung

In dieser Arbeit wird aus stilistischen Gründen das generische Maskulinum verwendet, jedoch sind jeweils Lernende und Lehrende beider Geschlechter gemeint.
An einigen Stellen wird auf den Lehrplan des Faches Informatik in der Sekundarstufe II für das Bundesland Nordrhein-Westfalen verwiesen. Falls nicht anders angegeben, kann der Lehrplan anderer Bundesländer daher abweichen.

Kapitel 1 Einleitung

1.1. Motivation

Die Sicherheit bei Onlinediensten, wie beispielsweise sozialen Netzwerken, hat in den letzten Jahren stark an Bedeutung gewonnen, da diese von immer mehr Menschen genutzt werden. Somit sind Onlinedienste für Angreifer interessant geworden, da einige dieser Onlinedienste sehr sensible Daten speichern, wie beispielsweise beim Onlinebanking oder bei Diensten, die den Alltag erleichtern. Daher ist es wichtig, dass diese Onlinedienste entsprechende Sicherheitsmechanismen integriert haben, sodass die Daten vor dem Zugriff Dritter geschützt sind.

Sicherheit bei Online-Diensten ist wichtig

Sicherheit ist allerdings nicht ausschließlich für Onlinedienste wichtig, denn in Zukunft werden dank Trends wie Smart Home[1] oder Internet of Things[2] immer mehr Geräte mit dem Internet verbunden. Diese Geräte stellen potentielle Angriffsziele für Hackern dar, weshalb die Hersteller entsprechender Geräte in der Pflicht sind, diese vor Hackerangriffen zu schützen.

Leider sind in den letzten Jahren aufgrund von Sicherheitslücken in Systemen immer mehr Angriffe durchgeführt worden, welche oftmals dazu genutzt wurden, Benutzerdaten wie beispielsweise E-Mail-Adressen, Passwörter oder Kreditkarteninformationen zu stehlen. Beispiele hierfür sind zum einen der Hackerangriff auf das Sony Playstation Network im Jahr 2009 [17] sowie ein Angriff auf die Adobe-Benutzerdatenbank im Jahr 2013 [5]. Bei diesen beiden Angriffen wurden sensible Benutzerdaten wie Passwörter oder Kreditkarteninformationen entwendet und später im Internet veröffentlicht.

Sicherheit bei kleinen Projekten darf nicht unterschätzt werden

Selbstverständlich sind nicht nur große Unternehmen potentielle Angriffsziele für Hacker. Viele Schüler lernen während ihrer Freizeit zu programmieren. Häufig wird dabei die Skriptsprache PHP benutzt. PHP ist einfach zu lernen und wird genutzt, um dynamische Websites zu generieren. Beim Lernen dieser oder auch anderer Sprachen zur Generierung von Websites wird jedoch häufig der Sicherheitsaspekt

[1] Smart Home beschreibt das Vernetzen von Haushaltsgeräten wie zum Beispiel Lampen, Jalousien oder Heizung. Damit ist es möglich, diese beispielsweise mittels Smartphone zu steuern. Dies ist meist auch von unterwegs möglich. Ziele von Smart Home sind unter anderem das Steigern der Lebensqualität sowie das Einsparen von Energie. [16]

[2] Unter „Internet of Things“versteht man die Tatsache, dass kleine elektronische Geräte immer mehr unseren Alltag bestimmen, ohne dabei jedoch abzulenken oder aufzufallen. Diese werden untereinader vernetzt und sind meist mit dem Internet verbunden. [12]

nur geringfügig beachtet. Dies ist für kleine Projekte auch nicht immer wichtig. Da vielen Schulen nur wenig Geld zur Verfügung steht, werden Schulhomepages oftmals von der Schule selbst erstellt. Dabei ist der Sicherheitsaspekt allerdings nicht zu vernachlässigen.
Mit der Software, welche im Rahmen dieser Arbeit weiterentwickelt wurde, sollen die Schüler besser für das Thema Sicherheit bei Webanwendungen sensibilisiert werden, indem sie selbst zu kleinen Hackern werden und Sicherheitsmechanismen umgehen.

1.2. Ausgangssituation

Diese Arbeit stellt eine Weiterentwicklung des HackIt!-Frameworks dar, welches Florian Kerber im Rahmen seiner Bachelorarbeit vom 18. September 2012 entwickelt hat. Das HackIt!-Framework ist eine Software, welche den Benutzer in die Position eines Hackers versetzt, welcher kleine Hackits lösen muss.
Hackits sind kleine, meist Web-basierte Challenges, bei denen Sicherheitslücken in einer Software ausgenutzt werden müssen, um diese zu bestehen. Die Challenges, welche als Teil dieser Arbeit aufbereitet wurden, befassen sind dabei mit JavaScript, Problemen mit Hashes, Webserver-Fehlkonfigurationen und SQL Injections, wobei die meisten Challenges bereits während eines Praktikums[3] in der Vergangenheit entstanden und innerhalb dieser Arbeit aufbereitet wurden.
Das HackIt!-Framework richtet sich besonders an Informatik-Schüler der Sekundarstufe II, welche sich über den Stoff des Informatikunterricht hinaus in die Thematik der Online-Sicherheit einarbeiten möchten.

1.2.1. Übersicht der Komponenten

Das HackIt!-Framework besteht aus zwei Komponenten: einer Webanwendung und einem Server-Backend. Das Server-Backend besteht dabei aus einem Debian-Server, welcher wiederum mehrere virtuelle Servern startet, auf denen schließlich die Web-Anwendung läuft.
Das Server-Backend wird als virtuelle Maschine ausgeliefert, welche auf einem Computer gestartet werden muss. Dieser Server beinhaltet aus Sicherheitsgründen weitere virtuelle Maschinen, auf denen die Web-Anwendung einschließlich der Hackits gespeichert sind und ausgeführt werden. Diese virtuellen Maschinen sind bereits fertig konfiguriert und einsatzbereit. Nach außen verhält sich das System so, als wäre es ein einziger Server auf dem die Web-Anwendung und die Hackits ausgeführt werden.
Die Web-Anwendung ist die eigentliche Anwendung, welche im Verlauf dieser Arbeit als „Software“ bezeichnet wird. Diese erlaubt das Lösen von Hackits, welche Teil dieser Web-Anwendung sind.

[3] Praktikum „eLearning Engineering project lab“ (Sommersemester 2013)

1.2.2. Schülersicht

Der Schüler wird in die Lage eines Hackers versetzt. Nach der Anmeldung im System sieht er eine Übersicht an Challenges und kann diese lösen (siehe Abbildung 1.1). Das Lösen einer Challenge besteht in der Regel daraus, sich Zugang zu einem gesicherten Bereich zu verschaffen.

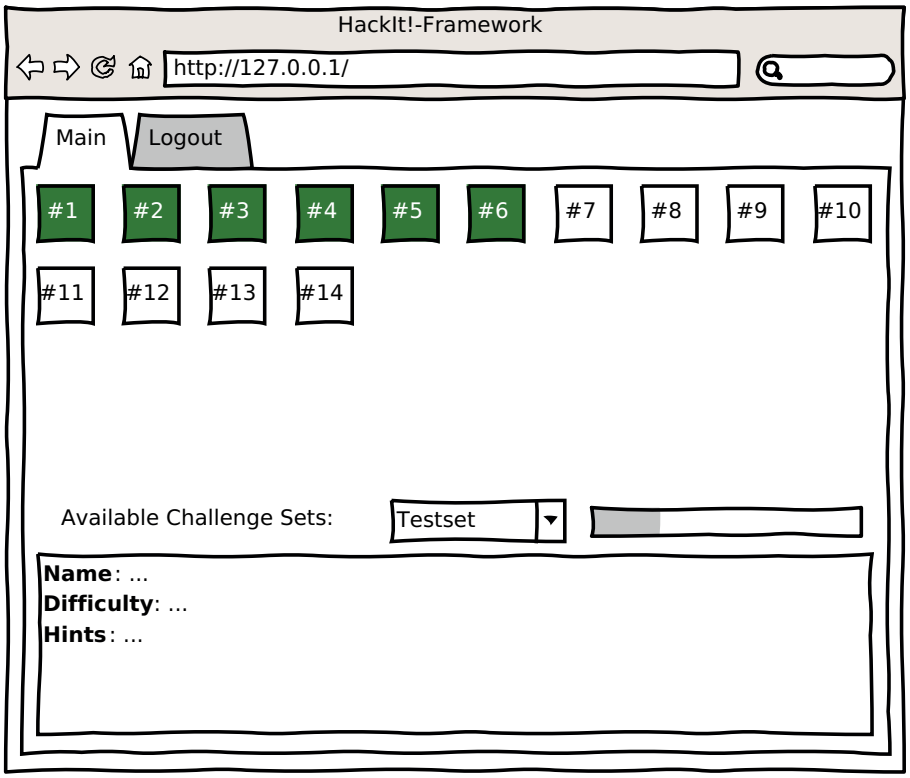

Abbildung 1.1.: Alte Übersicht der Challenges [2]

Vor einer Challenge wird dem Schüler eine Story angezeigt, welche der Motivation dienen soll. Bisher sind diese Stories thematisch unabhängig von der eigentlichen Challenge.
Während ein Schüler eine Challenge zu lösen versucht, hat er die Möglichkeit, Hinweise einzufordern. Allerdings stehen bisher nur für wenige Challenges Hinweise zur Verfügung.

1.2.3. Lehrersicht

Der betreuende Lehrer hat neben der Möglichkeit, wie Schüler Challenges zu lösen, weitere administrative Werkzeuge zur Verfügung.
Damit Schüler Challenges lösen können, müssen sie Mitglied einer Benutzergruppe sein, welcher Challengesets zugeordnet werden können (siehe Abbildung 1.2).

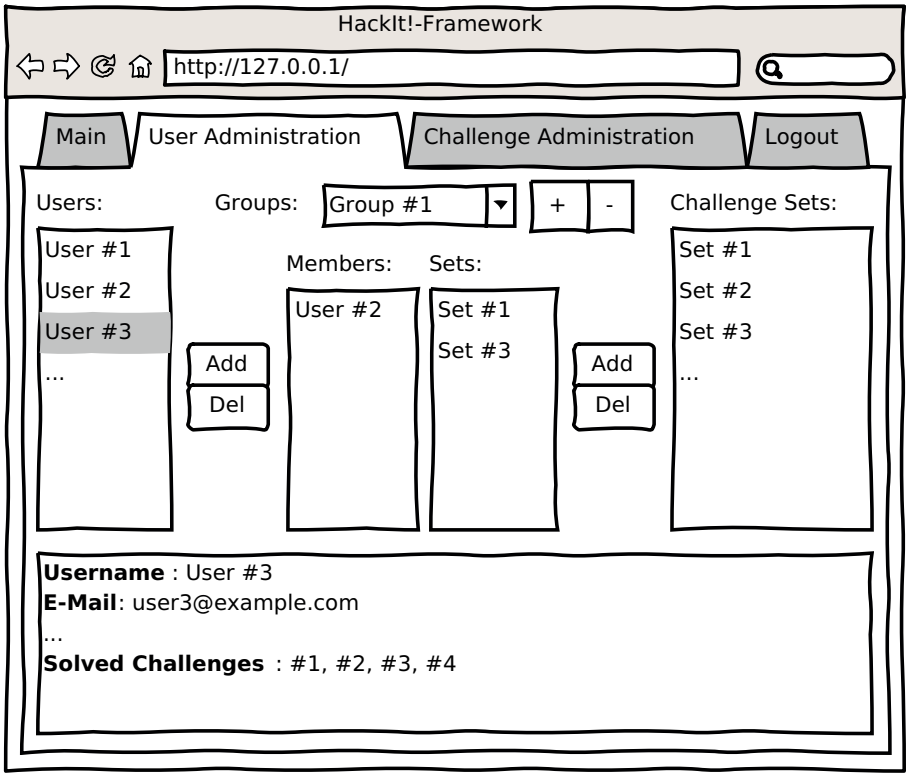

Abbildung 1.2.: Oberfläche zum Administrieren von Benutzern, Benutzergruppen sowie deren Zuordnung zu Challengesets [2]

Challengesets bestehen aus einer oder mehreren Challenges. Die Reihenfolge, in der die Challenges dem Schüler präsentiert werden, kann dabei innerhalb eines Challengesets festgelegt werden. Die bisherige Oberfläche zur Administration ist in Abbildung 1.3 skizziert.

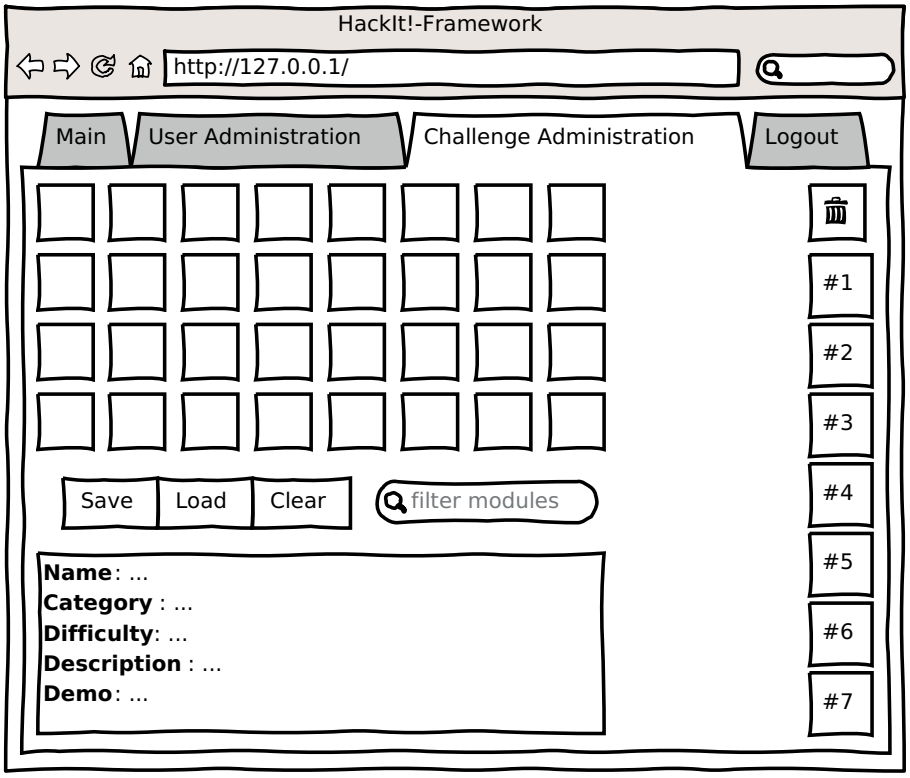

Abbildung 1.3.: Oberfläche zum Administrieren der Challengesets [2]

1.3. Bezug zum Informatikunterricht

Die Inhalte des Informatikunterrichtes im Lehrplan der Sekundarstufe II werden in fünf Kompetenzbereiche und fünf Inhaltsfelder unterteilt (siehe Abbildung 1.4). Kompetenzbereiche beschreiben dabei die fachbezogenen Fähigkeiten und Fertigkeiten, welche die Schüler im Rahmen des Unterrichtes erlernen sollen. Die Inhalte, an denen die Kompetenzen entwickelt werden sollen, lassen sich in Inhaltsfelder einordnen.

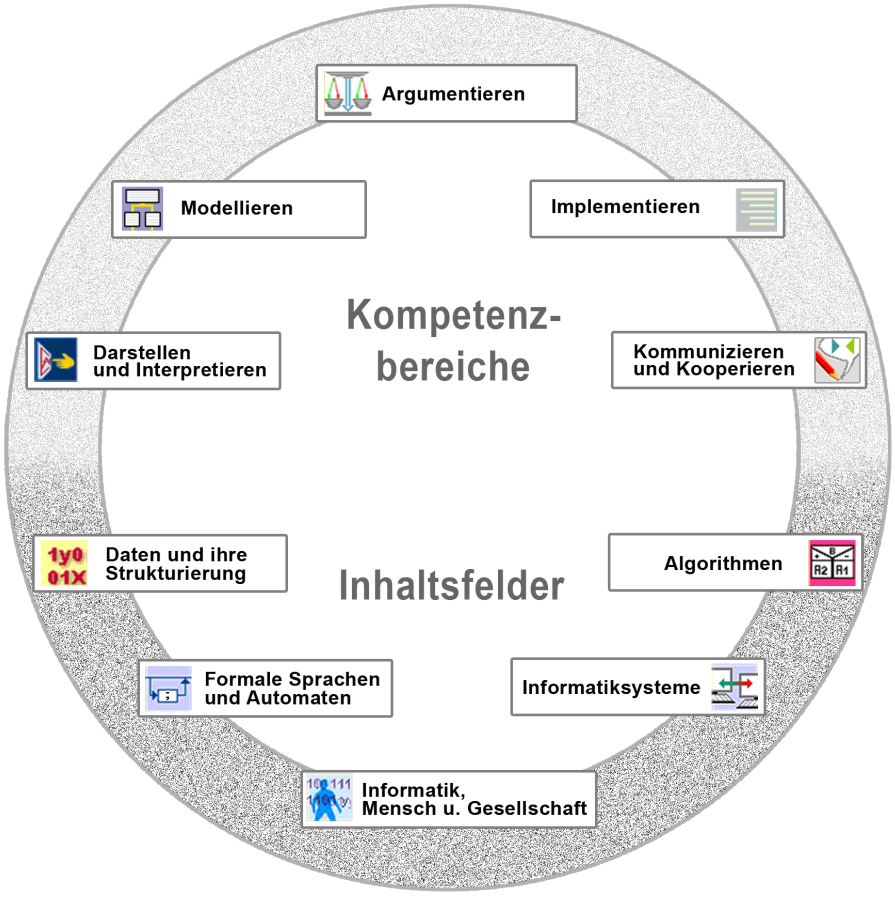

Abbildung 1.4.: Kompetenzbereiche und Inhaltsfelder im Fach Informatik in der Sekundarstufe II [3]

Die Kompetenzbereiche für das Fach Informatik sind „Argumentieren“, „Modellieren“, „Implementieren“, „Darstellen und Interpretieren“ sowie „Kommunizieren und Kooperieren“. Die Inhaltsfelder sind aufgeteilt in „Daten und ihre Strukturierung“, „Algorithmen“, „Formale Sprachen und Automaten“, „Informatiksysteme“ sowie „Informatik, Mensch und Gesellschaft“.

Einordnen des Themas Hacken in Kompetenzbereiche und Inhaltsfelder

Beim Hacken beschäftigt man sich mit Sicherheitsmechanismen, indem man versucht, Schwachstellen in diesen Mechanismen zu finden. Häufig werden diese ausgenutzt, um einen Angriff auf ein System durchzuführen. Daher lässt sich das Thema Hacken in das Inhaltsfeld „Informatiksysteme“ einordnen. Da Onlinedienste von Millionen Menschen täglich genutzt werden, kann man Hacken darüber hinaus dem Inhaltsfeld „Informatik, Mensch und Gesellschaft“ zuordnen. Um einige Sicherheitsschwachstellen zu finden, muss man Algorithmen studieren, da sich auch dort die vermeintlichen Lücken befinden können. Somit lässt sich das Thema Hacken auch mit dem Inhaltsfeld „Algorithmen“ verbinden.

Beim Versuch, aktiv eine Sicherheitslücke auszunutzen, bedarf es manchmal auch kleiner Programme, welche vom Hacker entworfen und implementiert werden müssen. Aus diesem Grund lässt sich Hacken in der Kompetenzfeld „Implementieren“ einordnen.

1.4. Ziel

Diese Arbeit verfolgt zwei große Ziele. Zum einen soll das HackIt!-Framework technisch weiterentwickelt und um zusätzliche Funktionen ergänzt werden und zum anderen sollen die verfügbaren Challenges für Schüler aufbereitet werden.
Bei der Entwicklung stehen die Zielgruppen Schüler und Lehrer im Vordergrund. Die Software soll einfach und intuitiv zu bedienen sein und darüber hinaus ein einheitliches Erscheinungsbild erhalten. Für Lehrer sollen nützliche Funktionen entstehen, welche die Verwaltung des Systems erleichtern und Möglichkeiten zur Fortschrittskontrolle bieten.
Inhaltlich sollen alle Challenges aufbereitet werden, sodass es möglich ist, das HackIt!-Framework im Rahmen einer Unterrichtseinheit zu verwenden.

1.5. Struktur

Diese Bachelorarbeit beinhaltet vier Kapitel, wobei dieses Kapitel das erste darstellt. In Kapitel 2 wird die Weiterentwicklung der Software vorgestellt, wobei dabei zunächst auf technische Details der Software eingegangen wird. Anschließend werden die neue Benutzeroberfläche sowie neue Funktionen vorgestellt. Kapitel 3 beinhaltet eine inhaltliche Beschreibung der Challenges, welche im Rahmen dieser Arbeit aufbereitet wurden. Abgeschlossen wird diese Arbeit in Kapitel 4 mit einem Fazit und einem Ausblick.

Kapitel 2 Implementierung

2.1. Probleme der ursprünglichen Version

Die ursprüngliche Version des HackIt!-Frameworks war als AJAX-Anwendung implementiert, was bedeutet, dass alle Aktionen und Anfragen im Hintergrund als `XMLHttpRequest` ausgeführt wurden und die Seite danach entsprechend der Antwort aktualisiert wurde. Diese Aktualisierung wurde mit Hilfe des JavaScript-Frameworks jQuery durchgeführt.

Benutzerfreundlichkeit nicht optimal und Code schlecht wartbar

Diese Implementierung brachte zwei große Probleme mit sich. Aus Benutzersicht wurde bei jedem versehentlichen Drücken der Taste `F5` („Neu laden“) die Seite neu geladen und man wurde in Folge dessen ausgeloggt. Die Probleme aus technischer Sicht waren allerdings bedeutender, denn da das Aussehen der Seite dynamisch mittels JavaScript geändert wurde, befand sich der HTML-Code für alle Ansichten in einer großen HTML-Datei. Die dazugehörige JavaScript-Datei war aufgrund der Tatsache, dass sie entsprechende Hintergrund-Anfragen starten und auf diese entsprechend reagieren musste, ebenfalls sehr groß. Als Endpunkt dieser Hintergrund-Anfragen diente ein PHP-Skript, welches ebenfalls sehr groß und unübersichtlich wurde, da sich die Funktionalität der gesamten Software in einer Datei befand. Diese großen Dateien machten die Software schlecht wart- und erweiterbar.

2.2. Weiterentwicklung des Grundsystems

Die Software nutzt wie bereits zuvor PHP zum Generieren von HTML5 Seiten sowie MySQL als Datenbank-Backend.

Die Software wurde von Grund auf neu geschrieben und basiert nun auf einem kleinen Kern von Klassen, welche verschiedene Funktionalitäten liefern, wie Session-Management, Datenbank-Anbindung oder Generierung von HTML-Seiten[1]. Dieser neue Kern erlaubt das schnelle Erweitern des Systems um neue Features.

OOP und MVC-Pattern verbessen Code-Qualität

Die neuen Komponenten wurden nach dem Konzept der Objektorientierung modelliert und implementiert. Insbesondere verfolgt das System das „Model View Controller“ Pattern, kurz MVC. Hierbei werden Anwendungscode (Controller) und Code, der für die Anzeige zuständig ist (View), voneinander getrennt [14]. Diese Trennung macht den

[1] Die Dokumentation der Klassen befindet sich im Quellcode in Form von phpDocumentor-Kommentaren.

Code wartbarer und steigert die Wiederverwendbarkeit von Code. Außerdem lässt sich der Code nun einfacher erweitern.
Hier wird Smarty als sogenannte Template-Engine verwendet. Smarty ist eine sehr mächtige Bibliothek zum Erzeugen von HTML-Dateien und bietet eine Fülle an Funktionen, sodass PHP-Code und HTML-Code sauber voneinander getrennt werden können [6].
Aufgrund der Implementierung des neuen Systems wurde die Anwendung nicht als AJAX-Anwendung weiterentwickelt. Dies steigert insbesondere die Benutzerfreundlichkeit des Systems (siehe Abschnitt 2.1).
Darüber hinaus wird das Aussehen der Seite wie in der Web-Entwicklung üblich mittels Cascading Style Sheets (CSS) vorgenommen. Die externen CSS-Dateien sorgen dafür, dass Inhalt und Aussehen der Seiten getrennt werden. Das aktuelle System nutzt nun das CSS-Framework Bootstrap als Grundlage, welches nützliche Komponenten wie beispielsweise Menüleisten oder Formular-Elemente liefert. Diese sind bereits für gängige Browser getestet worden und sehen einheitlich aus. Als Nebeneffekt sind die Seiten „responsive“, was bedeutet, dass sie sich automatisch und sinnvoll der Fenstergröße des Browsers anpassen.
An vielen Stellen im System kann die Lehrkraft einfach formatierte Texte hinterlegen. Dazu wird die Markdown-Syntax verwendet, welche unter anderem bei GitHub zum Einsatz kommt. Diese Syntax ist intuitiver und lässt sich schneller schreiben als Alternativen wie BBCode [13]. Das vorige System erlaubte dem Benutzer, beliebigen HTML-Quellcode zu hinterlegen. Dies hat jedoch den Nachteil, dass die Ausgabe stilistisch nicht zum restlichen Teil der Seite passt, da sich die Formatierung grundlegend unterscheidet.

2.3. Neues Design

Die auf den ersten Blick größte Veränderung betrifft das äußere Erscheinungsbild der Software, welche in diesem Kapitel thematisiert wird.

2.3.1. Globale Veränderungen

Menüs statt Tabs sorgen für Ordnung

Bisher nutzte die Software ein Tab-Control zum Trennen der einzelnen Ansichten wie Challegeübersicht und Administrationsbereiche zu trennen (siehe Abbildung 1.3). Diese Lösung war nicht mehr praktikabel, da einige Bereiche hinzugekommen sind und der horizontale Platz nicht mehr ausgereicht hat. Die neue Oberfläche hat daher im oberen Bereich eine Menüleiste erhalten, welche ähnlich zu den bisher vorhandenen Tabs ist (siehe Abbildung 2.1). Der Vorteil dieses Menüs ist, dass mehrere Menüeinträge thematisch gruppiert und als Dropdownmenü angeboten werden können. Menüpunkte wie Benutzer- und Gruppenverwaltung sowie Menüpunkte zur Verwaltung der Challenges wurden daher jeweils zusammengefasst, da dies zum einen Platz spart und zum anderen für Ordnung sorgt (siehe Abbildung 2.2).

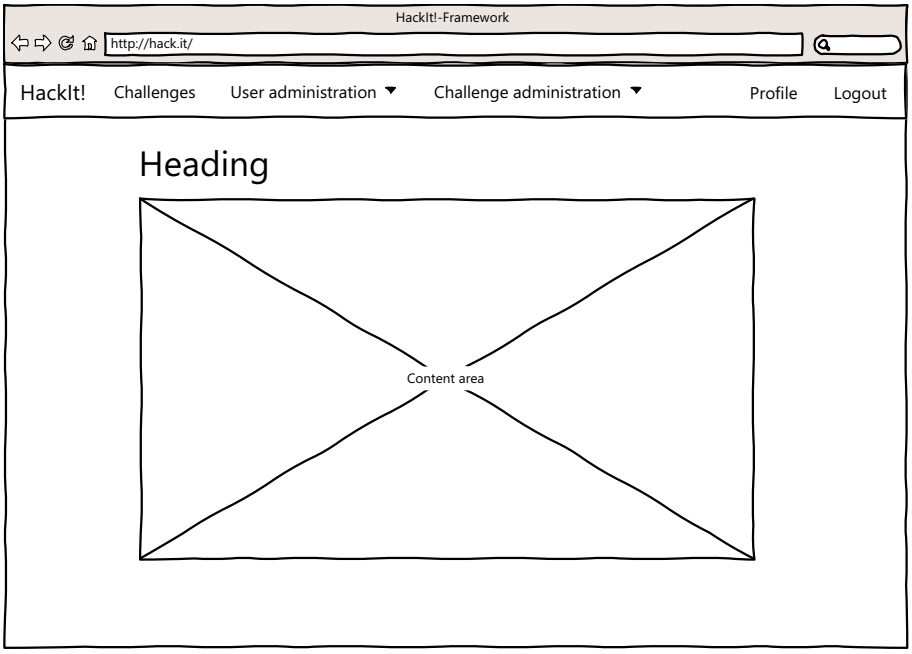

Abbildung 2.1.: Vorlage für alle Ansichten inklusive Menüleiste

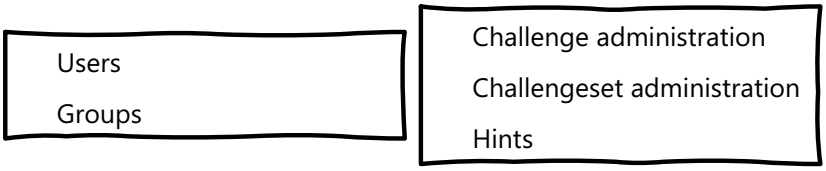

Abbildung 2.2.: Links das Untermenü „User administration“ und rechts das Untermenü „Challenge administration“

2.3.2. Anmelden & Registrieren

Die bisherigen Ansichten für das Anmelden und Registrieren wurden vollständig neu gestaltet.

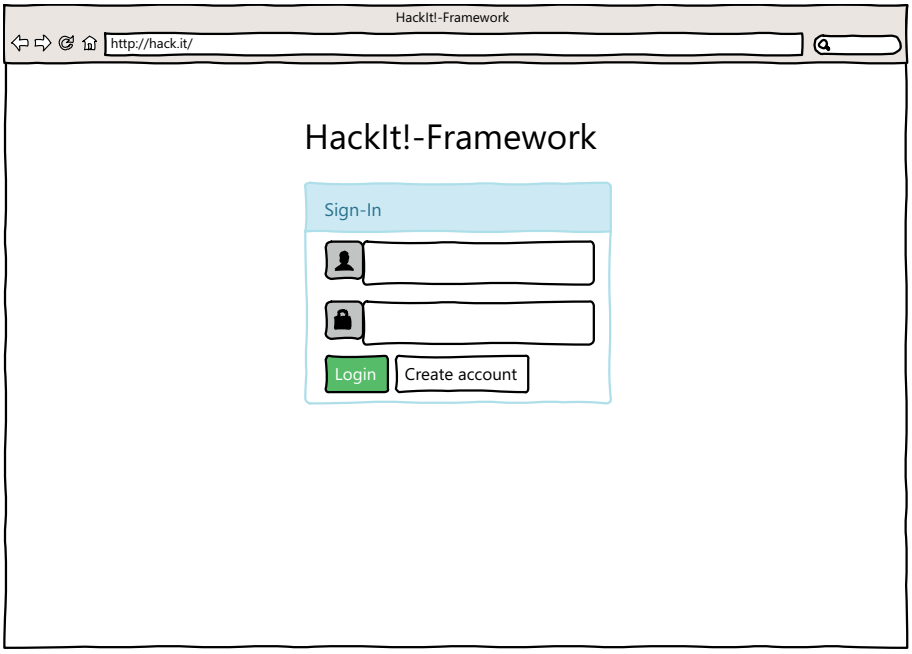

Abbildung 2.3.: Die neue Anmelde-Ansicht

Daher ist die Ansicht zur Registrierung neuer Benutzer nun nicht mehr in mehrere Schritte gegliedert sondern besteht aus einer einzelnen Ansicht, in der alle Daten eingetragen werden können (siehe Abbildung 2.4). Dies ist zum einen dadurch begründet, dass mehrschrittige Formulare aufwändiger zu implementieren sind, und zum anderen dadurch, dass der Benutzer bei einem nicht mehrschrittigen Formular sofort sieht, welche Daten benötigt werden.

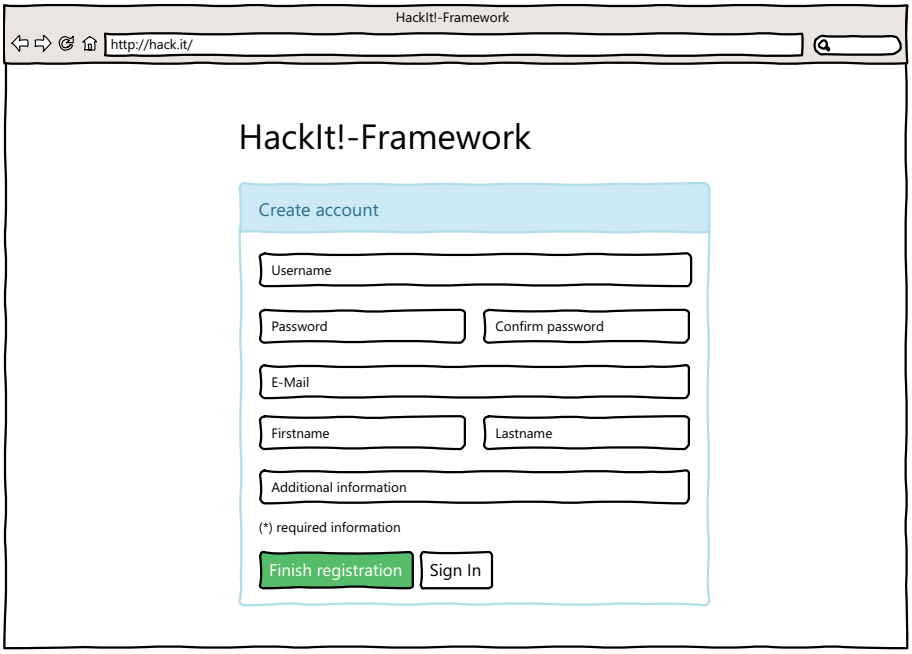

Abbildung 2.4.: Die neue Registrierungsansicht

2.3.3. Challengeübersicht für Schüler

Auch die Übersicht der Challenges wurde in vollem Umfang neu gestaltet. Oberhalb der Challenges befindet sich nun ein Menü, in dem ein Challengeset ausgewählt werden kann, wobei das oberste Challengeset durch das System automatisch ausgewählt wird. Unmittelbar daneben befindet sich eine Checkbox, mit welcher bereits gelöste Challenges ausgeblendet werden können. So sehen Schüler noch schneller, welche Challenges sie noch absolvieren müssen.
Unterhalb des Menüs befindet sich eine Fortschrittsanzeige, welche anzeigt, wie viel Prozent der Challenges im aktuell ausgewählten Challengeset bereits gelöst wurden. Diese Fortschrittsanzeige soll vor allem der Motivation dienen, möglichst viele Challenges zu lösen. Daher wurde diese auch oberhalb der Challenges platziert.
Die Challenges werden hier untereinander in einer Liste angezeigt, statt wie bisher nebeneinander. Informationen zur Challenge befinden sich direkt unterhalb des Challengenamens, statt wie zuvor in einem extra dafür vorgesehenen Fenster, welches nur angezeigt wurde, wenn sich der Mauszeiger über der Challenge befand (siehe dazu Abbildung 1.1). Dieses Verhalten war nicht sehr intuitiv, da das vorgeseheneFenster nicht entsprechend ausgezeichnet wurde. Die Informationen beinhalten die Schwierigkeitsstufe und die erreichte sowie die maximale Punktzahl der Challenge. Ein farbiges Label hinter dem Challengenamen zeigt an, ob die Challenge bereits gelöst wurde (grün) oder nicht (grau).

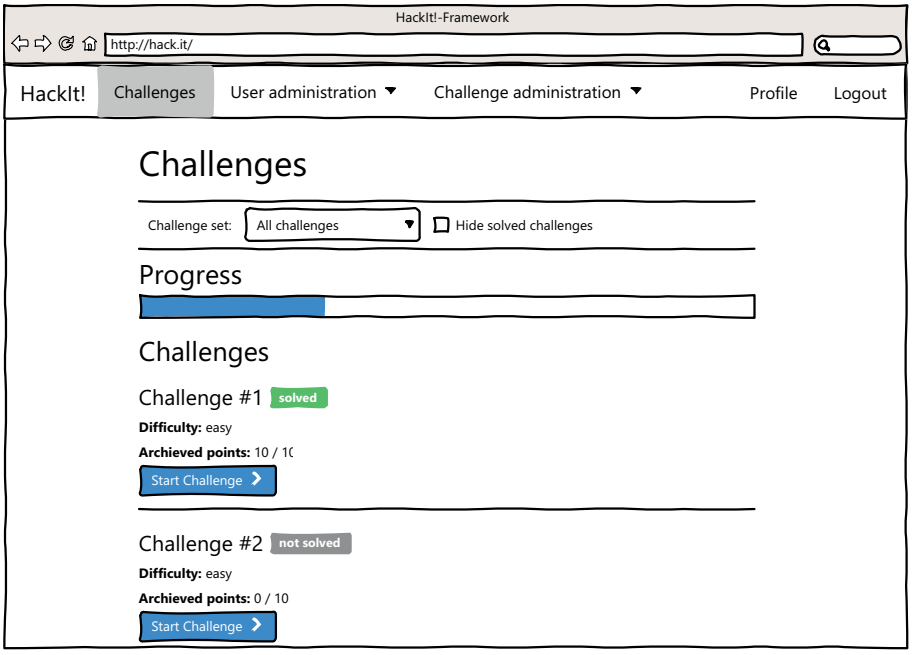

Abbildung 2.5.: Die neue Challengeübersicht

2.3.4. Ansicht der einzelnen Challenges

Bisher war das Erscheinungsbild aller Challenges weder ansprechend noch funktional oder konsistent. Es bestand aus zwei Frames, wobei im oberen Bereich die Hinweise und im unteren Bereich die eigentliche Challenge angezeigt wurde.
Die Verwendung von Frames hatte zwei große Nachteile. Zum einen nahm der obere Frame fast die Hälfte des Bildschirmes ein. Dies war damit begründet, dass die Hinweise dort untereinander angezeigt werden sollten und vertikale Scrollbalken vermieden werden sollten. Zum anderen sah man beim Abrufen des Quelltextes nur den Quelltext der Ansicht, welche die Hinweis- und Challenge-Ansichten eingebunden hat, statt dem Quelltext der eigentlichen Challenge. Zwar konnte man von dort aus zum Quellcode der Challenge navigieren, allerdings ist diese Lösung nicht besonders intuitiv, da gerade Schüler sich schnell durch zusätzliche und sich wiederholende Schritte demotivieren lassen.
Damit die Schüler nicht unnötig demotiviert werden und das System eine konsistente Oberfläche besitzt, wurde die Challengeansicht komplett überarbeitet. Im oberen Bereich wurde eine Menüleiste eingeführt, mit der man Hinweise zur aktuellen Challenge abfragen kann (siehe Abbildung 2.6).

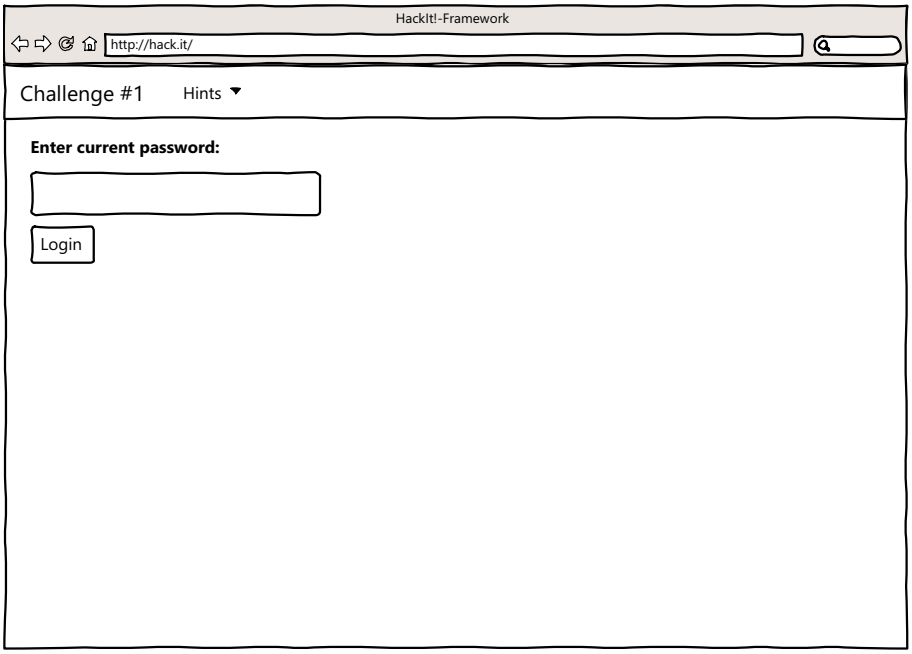

Abbildung 2.6.: Die neue Challenge-Ansicht: oben befindet sich das Menü, wo Hinweise anfragen werden können, darunter befindet sich die Challenge

Das Hinweis-Menü ist dabei ein Dropdown-Menü, mit den Optionen wie in Abbildung 2.7.

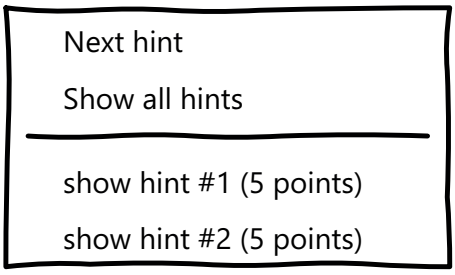

Abbildung 2.7.: Hinweis-Menü

Fragt ein Benutzer nun nach einem Hinweis, so erscheint dieser zwischen Menüleiste und Challenge, wobei der Benutzer die Möglichkeit hat, Hinweise nach und nach über den Menüpunkt „next hint“ anzufragen. Um dabei Transparenz zu gewährleisten, wird bereits vor dem Anfragen angezeigt, wie „teuer“ ein Hinweis ist[2]. Dies ist wichtig, damit der Schüler bereits vorher entscheiden kann, ob er einen Hinweis anfordern möchte. Ist die Option „show all hints“ aktiviert, werden die Hinweise nicht einzeln sondern untereinander angezeigt.

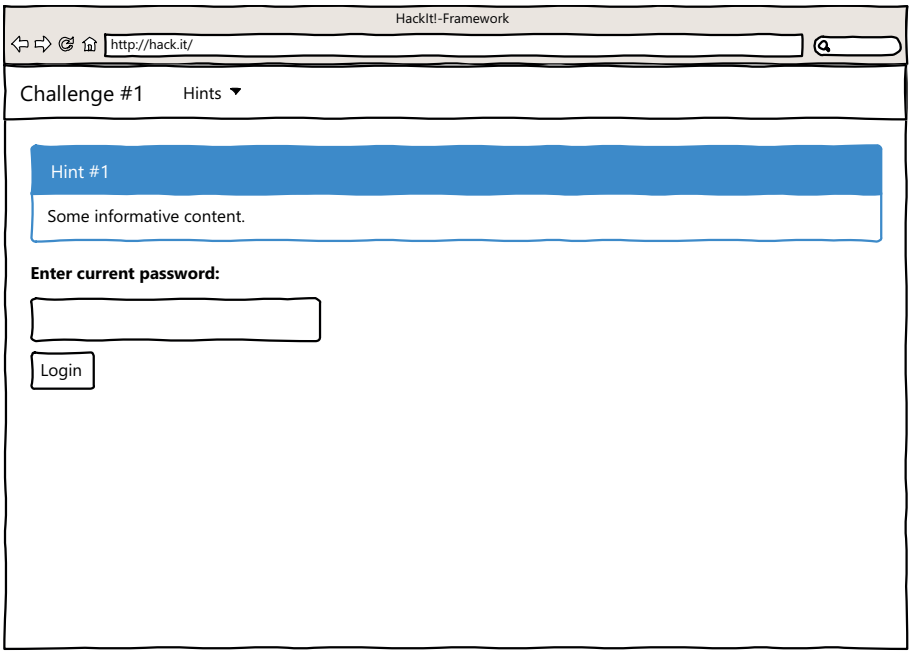

Abbildung 2.8.: Anzeige eines Hinweises

Somit wird die Ansicht nicht mehr aus mehreren Ansichten zusammengesetzt wie bisher. Öffnet man den Quelltext der Ansicht, so sieht man ohne weiterzunavigieren auch den Quelltext der Challenge. Ein Nachteil dieser Methode ist jedoch, dass der Quelltext der Challenge in den Quelltext der Ansicht eingebettet ist. Um den Quelltext der Challenge nun vom restlichen Quelltext hervorzuheben, sind diese Teile des Quelltextes mit Hilfe von Kommentaren umrahmt:

[2] Im Rahmen dieser Arbeit wurde ein Punktesystem eingeführt. Dieses wird in Abschnitt 2.4.1 vorgestellt.

Listing 2.1: Innerhalb dieser Kommentare wird Challenge-spezifischer Code eingebunden

```
1     <!-- CHALLENGE BEGIN -->

      <!-- CHALLENGE END -->
```

2.3.5. Verwaltungsansichten

Alle Ansichten, die der Verwaltung des Systems und der Inhalte dienen, wurden ebenfalls überarbeitet. Die meisten Ansichten bestehen dabei aus einfachen Tabellen für Übersichten und Formularen zum Ändern von Daten wie beispielsweise Benutzerdaten, Gruppenzugehörigkeiten oder Eigenschaften von Challenges. Diese dienen der Übersichtlichkeit der einzelnen Ansichten.

2.4. Neue Funktionen

Alle vorhanden Features wurden übernommen. Dazu zählt das Benutzermanagement, Einteilung der Benutzer in Gruppen sowie das Zusammenfügen von Challenges zu Challengesets (nachzulesen in [2]). Dieses Kapitel widmet sich nun den neu entwickelten Features.

2.4.1. Punktesystem

Damit Schüler einen merkbaren Fortschritt beim Lösen der Challenges sehen, wurde ein Punktesystem implementiert. Für jede Challenge kann eine Punktzahl definiert werden. Löst ein Schüler eine Challenge, so wird ihm die entsprechende Punktezahl gutgeschrieben.

Beim Lösen von Challenges werden dem Schüler Punkte gutgeschrieben

Für die Challenges, welche mit dieser Arbeit ausgeliefert werden, sind bereits Punktzahlen definiert. Diese können nachträglich von der Lehrkraft über die Administrations-Oberfläche geändert werden. Die nachträgliche Änderung der Punktzahl beeinflusst dabei auch die Punktzahl bereits gelöster Challenges, da die Anzahl der erreichten Punkte bei jedem Aufruf neu berechnet werden und nicht in der Datenbank persistiert werden. Das hat den Vorteil, dass Schüler bei einer nachträglichen Änderung der Punktzahl nicht bevorzugt oder benachteiligt werden.
Derzeit kann die erreichbare Punktzahl durch das Abrufen von Hinweisen beeinflusst werden.

2.4.2. Hinweissystem

Damit Schüler beim Lösen der Challenges unterstützt werden, haben sie die Möglichkeit, Hinweise zu erhalten. Das Hinweissystem wurde bereits von einer Praktikumsteilnehmerin[3] implementiert, musste jedoch aufgrund einer unvollständigen und teils fehlerhaften Implementierung überarbeitet werden.

Hinweissystem nun vollständig implementiert

Im Rahmen dieser Arbeit wurden die Hinweise aller Challenges verbessert und ergänzt. Jedoch hat die Lehrkraft die Möglichkeit, Hinweise und deren Reihenfolge über die Administrations-Oberfläche nachträglich abzuändern. Somit kann die Lehrkraft die Software an die Lernenden anpassen. Das Ändern der Reihenfolge kann nun mittels Drag & Drop erfolgen und Hinweise können mit Hilfe der Markdown-Syntax formatiert werden.
Wie bereits bei der Vorstellung des Punktesystems erwähnt, beeinflusst das Abrufen von Hinweisen die erreichte Punktzahl bei Challenges. Dazu können pro Hinweis „Kosten“ definiert werden, welche von der Punktzahl der Challenge subtrahiert werden, falls diese vom Benutzer abgerufen wurden.

2.4.3. Pre- und Poststories

Um die Motivation der Schüler für die einzelnen Challenges zu steigern, wird zu Beginn einer jeden Challenge eine kurze Geschichte angezeigt, in denen die Challenge herausfordernd formuliert wird. Ebenso geben die Geschichten kleine Hinweise über die Challenge, sodass die Schüler einen – wenn auch kleinen – Ansatzpunkt haben.

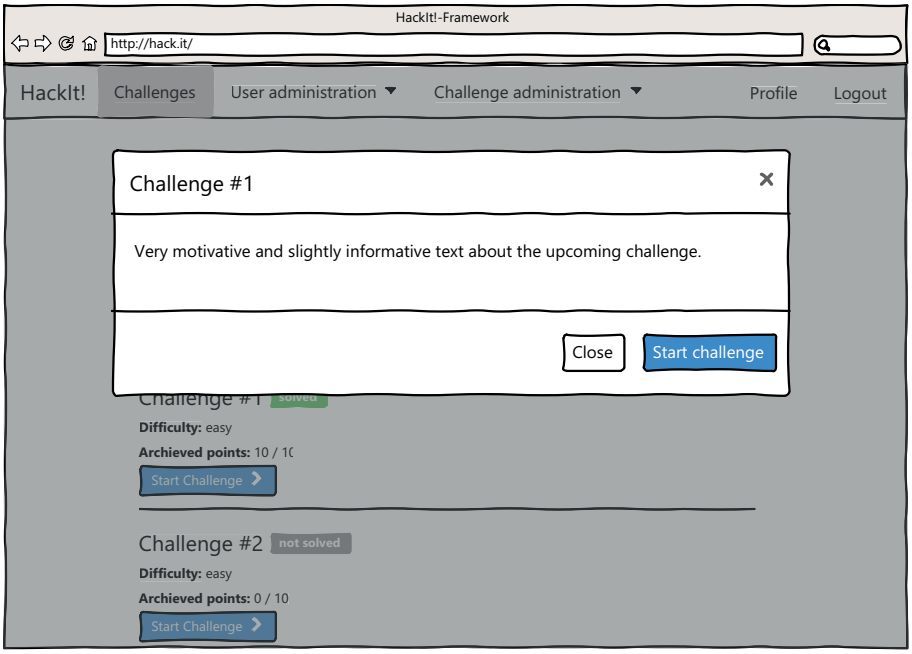

Abbildung 2.9.: Pre-Story vor dem Beginn einer Challenge

Post-Stories bieten dem Schüler weitere Informationen zum Problem der Challenge

Weiterhin wurde die Möglichkeit geschaffen, nach erfolgreichem Lösen einer Challenge, weiteren Text anzeigen zu lassen. Die Idee dahinter ist es, den Schülern weitere Informationen zu dem in der Challenge behandelten Thema zu geben. In vielen der bereitgestellten Challenges gibt es Informationen zu dem Thema sowie Beispiel-Code, der das Problem darlegt. Ebenso werden auf möglichst einfache Weise Lösungen für das in der Challenge vorgestellte Problem präsentiert und falls

[3] Praktikum „eLearning Engineering project lab“ (Sommersemester 2013)

sich Schüler für spezielle Themen interessieren, weiterführende Links bereitgestellt.

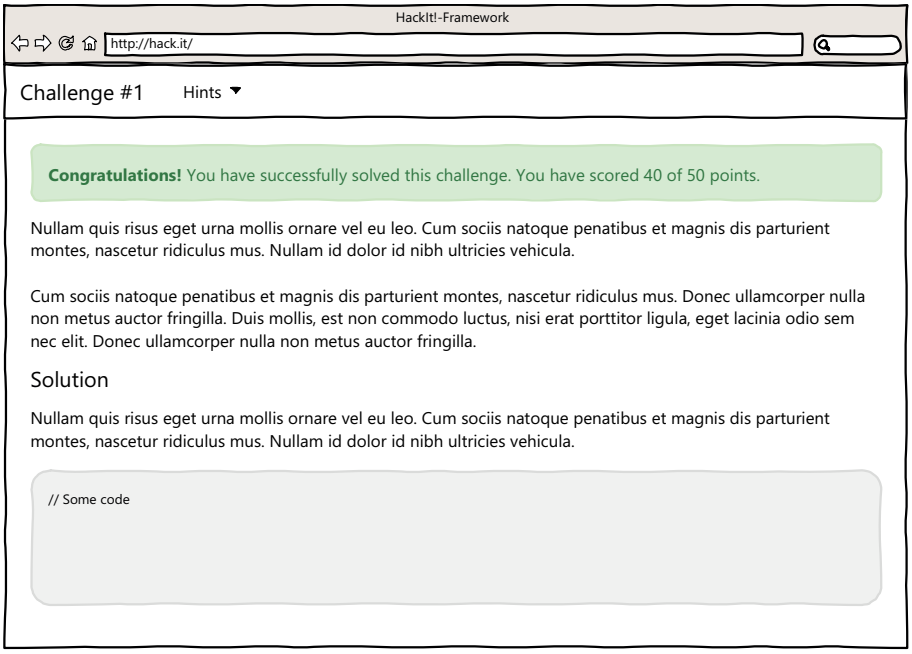

Abbildung 2.10.: Ansicht einer gelösten Challenge: im grünen Bereich wird die erreichte Punktzahl angezeigt und darunter – falls vorhanden – die Post-Story

Intern werden diese Texte als Pre- und Post-Story betitelt und lassen sich nachträglich von der Lehrkraft abändern. Die verfassten Texte unterstützen die bereits vorgestellte Markdown-Syntax, um einfache Formatierungsmöglichkeiten zu bieten. Der Editor selbst hat auch die Möglichkeit, den eingetippten Text in einer Vorschau zu sehen, um mögliche Syntaxfehler aufzuspüren.

2.4.4. Verwalten der Challenges

Bislang konnten angelegte Challenges nicht bearbeitet werden und somit war es nachträglich nicht möglich, Challenges umzubenennen oder den Schwierigkeitsgrad zu ändern. Während der Implementierung des bereits vorgestellten Punktesystems und den Pre- und Post-Stories, wurde jedoch die Möglichkeit geschaffen, auch den Namen oder den Schwierigkeitsgrad von Challenges nachträglich abzuändern.
Die Ansicht zum Verwalten der Challengesets wurde ebenfalls überarbeitet. So können wie bisher auch, Challenges mittels Drag & Drop in ein Challengeset aufgenommen oder daraus entfernt werden. Darüber hinaus kann jedoch nun die Reihenfolge der Challenges durch Drag & Drop geändert werden. Anders als beim vorigen System werden die Challenges dabei mit vollem Namen angezeigt, statt wie bisher nur mit einer eindeutigen ID, und die Reihenfolge der verfügbaren Challenges wird persistiert.

2.4.5. Benutzerimport

Das Anlegen mehrerer Benutzer gleichzeitig kann sehr aufwendig sein, wenn jeder Benutzer einzeln angelegt werden muss. Um der Lehrkraft die Arbeit zu erleichtern, wurde die Möglichkeit eines Benutzerimports geschaffen, womit mehrere Benutzer gleichzeitig im System registriert werden können. Zwar können sich Benutzer auch selbstständig registrieren, müssen allerdings zuerst in eine Gruppe eingefügt werden, bevor ihnen Challenges zur Verfügung stehen.

Anlegen vieler Benutzer gleichzeitig möglich

Für den Benutzerimport muss die Lehrkraft eine CSV-Datei in das System laden, in der er Vorname, Nachname und optional E-Mail-Adresse bereitstellt. Falls gewünscht kann er auch Benutzernamen vorgeben. Andernfalls werden diese vom System nach dem Muster `vorname.nachname` erstellt. Bereits während des Importvorgangs können den neuen Benutzern eine oder mehrere Gruppen zugeteilt werden, sodass der Benutzer nach dem Anmelden direkt über Challenges verfügt. Bevor die Benutzerdaten endgültig angelegt werden, sieht die Lehrkraft eine Übersicht der zu importierenden Daten, sodass diese noch auf Fehler in der CSV-Datei reagieren kann. Falls es bei der Generierung von Benutzernamen zu Kollisionen kommen sollte, hat die Lehrkraft hier die Möglichkeit, neue Benutzernamen zu vergeben. Die Passwörter der neuen Benutzer werden schließlich beim Anlegen vom System generiert und dem Lehrer bereitgestellt, sodass dieser jedem Schüler Benutzername und Passwort aushändigen kann.

2.4.6. Fortschrittsanzeige für Lehrer

Lehrer können Fortschritt der Schüler einsehen

Ein wichtiger Aspekt beim Lehren ist es, den Fortschritt der Lernenden zu überprüfen. Damit die Lehrkraft einen Überblick darüber erhält, wie viele Challenges bereits gelöst wurden, kann sie sich für jedes Challengeset den Fortschritt jeder Benutzergruppe anzeigen lassen. Dort wird zum einen dargelegt, wie viele Schüler eine Challenge bereits gelöst haben. Des weiteren sieht man den Fortschritt jedes einzelnen Schülers, der für das Challengeset freigeschaltet ist. Die Fortschritte werden jeweils durch Balkendiagramme visualisiert (siehe Abbildung 2.11).

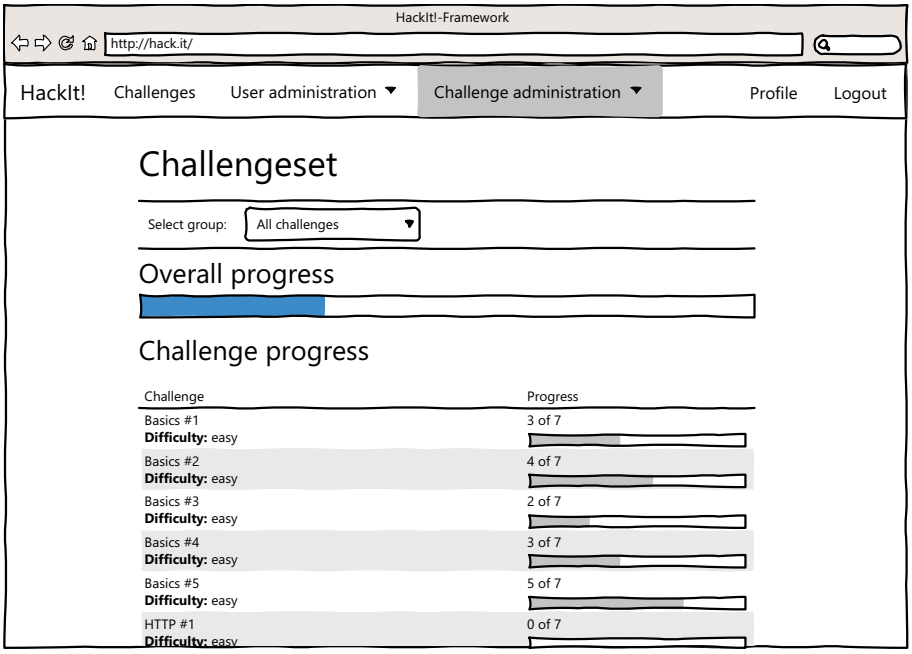

Abbildung 2.11.: Fortschrittsanzeige der Challengesets

2.4.7. Statistiken zu Hinweisen

Lehrer können Abruf-Statistik von Hinweisen einsehen

Darüber hinaus ist es möglich, Statistiken zu den bereitgestellten Hinweisen zu erhalten. Diese zeigen auf, welcher Hinweis von wie vielen Schülern in Anspruch genommen wurde. Dabei wird zwischen Schülern, die die zugehörige Challenge bereits gelöst, und denen, die die Challenge noch nicht gelöst haben, unterschieden. Die Daten werden dabei nicht nur textuell angezeigt, sondern auch in einem Tortendiagramm visualisiert (siehe Abbildung 2.12).
Mit Hilfe dieser Statistik lässt sich unter anderem deuten, wie zielführend die Hinweise für die Schüler waren. Dabei hat der Lehrer die Möglichkeit, die Statistik sowohl für eine als auch für alle Benutzergruppen anzeigen zu lassen.

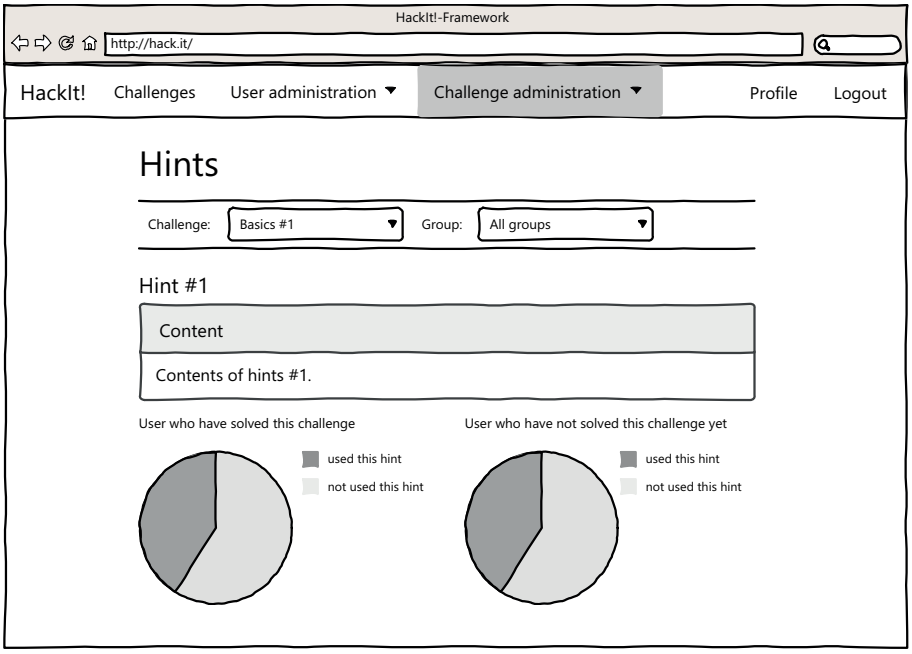

Abbildung 2.12.: Statistik zu Hinweisen inklusive graphischer Auswertung

2.4.8. Hilfebereich

Neben einer Anleitung für Schüler in Form eines PDF-Dokumentes (siehe Appendix C), existiert innerhalb der Software ein Hilfebereich. Aktuell existieren dort zwei Hilfeseiten, wobei diese im Grunde eine Online-Version des PDF-Dokumentes „Student's guide" (siehe Anhang C) sind. Das PDF sowie die Hilfeseiten geben den Schülern einen kurzen Einstieg in das Hacken, bieten nützliche Tipps zu den Browsern Mozilla Firefox und Google Chrome und erklären kurz das Hinweissystem. Im Gegensatz zur PDF-Version deckt die Online-Hilfe nicht nur den Browser Mozilla Firefox, sondern zusätzlich auch den Browser Google Chrome ab. Die Anleitungen zur Benutzung beider Browser sind inhaltlich identisch, lediglich die Screenshots unterscheiden sich. Dies wurde mittels einer JavaScript-Weiche realisiert. Falls jemand die Hilfe mit einem anderen Browser als Mozilla Firefox oder Google Chrome öffnet, erscheint ein Hinweis, dass derzeit offiziell nur Mozilla Firefox und Google Chrome unterstützt werden. Die Verwendung anderer Browser wird nicht empfohlen, da diese entweder nicht getestet werden konnten oder nicht über die benötigtige Funktionalität, wie beispielsweise Entwicklertools oder die Möglichkeit, Plugins zu installieren, verfügen.
Ebenso gibt es zwei Hilfeseiten für Lehrer, wobei sich eine mit der Markdown-Syntax und die andere mit dem Benutzerimport befasst. Erstere Hilfeseite ist bei jedem Eingabefeld, welches Markdown-Syntax unterstützt, verlinkt, sodass die Hilfeseite besonders schnell erreichbar ist.

Aus technischer Sicht bestehen die Hilfeseiten aus HTML-Dateien, welche mit Hilfe der Template-Engine Smarty (siehe Abschnitt 2) gerendert und ausgegeben werden. Somit können auch technisch komplexere Hilfeseiten mit JavaScript erstellt werden. Mit Hilfe von JavaScript ist es beispielsweise möglich, dass Schüler immer für ihren Browser passende Screenshots erhalten.

2.5. Sicherheitsmaßnahmen

Einer Software, welche sich mit der Sicherheit von Web-Anwendungen beschäftigt, sollte selbst ebenfalls ausreichend gegen Angriffe geschützt sein.

2.5.1. Sicherheit der Challenges

Wie bereits in der ursprünglichen Version von Florian Kerber werden die Challenges auf separaten virtuellen Servern ausgeführt, sodass diese abgeschottet voneinander sind. Zwar greifen die Challenges auf die Haupt-Datenbank zu, in welcher sämtliche Daten wie Benutzer, Challenges, Challegesets et cetera gespeichert sind, allerdings können diese nur lesend auf die Datenbank zugreifen, sodass bei einem Angriff keine Daten sabotiert werden können. Weiterhin können Challenges nur auf Dateien und Ordner in ihrem eigenen Verzeichnis und insbesondere nicht auf System-Dateien oder andere wichtige Dateien zugreifen, sodass potentielle Directory Traversal-Angriffe abgewehrt werden. Bei diesen Angriffen wird eine fehlende Überprüfung von Parametern, welche Pfade zu Dateien beinhalten, ausgenutzt[4] [10].

2.5.2. Sicherheit der Web-Anwendung

Die Web-Anwendung selbst ist gegen SQL Injections[5] abgesichert, da bei allen Datenbankanfragen sogenannte „prepared statements“ verwendet werden.

Sicherheit: XSS wird vorgebeugt

Darüber hinaus wird Cross-Site-Scripting-Angriffe, kurz XSS, vorgebeugt. Bei Cross-Site-Scripting handelt es sich um eine Art HTML Injection, das bedeutet es wird fremder HTML-Code in die Seite eingefügt [9]. Dadurch ist unter anderem es möglich, schädlichen JavaScript-Code im Kontext eines Benutzers. Dieser HTML-Code wird durch Benutzereingaben in Web-Anwendungen eingeschleust. Da alle Benutzereingaben von HTML-Code gesäubert werden bevor sie in der Datenbank persistiert werden, wird Cross-Site-Scripting-Angriffen vorgebeugt.

Sicherheit: CSRF wird abgewehrt

Ein weiteres Sicherheitsfeature ist die Abwehr von Cross-Site-Request-Forgeries, kurz CSRF, was eine Angriffsmöglichkeit für Hacker beschreibt. Dabei werden einem potentiellen Opfer Links untergeschoben, welche bei Abruf gewisse Aktionen im Kontext des Opfers ausführen. Mögliche Aktionen können hierbei das Anlegen oder Löschen

[4] Ein Beispiel für einen solchen Angriff wird in der Challenge *Parameter Injection* in Kapitel 3.1.6 vorgestellt

[5] Eine Erklärung zu SQL Injections wird ihm Rahmen der Challenge *SQL Injection* in Kapitel 3.1.8 gegeben

eines Benutzers sein. Um dieses Problem zu beheben, wird bei allen auszuführenden Aktionen, zusätzlich zu den zu übertragenden Informationen, eine weitere, geheime Information übertragen. Diese Information ist ein Token, welches für jeden Benutzer individuell bei der Anmeldung generiert wird. Dieses Token wird bei jeder Aktion überprüft und nur bei Übereinstimmung wird die auszuführende Aktion tatsächlich ausgeführt.

2.6. Verbesserung der Datenbank

Nicht nur das Aussehen der Software wurde im Rahmen der Arbeit verbessert, sondern auch das Datenbank-Backend. Dabei wurde die Tabellen-Struktur übernommen und um Constraints ergänzt. Mit Hilfe von Constraints können Bedingungen an die Tabellen geknüpft werden, welche beim Einfügen, Ändern oder Löschen von Datensätzen erfüllt sein müssen. Damit kann die Integrität der Datenbank sichergestellt werden.

Fremdschlüssel stellen Integrität der Datenbank sicher

Im Rahmen dieser Arbeit wurden überall Fremdschlüssel definiert, wo es möglich war, um entsprechende Datenfelder miteinander zu verknüpfen. Dies stellt sicher, dass beim Löschen von Datensätzen auch über Fremdschlüssel verknüpfte Datensätze entfernt werden. So ist sichergestellt, dass keine Daten in der Datenbank verwaisen.

Darüber hinaus wurde die Standard-Kodierung der Datenbank und aller Tabellen von `latin1` auf `UTF-8` umgestellt. Dies wurde notwendig, da die PHP-Skripte ebenfalls mit `UTF-8` kodiert sind und es anderenfalls zu Anzeigeproblemen bei Umlauten oder anderen Sonderzeichen kommen kann.

Kapitel 3 Challenges

3.1. Übersicht aller Challenges

Die Software beinhaltet derzeit 14 Challenges. Der Schwierigkeitsgrad der Challenges ist unterschiedlich, wobei diese in einfach, mittel und schwer eingeteilt sind. Die schweren Challenges sind insbesondere an Schüler gerichtet, welche bereits etwas Erfahrung mit dem Thema haben oder allgemein sehr fit in Informatik sind.

14 Challenges verfügbar

Folgende Challenges wurden im Rahmen dieser Arbeit aufbereitet oder neu entwickelt:

1. Basics #1 (einfach)
2. Basics #2 (einfach)
3. Basics #3 (einfach)
4. Basics #4 (einfach)
5. Basics #5 (einfach)
6. Code Execution (mittel)
7. Hash #1 (einfach)
8. Hash #2 (schwer)
9. Hash #3 (schwer)
10. HTTP #1 (einfach)
11. HTTP #2 (mittel)
12. Parameter Injection (mittel)
13. SQL Injection (mittel)
14. User Agent (einfach)

3.1.1. Nötige Vorkenntnisse

Die Challenges bestehen meist aus einer Kombination aus HTML- und JavaScript-Quelltext. Damit die Schüler gut damit zurechtfinden, sollten sie sich bereits zuvor mit HTML auseinandergesetzt haben. In Nordrhein-Westfalen wird dies meist im Rahmen des Informatikunterricht in den Jahrgangsstufen 8 oder 9 durchgenommen[1]. Darüber hinaus sollten die Schüler mit den Grundkonzepten der Skriptsprache JavaScript vertraut gemacht worden sein. Die vorausgesetzten Grundkonzepte beschränken sich bei den mitgelieferten Challenges auf Variablendeklaration, `if`-Statements inkl. Vergleichsoperatoren, Funktionen und das Abrufen von Werten von Formularelementen.
Einige Challenges erfordern das Auseinandersetzen mit PHP-Code. Dieser ist gut kommentiert und sehr ähnlich zu JavaScript-Code. Es kann daher seitens der Lehrkraft gegebenenfalls eine kleine Einführung in PHP gegeben werden.
Eine Challenge beschäftigt sich mit der Datenbanksprache Structured Query Language (SQL). Von einer einfachen Einführung in SQL ist aufgrund der Komplexität abzusehen. SQL muss innerhalb des Themengebietes „Datenbanken“ im Rahmen der Abiturvorbereitung behandelt werden [4]. Daher ist es ratsam, erst nach dem Thema Datenbanken das HackIt!-Framework im Unterricht einzusetzen oder gegebenenfalls auf diese Challenge zu verzichten.

3.1.2. Basis-Challenges

Die Challenges *Basic #1* bis *Basic #5* stellen Basis-Challenges dar. Diese sollen als Einstieg in die Thematik des Hackens dienen. In den Challenges Basic #1 bis #3 müssen die Schüler kurzen JavaScript-Code studieren, um ein Passwort herauszufinden. Der entsprechende Code ist dabei nicht immer im Quellcode der Seite enthalten, sondern in einem Fall auch in einer externen Datei.
In Challenge *#4* ist die Passwort-Datei in einem versteckten Formular-Element gespeichert, was die Schüler herausfinden und mit Hilfe der Passwort-Datei lösen sollen. Hier werden die Schüler dazu sensibilisiert, wichtige Daten nicht auf Client-Seite zu speichern, auch wenn diese benötigt werden, um in mehrschrittigen Anfragen Daten zwischenzuspeichern.

Basis-Challenges als Einstieg

Ab Challenge *Basic #5* treten die Schüler erstmals in Kontakt mit den Entwicklertools von Google Chrome oder Mozilla Firefox, da diese auch eine Konsole enthalten, um JavaScript-Code direkt auszuführen. Bei der Challenge müssen die Schüler einen Formularwert, welcher nur mit Hilfe der Entwickler-Tools abgeändert werden kann, vor dem Abschicken ändern. Unabhängig von dem eigentlichen Problem der Challenge sollen die Schüler hier lernen, dass sich Formular-Daten auf dem Client vor dem Abschicken manipulieren lassen. Dies gilt insbesondere für versteckte Formular-Elemente. Ein weiteres Problem dieser Challenge ist die Tatsache, dass „Passwort vergessen“-Funktionen

[1] HTML fällt in den Bereich „Umgang mit Software“ in der Jahrgangsstufe 8 [1]

Passwörter im Klartext versenden. Noch schlimmer dabei ist die Implikation, dass die Passwörter entweder im Klartext oder mittels eines symmetrischen Verschlüsselungsalgorithmus gespeichert werden. Hat ein Angreifer Zugriff auf die Datenbank mit Passwörtern, kann er die Passwörter entweder direkt auslesen oder entschlüsseln. Um dieses Problem zu lösen, sollte nicht das Passwort sondern der Hashwert des Passwortes gespeichert werden. Der Vorteil von Hash-Funktionen ist, dass sich aus der Ausgabe nur sehr schwer die ursprüngliche Eingabe berechnen lässt. In den Hash-Challenges wird die Verwendung von Hash-Funktionen aufgegriffen und weiter verbessert.

3.1.3. Hash-Challenges

Die Challenges *Hash #1* bis *Hash #3* befassen sich mit Hash-Problemen. Idealerweise haben die Schüler bereits Vorkenntnisse über Hashes. Falls dies jedoch nicht der Fall ist, werden sie im Rahmen der Challenge *Hash #1* dazu aufgefordert, sich über den Hash-Algorithmus `MD5` zu informieren.
In der Challenge *Hash #1* werden die Schüler aufgefordert, einen `MD5`-Hash zu „entschlüsseln“. Hierbei steht das Wort Entschlüsseln für das Herausfinden des Eingabewertes des Hash-Algorithmus. Trotz der Tatsache, dass ein Hash-Algorithmus immer eine asymmetrische Verschlüsselung darstellt, ist es unter gewissen Umständen möglich, aus einem Hash-Wert die ursprünglichen Daten zu gewinnen. In dieser Challenge wird dabei der Umstand ausgenutzt, dass es Online-Dienste gibt, welche `MD5`-Hashes aus Benutzereingaben erzeugen. Diese speichern beim Erzeugen sowohl Eingabe des Hashes als auch den berechneten Hash, wodurch eine Rückwärtsabfrage möglich wird.

Salzen von Passwörtern ist wichtig

Da Hashes häufig zum Speichern von Passwörtern benutzt werden, wird nach der Challenge erklärt, wie man dieses Problem umgehen kann: das „Salzen“ von Passwörtern, bevor diese abgespeichert werden. Das bedeutet, dass die Eingabe der Hash-Funktion vor der Berechnung des Hashes verändert wird. Eine solche mögliche Veränderung ist das Voranstellen und Hintenanstellen einer Zeichenkette (siehe Listing 3.1), womit sich der Hash einer Eingabe von dem Hash der gleichen Eingabe ohne Veränderung unterscheidet und eine Entschlüsselung erschwert wird.

Listing 3.1: „Salzen“ von Hashes

```
1    $salt = "somerandombutfixedstring";
     $hash = md5($salt . $password . $salt);
```

Die zweite Hash-Challenge befasst sich mit Kollisionen bei Hash-Funktionen, wobei hier konkret der `MD5`-Algorithmus genutzt wird. Unter Kollision versteht man die Tatsache, dass zwei verschiedene Eingabewerte den gleichen Hash erzeugen. Leider ist `MD5` sehr anfällig für Kollisionen. Mit Hilfe von Grafikkartenprozessoren lassen sich sehr schnell Kollisionen finden. Im Jahr 2009 hat die c’t mit Hilfe einer nVidia GeForce 9800 GX2 Kollisionen innerhalb von 35 Minuten entdeckt

[7]. Seitdem ist die Technik weiter vorangeschritten, sodass davon ausgegangen werden kann, dass Kollisionen mit neuerer Hardware noch schneller gefunden werden können.
Innerhalb der Challenge sollen die Schüler jedoch nicht selbst eine Kollision finden. Stattdessen sollen sie sich zunächst über das Thema im Internet informieren und schließlich versuchen, die im Netz kursierenden Kollisionen anzuwenden.

Wahl der Hash-Funktion ist entscheident

Als Lösung dieses Problems wird vorgeschlagen, eine Hash-Funktion zu verwenden, für die es aktuell keine Möglichkeit gibt, Kollisionen zu begünstigen. Als zwei mögliche Hash-Funktionen werden `SHA-256` oder `SHA-512`[2] vorgeschlagen.
In der dritten Hash-Challenge soll eine sogenannte „length extension attack“ auf `SHA-1` durchgeführt werden. Diese Attacke kann durchgeführt werden, wenn ein Hash nach dem Muster `H(key || data)` betimmt wird, wobei `H` die entsprechende Hashfunktion darstellt, `key` ein geheimer und somit unbekannter Schlüssel ist und `data` die bekannten Daten darstellt, welche gehasht werden sollen. Innerhalb der Challenge sind die Daten bekannt sowie ein Hash, welcher dazu dient, die Echtheit der Daten zu überprüfen. Um Zugriff auf den geschützten Bereich zu erhalten, müssen die Schüler die Daten verändern. Dies führt jedoch dazu, dass mit Hilfe des ursprünglich angegebenen Hashes die Echtheit der Daten nicht mehr gewährleistet ist. Da der Schlüssel `key` unbekannt ist, können die Schüler nicht einfach den entsprechenden Hash berechnen. Mit Hilfe der „length extension attack“ müssen sie daher die Daten manipulieren, um auch ohne Wissen des geheimen Schlüssels das Skript zu überlisten, sodass der vom Benutzer angegebene Hash die Echtheit der Daten bestätigt.
Aufgrund ihrer Komplexität wurden die Challenges #2 und #3 als schwer eingestuft. Diese sind für Schüler gedacht, welche besonders fit in der Thematik sind und bereits alle einfachen und mittleren Challenges abgeschlossen haben.

3.1.4. HTTP-Challenges

In den HTTP-Challenges *HTTP #1* und *HTTP #2* geht es um Authentifizierung und Authentizität über HTTP.
Bei der Challenge *HTTP #1* wird ein Cookie genutzt, um festzulegen ob ein Benutzer angemeldet ist. Dazu wird ein Cookie mit dem Namen `authorized` initial auf `0` gesetzt. Die Schüler werden dann aufgefordert, sich die HTTP Antwort des Servers genau anzuschauen. Anschließend müssen sie versuchen, mit Hilfe geeigneter Tools oder mit Hilfe von JavaScript den Wert des Cookies auf `1` zu setzen.

So wenig Daten wie möglich beim Client speichern

Da HTTP ein `stateless` Protokoll ist, werden Cookies genutzt, um Daten beim Client zu speichern. Da diese Daten bei jeder Anfrage an den Server mitgesendet werden, hat der Server damit die Möglichkeit, einen Client wiederzuerkennen. Beim Speichern von Daten in Cookies müssen Programmierer jedoch beachten, dass diese Daten vom Client

[2] Zum derzeitigen Zeitpunkt (Mai 2015) sind keine Angriffsmöglichkeiten für SHA-256 oder SHA-512 bekannt [11]

manipuliert werden können. Daher wird als Lösung vorgeschlagen, die eigentlichen Daten auf dem Server in einer Datenbank zu speichern und lediglich eine entsprechende ID, auch Session ID genannt, an den Client zu schicken. Wird diese ID manipuliert, findet der Server die ID nicht mehr in der Datenbank und erkennt den Benutzer nicht.
Eine weitere Möglichkeit zur Authentifizierung bieten Webserver wie beispielsweise Apache Webserver oder nginx. Diese wird in Challenge *HTTP #2* thematisiert. Hierbei handelt es sich um die `Basic Auth`-Funktion von Webservern. Dabei wird auf dem Dateisystem eine Datei mit Benutzernamen und Passwörtern gespeichert und dem Webserver entsprechend mitgeteilt, welches Verzeichnis er damit schützen soll. Der Benutzer bekommt bei Anfrage einer Datei in dem gesicherten Verzeichnis eine Aufforderung sich anzumelden. Anschließend überprüft der Webserver die eingegebenen Daten und liefert die Datei entsprechend aus.

Passwort-Dateien außerhalb des Webserver-Verzeichnises speichern

Das Problem bei dieser Authentifizierungsmöglichkeit ist jedoch, dass die Datei mit den Passwörtern oftmals in einem Verzeichnis gespeichert wird, welches von außen über den Webserver erreichbar ist. Ist der Webserver – wie bei der Challenge – falsch konfiguriert, liefert er auf Anfrage diese Passwort-Datei aus und somit kann sich ein Hacker unbemerkt Zugang zu dem gesicherten Verzeichnis verschaffen. Dieses Problem wird am Ende der Challenge aufgegriffen, indem entsprechende Gegenmaßnahmen präsentiert werden, welche darin bestehen, dafür zu sorgen, dass der Webserver diese Dateien nicht ausliefert. Dies kann zum einen durch eine entsprechende Konfiguration sichergestellt werden oder zum anderen dadurch, dass man die Datei in einem Verzeichnis speichert, welches der Webserver nicht ausliefert.

3.1.5. Code Execution

Die Challenge *Code Execution* befasst sich mit der Ausführung von externen UNIX-Skripten. Ein Beispiel dafür könnte das Aktivieren oder Deaktivieren von sogenannten Cron-Jobs[3] über ein PHP-Skript sein. Innerhalb der Challenge haben die Schüler die Möglichkeit, UNIX-Befehle auszuführen. Damit die Sicherheit innerhalb der Challenge gewährleistet ist, werden die Befehle intern überprüft und nur der Befehl `ls` führt zu einem Ergebnis. Mit Hilfe des UNIX-Programms `ls` kann der Inhalt von Verzeichnissen angezeigt werden. Diese Möglichkeit muss genutzt werden, um den Namen der Passwort-Datei herauszufinden. Diese hat einen kryptischen Namen, sodass diese innerhalb der Datei- und Verzeichnisliste auffällt. Da es sich bei dieser Datei um eine Textdatei handelt, kann die Datei über den Webserver abgerufen werden.

Gefahr beim Ausführen externer Programme

Damit dieses Problem in den eigenen Anwendungen nicht ausgenutzt werden kann, sollte man insbesondere bei Web-Anwendungen auf die Ausführung externer UNIX-Skripte verzichten. Außerdem sollte man

[3] Cron-Jobs sind Prozesse, welche vom System automatisch in vom Benutzer definierten Abständen ausgeführt werden [8]

dem Benutzer unter keinen Umständen die Möglichkeit geben, UNIX-Befehle auszuführen.

3.1.6. Parameter Injection-Challenge

Bei der Challenge *Parameter Injection* wird ein sehr schwerwiegendes und häufig ausgenutztes Problem angesprochen. Bei vielen Websites werden Dateien nicht direkt über den Webserver ausgeliefert, sondern passieren zunächst ein Skript, wie beispielsweise ein PHP-Skript. Solche Skripte werden häufig dazu genutzt, um Bilddateien zu verkleinern, bevor sie an den Client ausgeliefert werden.
In der Challenge wird das Skript dazu genutzt, Dokumentinhalte innerhalb der Seite anzuzeigen, wobei dem Skript der Dateinamen der anzuzeigenden Datei übergeben wird. Die Schüler sollen mit Hilfe dieses Skriptes versuchen, die bereits in HTTP #2 kennengelernte Passwort-Datei anzeigen zu lassen, um sich Zugang zum gesicherten Bereich zu verschaffen und damit die Challenge zu lösen.
Damit haben die Schüler bereits eins der kritischen Szenarien dieses Problems durchgespielt. Da das Skript nicht überprüft, ob sich die angeforderte Datei nicht in einem gesicherten Verzeichnis befindet, können so sensible und vermeintlich geschützte Dateien an den Client und somit an potenzielle Hacker ausgeliefert werden. Neben Passwort-Dateien sind auch Konfigurationsdateien besonders zu schützen, da diese meist Zugangsdaten für die Datenbank oder andere sensible Daten enthalten.

Überprüfen von Datei-Parametern wichtig

Um das Problem zu beheben, muss das entsprechende Skript vor dem Ausliefern der Datei überprüfen, ob die angeforderte Datei auch ausgeliefert werden darf. Meist soll das Skript auch nur auf Dateien innerhalb eines bestimmten Verzeichnisses zugreifen, sodass auch sichergestellt werden muss, dass man dieses Verzeichnis nicht verlässt, was zum Beispiel mit dem Aufruf `get.php?file=../config.php` geschehen würde.

3.1.7. User Agent-Challenge

In der Challenge *User-Agent* erfolgt die Authentifizierung über das Ändern des eigenen User-Agent. Mit User-Agent bezeichnet man eine Zeichenfolge, welche Webbrowser identifizieren (siehe Listing 3.2). Diese User-Agents werden automatisch von Browsern bei jeder Anfrage in den Kopfdaten übertragen.

Listing 3.2: Dieser User-Agent beschreibt die x64-Variante des Browsers Chrome in der Version 42.0.2311.135 auf dem Betriebssystem Windows 8.1 x64

```
1    Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit
        /537.36 (KHTML, like Gecko) Chrome/42.0.2311.135
        Safari/537.36
```

Die Aufgabe der Schüler ist es nun, mit Hilfe von Plugins den User-Agent des Browsers zu verändern. Der gewünschte User-Agent wird innerhalb der Challenge bekannt.

Glücklicherweise wird diese Art der Authentifizierung sehr selten bis gar nicht verwendet, allerdings kann es dennoch manchmal praktisch sein, den eigenen User-Agent zu ändern, da viele Seiten abhängig vom User-Agent verschiedene Inhalte anbieten. Durch das Ändern kann man auch ohne den Wechsel des Browsers an entsprechende Inhalte gelangen. Ähnlich arbeiten einige Seiten, welche für Inhalte Geld verlangen. Damit ihre Inhalte dennoch in den Suchergebnissen von Suchmaschinen erscheinen, bedienen sie sich eines kleinen Tricks, denn ruft man die Seite beispielsweise als „Google Bot“[4] auf, wird der Inhalt ganz normal angezeigt. Ruft man die Seite jedoch mit einem Webbrowser auf, wird man meist aufgefordert sich erst anzumelden, bevor man den Inhalt abrufen kann.

3.1.8. SQL Injection

Eine heutzutage sehr häufig angewendete Attacke sind sogenannte SQL Injections. Dabei wird ausgenutzt, dass bei Datenbankabfragen häufig auch Benutzereingaben benutzt werden, beispielsweise ein Suchbegriff bei einer Suche oder der Benutzername bei einer Anmeldung. Werden diese Eingaben nicht vor dem Ausführen der Datenbankabfrage überprüft, kann ein Angreifer beliebigen SQL Code in die Datenbankabfrage schmuggeln.

Beispiel
Auf einer Seite soll eine Suchfunktion implementiert werden. Ein entsprechender Datenbankabfrage könnte wie folgt lauten:

Listing 3.3: Beispielhafter SQL-Query für eine Suche

```
1   SELECT * FROM `articles` WHERE `content` LIKE '%
      $search%'
```

Dabei ist in der Variable `$search` der Suchbegriff gespeichert, welcher der Benutzer eingegeben hat. Wenn ein Angreifer jedoch als Suchbegriff

Listing 3.4: Gefährlicher Suchbegriff

```
1   begriff%'; DROP TABLE `articles`; --
```

eingibt, lautet die ausgeführte Abfrage:

Listing 3.5: Resultierender SQL-Query

```
1  SELECT * FROM `articles` WHERE `content` LIKE '%
     begriff%'; DROP TABLE `articles`; --%'
```

Diese Abfrage besteht jedoch nun aus zwei Abfragen:

1. `SELECT * FROM `articles` WHERE `content` LIKE '%begriff%';`
2. `DROP TABLE `articles`;`

[4] Der Google Bot ist die Software, welche für Google das Internet durchforstet

Die beiden Bindestriche stellen einen Kommentar bis zum Ende der Zeile dar in SQL, sodass die Überreste von dem ursprünglichen Query in Listing 3.3 ignoriert werden.
Die erste Anfrage ist eine harmlose Suche nach `begriff` in der Spalte `content`. Die zweite Abfrage dagegen stellt eine Gefahr dar, da hier die Tabelle `articles` aus der Datenbank gelöscht wird. Als Alternative zum Löschen der Datenbank könnte sich ein Angreifer mittels SQL Injection einen Account anlegen oder einen bestehenden Account höherstufen, sodass dieser mehr Rechte zur Verfügung hat.

Bei parametrisierten SQL Queries „prepared statements“ nutzen

Um SQL Injections effektiv zu verhindern, verwendet man sogenannte „prepared statements“. Der Datenbank wird nicht die fertige Abfrage übermittelt, sondern stattdessen eine Vorlage für einen Query. Bestimmte Werte werden dabei nicht als solche übergeben, sondern als Platzhalter. Im zweiten Schritt übergibt man der Datenbank die Werte, die sie für die Platzhalter einsetzen soll. Da der Query bereits im ersten Schritt von der Datenbank geparst wurde, werden nun lediglich die Werte eingesetzt, ohne dabei jedoch den ursprünglichen Query zu verändern wie in Query (3.5).
Eine verbesserte Version von Beispiel 3.1.8 würde in PHP wie folgt aussehen:

```
1  $pdo = new PDO(/** parameters **/);

   $statement = $pdo->prepare('SELECT * FROM `articles`
      WHERE `content LIKE ?');
   $statement->execute(['%' . $search . '%']);
```

In Zeile (3) ist `?` der Platzhalter für den in Zeile (4) übergebenen Suchtext.
Falls ein Angreifer nun den Suchbegriff aus Listing 3.4 eingeben würde, würde auch nach genau dem gesucht, da der Platzhalter intern in der Datenbank eingesetzt wird und nicht in den eigentlichen Query.
Innerhalb dieser Arbeit ist auch eine Challenge zu SQL Injections entstanden. In dieser Challenge müssen die Schüler die Abfrage zum Anmelden so manipulieren, dass sie sich anmelden können ohne das Passwort zu kennen. Um den Schülern entgegenzukommen, wurde die Anmeldefunktionalität so implementiert, dass bei einem fehlerhaften Query eine Fehlermeldung samt ausgeführtem Query ausgegeben wird. Somit behandelt diese Challenge nicht nur das Problem der SQL Injections, sondern zeigt darüber hinaus, dass Fehlermeldungen in Produktivsystemen nie sicherheitskritische Informationen herausgeben dürfen.
Die Challenge wurde auf vServer #3 (`vsrv3`) installiert. Dort wurde für diese Challenge ein Datenbankbenutzer angelegt, welcher lediglich über die Rechte `SELECT`, `INSERT` und `UPDATE` verfügt. Somit ist es nicht gestattet, Tabellen anzulegen, zu verändern, zu leeren oder zu löschen. Dies stellt zwar eine Einschränkung bei der Durchführung der SQL Injection dar, allerdings ist dies nötig um die Funktionalität der Challenge sicherzustellen. Wäre die verwendete Datenbanktabelle leer, nicht vorhanden oder hätte eine andere Struktur als die vorgegebene,

wären andere Schüler nicht in der Lage, mit Hilfe der Hinweise die Challenge zu lösen.

3.2. Erweiterungsoptionen

Wie bereits in der ursprünglichen Version des HackIt!-Frameworks ist es möglich, weitere Challenges zu erstellen. Das Erstellen neuer Challenges hat sich aufgrund der neuen Struktur von Challenges verändert. Challenges lassen sich derzeit nicht über das Webinterface, sondern nur über ein Installations-Skript anlegen. Anschließend müssen Challenge-spezifische Dateien in das angezeigte Verzeichnis kopiert werden.
Eine Challenge besteht aus folgenden Dateien, wovon die ersten beiden Pflicht sind:

- `challenge.php`
- `index.body.php`
- `index.head.php`

In der Datei `challenge.php` wird Challenge-spezifischer Quellcode erwartet. Dieser sollte unter anderem Code enthalten, welcher dafür sorgt, dass die Challenge als gelöst markiert wird. Diese Datei wird zu Beginn der Ausführung einer Challenge eingebunden und ausgeführt.
In der Datei `index.body.php` befindet sich die Ansicht der Challenge, wie beispielsweise Formulare oder andere HTML-Elemente. Sie kann darüber hinaus weiteren PHP-Code enthalten, welcher für die Challenge benötigt wird.
Die Datei `index.head.php` wird innerhalb des `<head>`-Tags der HTML-Seite eingebunden, sodass zum Beispiel JavaScript-Code eingebettet werden kann.
Das Grundgerüst der Challenge-Ansicht wird später durch das Installations-Skript `/var/www/challenge-setup.py` angelegt. Dieses besteht aus einer `config.php`-Datei, welche den Benutzernamen und das Passwort für die MySQL-Datenbank enthält, sowie aus einer `index.php` welche das Grundgerüst liefert und die Challenge-spezifischen Dateien an den entsprechenden Stellen einbindet.
Die Dateien für das Grundgerüst befinden sich im Verzeichnis `/var/www/common-files`, wobei die Dateien nicht kopiert werden, sondern nur ein Hardlink angelegt wird, sodass sich eine Änderung dieser Dateien automatisch auf alle Challenges auswirkt.
Innerhalb der Challenge können beliebige PHP-Funktionen genutzt werden. JavaScript-Code kann auf Grundlage von jQuery geschrieben werden, auch wenn dies nicht zwingend notwendig ist. Als Bibliothek für Cascading-Style-Sheets (CSS) kommt Bootstrap zum Einsatz, welches ebenfalls innerhalb der Challenge und ohne zusätzlichen Aufwand benutzt werden kann.

3.3. Einsatz im Unterricht

Um die Software im Unterricht einzusetzen, wurde im Rahmen der Arbeit die Anleitung zum Bereitstellen des HackIt!-Frameworks erneuert. Dieser ist im Appendix B enthalten. Dort wird neben der eigentlichen Bereitstellung auch aufgezeigt, welche Voraussetzungen für die Software nötig sind.

Zwei Challengessets verfügbar

Die im Abschnitt 3.1 vorgestellten Challenges sind aufgeteilt auf zwei verfügbare Challengesets „Basic challenges“ und „Advanced challenges“(siehe Appendix A).

Das erste Challengeset enthält dabei einfache und mittel schwere Challenges. Diese sind zunächst nach Themengebiet und innerhalb dessen nach Schwierigkeitsgrad sortiert. Diese Reihenfolge stellt dabei auch eine sinnvolle Bearbeitungsreihenfolge dar, da diese im Verlauf etwas in ihrer Komplexität und Schwierigkeit steigen. Einige Challenges bauen auf anderen Challenges auf.

Die „advanced challenges“ richten sich an erfahrene Schüler, welche bereits alle Challenges der „basic challenges“ gelöst haben. Diese Challenges besitzen den Schwierigkeitsgrad mittel oder schwer, sodass hier selbst erfahrene Schüler knobeln dürften.

Die Reihenfolge und Kategorisierung stellt dabei nur eine Grundlage dar und kann nach Belieben von der Lehrkraft verändert werden.

Kapitel 4 Fazit

4.1. Zusammenfassung

Im Rahmen dieser Arbeit wurde das ursprünglich von Florian Kerber entwickelte HackIt!-Framework technisch weiterentwickelt und inhaltlich für Schüler der Sekundarstufe II aufbereitet. Neben einem neuen Design wurden nützliche Funktionen entwickelt, unter anderem ein Hinweis- und Punktesystem. Für Lehrer bietet die Software weitere Feedback-Funktionen wie die Fortschrittsanzeige und Übersicht der aufgerufenen Hinweise. Außerdem wurden die Verwaltungstools überarbeitet, sodass die Administration der Software erleichtert wird. Während der Weiterentwicklung wurde das Kernsystem neu implementiert und dank des MVC-Patterns ist das System leicht um neue Features erweiterbar.
Neben den technischen Neuerungen wurden auch die Challenges überarbeitet. Diese verfügen nun über Hinweise und Post-Stories, sodass Schüler während der Challenge unterstützt werden und abschließend Informationen zu dem in der Challenge ausgenutzten Fehler erhalten. Darüber hinaus enthalten die Post-Stories auch Lösungen zu den in den Challenges angesprochenen Problemen, damit solche Fehler in selbst implementierten Systemen nicht auftreten.
Das System stellt nun eine abgeschlossene Software dar, welche für den praktischen Einsatz – beispielsweise im Informatikunterricht der Sekundarstufe II oder während einer Projektwoche – geeignet ist.

4.2. Ausblick

Zum einen kann das Hinweis-System um eine Feedback-Option erweitert werden, sodass Schüler nach einem Hinweis angeben können, ob sie diesen als hilfreich oder wenig hilfreich empfunden haben. Mit Hilfe dieses Feedbacks kann der Lehrer gegebenenfalls die Hinweise anpassen.
Weiterhin lässt die neue Challengeübersicht viel Raum für Verbesserung, da der zur Verfügung stehende Platz nicht optimal ausgenutzt wird und der Schüler mit Zunahme an Challenges mehr scrollen muss, um weiter unten positionierte Challenges zu erreichen.
Bisher lassen sich Challenges nur über ein Setup-Skript und das anschließende Hochladen der Dateien mittels SSH/SCP installieren. Um

diese Prozedur zu vereinfachen, könnte man eine Möglichkeit implementieren, neue Challenges als ZIP-Archiv im Web-Interface hochzuladen. Die eigentliche Installation wird nach dem Hochladen automatisch durchgeführt. Ergänzt man noch eine Export-Funktion für Challenges, wäre es auf sehr einfache Weise möglich, Challenges untereinander auszutauschen. Diese Funktion könnte Informatiklehrern das Austauschen von Challenges erleichtern.
Die mitgelieferte virtuelle Maschine basiert auf einem veralteten Debian, wobei die internen virtuellen Maschinen durch die Software `vserver` virtualisiert werden. Die auf dem System installierte Software ist veraltet und müsste aktualisiert werden, damit sichergestellt wird, dass das System nicht selbst zum Angriffsziel wird.
Inhaltlich lässt sich das System wie bereits in Abschnitt 3.2 beschrieben um neue Challenges erweitern. Insbesondere die Challenge *Cryptography* erfordert weitere Aufbereitung für Schüler. Da viele Challenges JavaScript voraussetzen und JavaScript nicht zwingend im Informatikunterricht behandelt werden muss, ließe sich das System um ein interaktives JavaScript-Tutorial ergänzen. Innerhalb dieses Tutorials könnte den Schülern auch der Umgang mit der JavaScript-Konsole in den Entwicklertools interaktiv gezeigt werden.

Literaturverzeichnis

Printed References

[1] Nadine Bergner. „IU in der Sek. I (Teil 1)“. In: *Einführung in die Fachdidaktik Informatik* (2014), S. 19.

[2] Florian Kerber. „HackIt!-Framework“. Bachelorarbeit. RWTH Aachen University, 2012.

[3] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen. *Kernlehrplan für die Sekundarstufe II Gymnasium / Gesamtschule in Nordrhein-Westfalen – Informatik*. 1. Aufl. Düsseldorf, 2013.

[4] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen. *Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe im Jahr 2015*. 2012.

[7] Matthias Deeg Stefan Arbeiter. „Bunte Rechenknechte – Grafikkarten beschleunigen Passwort-Cracker“. In: *c't 06/2009* (), S. 204.

Online References

[5] Heise online. *Einbruch bei Adobe: Millionen Kundendaten sowie Sourcecode von ColdFusion und Acrobat geklaut*. URL: `http://www.heise.de/newsticker/meldung/Einbruch-bei-Adobe-Millionen-Kundendaten-sowie-Sourcecode-von-ColdFusion-und-Acrobat-geklaut-1972175.html` (besucht am 25. 05. 2015).

[6] *Projekt-Homepage von Smarty*. URL: `http://www.smarty.net/` (besucht am 25. 05. 2015).

[8] Wikipedia. *Cron*. URL: `https://de.wikipedia.org/wiki/Cron` (besucht am 31. 05. 2015).

[9] Wikipedia. *Cross-Site-Scripting*. URL: `https://de.wikipedia.org/wiki/Cross-Site-Scripting` (besucht am 31. 05. 2015).

[10] Wikipedia. *Directory Traversal*. URL: `https://de.wikipedia.org/wiki/Directory_Traversal` (besucht am 31. 05. 2015).

[11] Wikipedia. *Hash function security summary*. URL: `https://en.wikipedia.org/wiki/Hash_function_security_summary` (besucht am 25. 05. 2015).

[12] Wikipedia. *Internet der Dinge*. URL: https://de.wikipedia.org/wiki/Internet_der_Dinge (besucht am 25. 05. 2015).

[13] Wikipedia. *Markdown*. URL: https://de.wikipedia.org/wiki/Markdown (besucht am 31. 05. 2015).

[14] Wikipedia. *Model View Controller*. URL: https://de.wikipedia.org/wiki/Model_View_Controller (besucht am 25. 05. 2015).

[15] Wikipedia. *Prepared Statements*. URL: https://de.wikipedia.org/wiki/Prepared_Statement (besucht am 25. 05. 2015).

[16] Wikipedia. *Smart Home*. URL: https://de.wikipedia.org/wiki/Smart_Home (besucht am 25. 05. 2015).

[17] Wikipedia. *Sony*. URL: https://de.wikipedia.org/wiki/Sony#Hacker-Angriffe_2011 (besucht am 25. 05. 2015).

Anhang A Übersicht der Challengesets

A.1. Basic challenges

1. **Basics #1 (einfach)**
   Das Passwort ist im Klartext im JavaScript-Code enthalten.

2. **Basics #2 (einfach)**
   Das Passwort ist im Klartext in einer externen JavaScript-Datei enthalten.

3. **Basics #3 (einfach)**
   Das Passwort muss mit Hilfe von `String.fromCharCode()` dekodiert werden.

4. **Basics #4 (einfach)**
   Das Passwort ist in einer Textdatei auf dem Webserver gespeichert. Der Dateiname ist im HTML-Quellcode versteckt.

5. **Basics #5 (einfach)**
   Das Passwort wird über die Passwort vergessen-Funktion mitgeteilt. Dazu muss zunächst die Empfänger E-Mail-Adresse mittels JavaScript geändert werden.

6. **User Agent (einfach)**
   Der User Agent des Browsers muss verändert werden, um Zugang zu erhalten.

7. **Hash #1 (einfach)**
   Das Passwort ist als MD5-Hash im JavaScript-Code enthalten und muss zunächst „entschlüsselt“ werden.

8. **HTTP #1 (einfach)**
   Der Wert eines Cookies muss geändert werden.

9. **HTTP #2 (mittel)**
   Die `htpasswd`-Datei des Webservers muss ausgelesen werden, da dort Benutzername und Passwort enthalten sind.

10. **Parameter Injection (mittel)**
    Durch Parameterinjektion müssen sie eine Datei in einem geschützten Verzeichnis auslesen. Dort befinden sich Benutzername und Passwort.

11. **SQL Injection (mittel)**
    Es muss eine SQL Injection durchgeführt werden, um die Anmeldeprozedur zu umgehen.

A.2. Advanced challenges

1. **Code Execution (mittel)**
   Es muss das Programm `ls` aufgerufen werden, sodass eine Auflistung der Dateien angezeigt wird. Dort befindet sich eine Datei mit einem Passwort. Diese kann über den Webserver aufgerufen werden.

2. **Hash #2 (schwer)**
   Es muss eine Kollision gefunden werden beziehungsweise es muss eine bekannte Kollision angewendet werden.

3. **Hash #3 (schwer)**
   Es muss eine „length extension attack“ durchgeführt werden.

Anhang B Anleitung zum Bereitstellen der Software (Lehrer)

Contents

1 Introduction **1**

2 Requirements **1**
2.1 Enable hardware virtualisation . . . 1
2.2 Requirements on client-side . . . 2

3 Step by step guide **2**

4 Update your firewall **4**

5 Starting **5**

6 Shutting down **6**

7 Important credentials **7**
7.1 root access . . . 7
7.2 Admin access to the framework . . . 7

8 Troubleshooting **8**
8.1 VT-x is not available . . . 8

1 Introduction

This guide will show you how to deploy the HackIt!-Framework for you own usage. The framework comes as a VirtualBox appliance that can easily be imported.

This guide was created on Windows 8.1, but all menus etc. should be equal on Linux/Mac.

2 Requirements

To keep it very simple: all you need is a computer that runs VirtualBox and features 2 GB of RAM and 4 GB of hard disk space. For great experience I suggest the following:

1. Dual-Core CPU with virtualisation capabilities (this includes all Intel Core i processors)
2. Hardware virtualisation should be enabled in BIOS/UEFI
3. 4 GB of RAM

Hint

In order to enable hardware virtualisation, ensure you only have one hypervisor installed on your computer. Other hypervisors other than VirtualBox are Hyper-V, VMWare, XEN, KVM.

If your computer does not support hardware virtualisation, you are still able to deploy the system but the overall performance might be a bit slower.

2.1 Enable hardware virtualisation

Somehow, hardware virtualisation is disabled by default on most computers and notebooks. To enable it, open your BIOS or UEFI setup and seach for an option like "Intel virtualization technology" (other indicators can be VT-x or AMD-V).

If your computer is running Windows 8 or 8.1 you can easily open UEFI setup through Windows itself. Simply follow the following guides provided here[1] (Windows 8) and here[2] (Windows 8.1). If your computer runs Windows 7 or Linux/Mac, please ask the internet on how to open BIOS setup or how to enable hardware virtualisation.

In case you own a Mac, consider https://support.apple.com/en-us/HT203296 for futher information on hardware virtualisation.

2.2 Requirements on client-side

As this software is a web-application, you can use any browser you want. Nevertheless, it is recommended to use either Google Chrome or Mozilla Firefox, because all guides only cover those two browsers. Also, other browsers might have some disadvantages (for example, Internet Explorer does not have support for plugins which is needed for at least one challenge). In case you want to use another browser than Mozilla Firefox or Google Chrome, you must ensure that you provide a student's guide. Also, take a look at all hints and watch our for browser-specific hints.

3 Step by step guide

First, install VirtualBox on the computer that you want to host this framework on. You can either use your package manager or download the software at https://www.virtualbox.org/.

Aftwards, open VirtualBox and click File > Import Appliance. Then select the image and continue with Next.

[1] https://www.youtube.com/watch?v=QGiG1oIjjZI
[2] https://www.youtube.com/watch?v=XZcYbxOcPV0

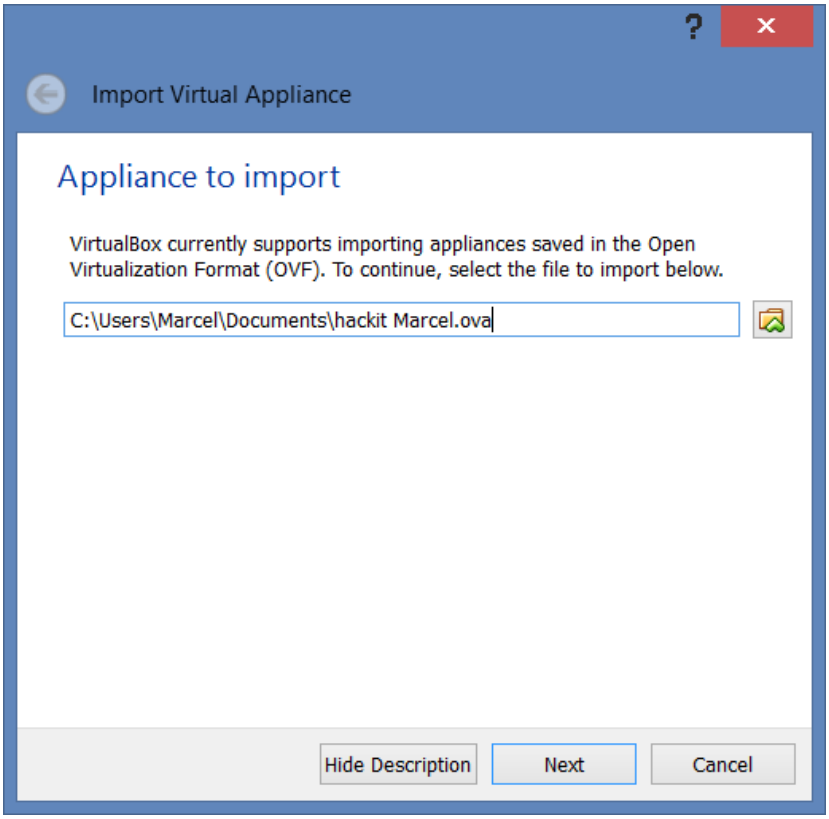

Figure 1: Import wizard: select the provided appliance

In the upcoming "appliance settings" you do not need to change anything.

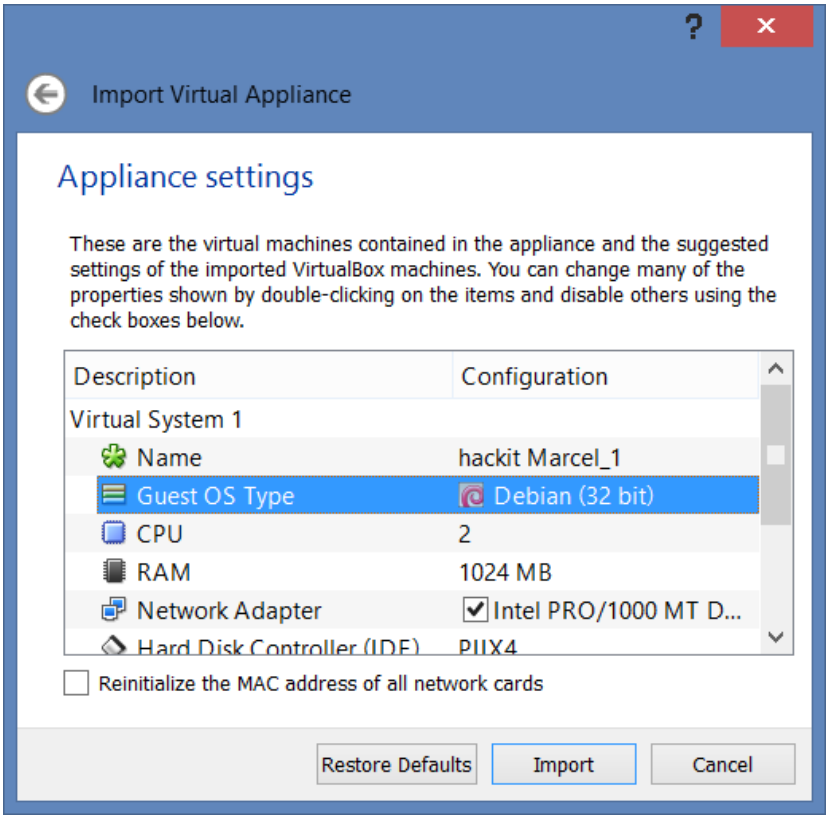

Figure 2: Import wizard: summary of the imported appliance

Finally click Import and wait until the import process is finished.

Now you have successfully imported the appliance and are ready to go.

4 Update your firewall

The network of the imported VirtualBox appliance was configured as NAT which means that the virtual machine is not visible from outside the host machine. Thus, you must open port 80 in the firewall of your host machine.

On Windows Vista or greater, you can do this by opening cmd.exe as administrator and execute

```
netsh advfirewall firewall add rule name="HTTP Server (HackIt Framework)" dir=out action=allow protocol=TCP localport=80
netsh advfirewall firewall add rule name="HTTP Server (HackIt Framework)" dir=in action=allow protocol=TCP localport=80
```

Another way to do this on Windows is provided here[3].

[3] http://windows.microsoft.com/en-us/windows/open-port-windows-firewall

5 Starting

Before you hit the Start button, be sure you are not running any service on port 22 or 80. Those ports are used for SSH access (port 22) and the web server (port 80).

Now hit the Start button and wait until you see the following prompt as in figure 3. If hardware virtualisation is disabled, VirtualBox will fail to start the machine. See chapter 8.1 for further help.

Note: You can ignore the error message at the end of the promt which says that "mounting failed".

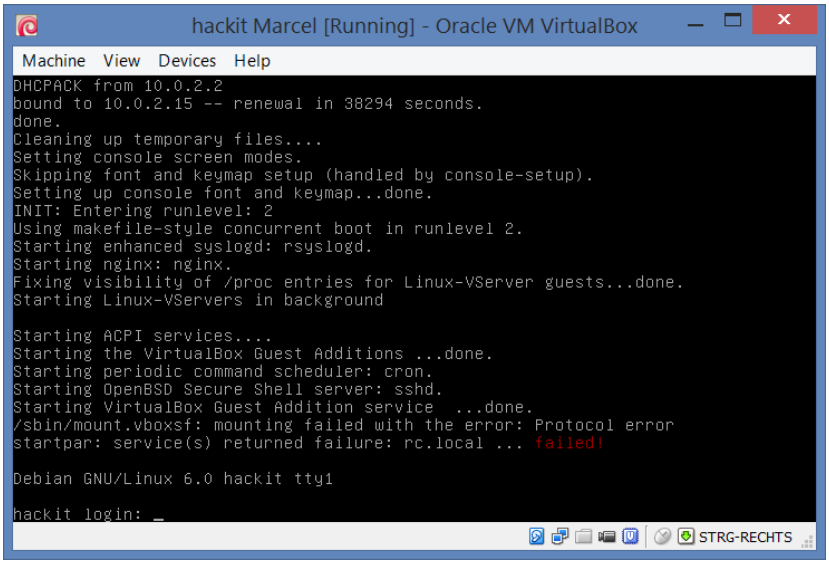

Figure 3: Linux shell

Then simply open a browser of your choice and head over to http://localhost/. You should see a login prompt as in figure 4.

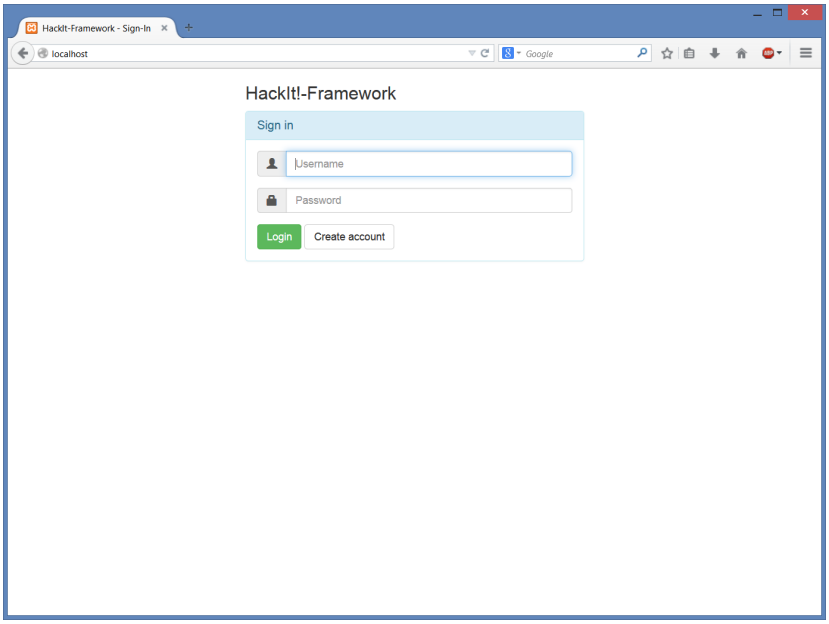

Figure 4: Login prompt

6 Shutting down

There are two ways to shut down the virtual computer. You can either login to the machine and use the linux programm "halt" or you can click on "Machine" > "ACPI Shutdown" in the VirtualBox window (see figure 5).

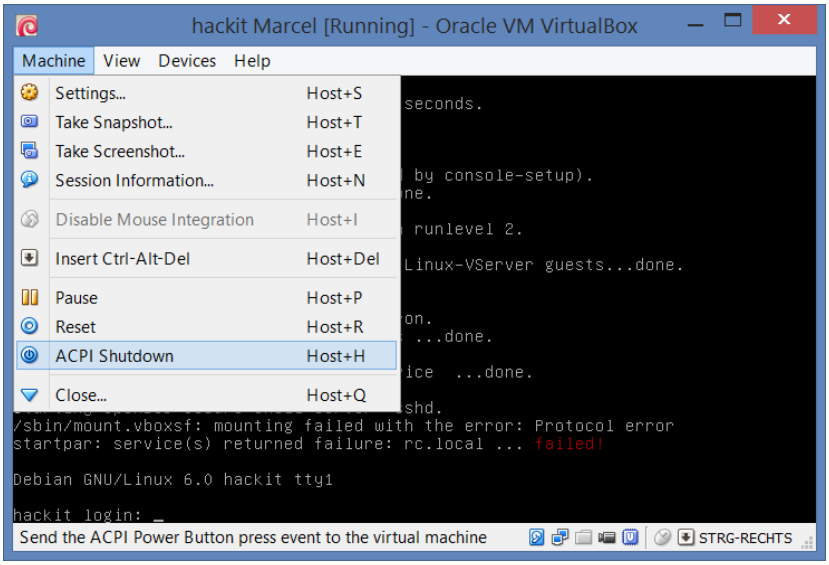

Figure 5: This way you can make the machine shut down

7 Important credentials

7.1 root access

If you need root access to the linux guest system, you can use the following credentials:

Username: root
Password: 12345

I recommend changing this password by logging into the machine and executing "passwd".

Hint

Instead of using the console inside the VirtualBox window, you may use an SSH client (like Putty) to connect to the guest system. You have to connect to localhost on port 22 and use the credentials provided above.

7.2 Admin access to the framework

The framework itself runs in your browser, too.

URL: http://localhost/
Username: admin
Password: adminadmin

I recommend changing this password by logging into the framework and clicking on the username (admin) on the top menu (upper right corner).

8 Troubleshooting

8.1 VT-x is not available

If you want to use the software on an computer which does not support hardware-enabled virtualisation, VirtualBox will fail to start the virtual machine (see figure 6).

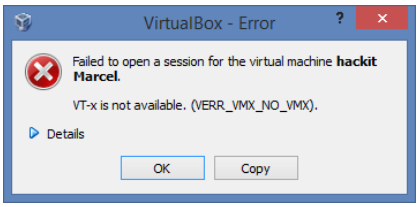

Figure 6: This way you can make the machine shut down

In order to fix this, you have to disable hardware acceleration in the settings of the virtual machine. To do so, right-click on the machine and click "settings". A settings dialogue appears. Click "System" on the left hand-side of the window and inside the System settings, select "Processor" and disable the "PAE/NX" option (see figure 7). Apply the new settings by clicking "OK".

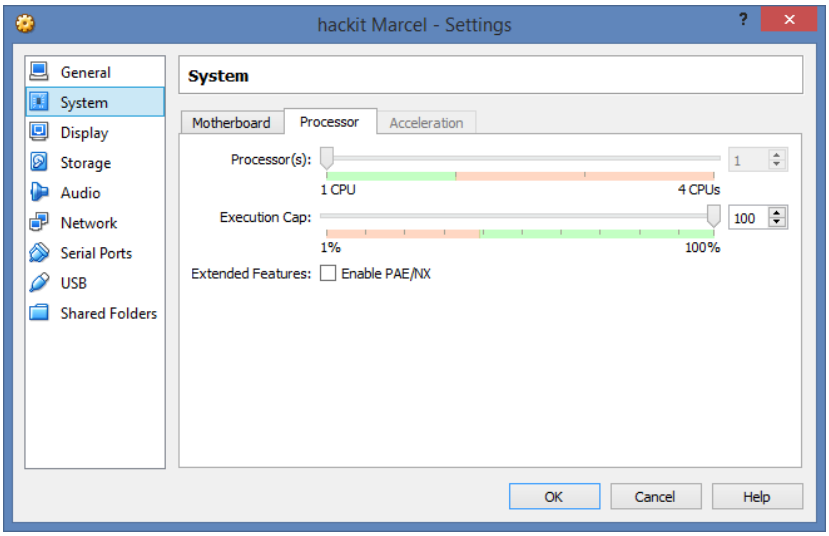

Figure 7: This way you can make the machine shut down

Now VirtualBox will not fail to start anymore.

Anhang C Anleitung zur Benutzung der Software (Schüler)

Contents

1 Introduction **1**

2 Requirements **1**

3 Using the software **1**
3.1 Registration 1
3.2 Login 1
3.3 Challenges 2

4 Getting started hacking **3**
4.1 Using the developer tools 4

5 Hints **5**
5.1 Requesting a hint 5

1 Introduction

This guide will show you how to use the HackIt!-Framework. This web-based software confronts you with common security issues. In each challenge you have to get access to a private area. But before we get started, ensure that all requirements are fulfilled.

2 Requirements

All you actually need is a web browser and a working network connection to the server on which the software runs. It is recommended to use advanced web browsers like Google Chrome or Mozilla Firefox. In this guide, we'll use Mozilla Firefox.

3 Using the software

3.1 Registration

First, you need to register. Therefore, open your browser and head to the URL where the HackIt!-Framework is located.

Click on the button "Create Account" and follow the registration wizard. In the section "Additional information" you can leave all fields empty. After you have finished the registration wizard you should see a popup indicating that your account was created successfully.

3.2 Login

You are now able to login. After your first login you probably won't see any challenge. That is because the administrator has to assign the challenges to you.

Caution

Do not press F5 on your keyboard when you see this page. Otherwise, you will be logged out. (This will be fixed in the future.)

3.3 Challenges

After your administrator has assigned challenges to you, you see a screen like the one in figure 2. Challenges are grouped into challenge sets. Your administrator can assign one or more challenge sets to you. To switch between challenge sets, use the dropdown menu "Available Challenge Sets".

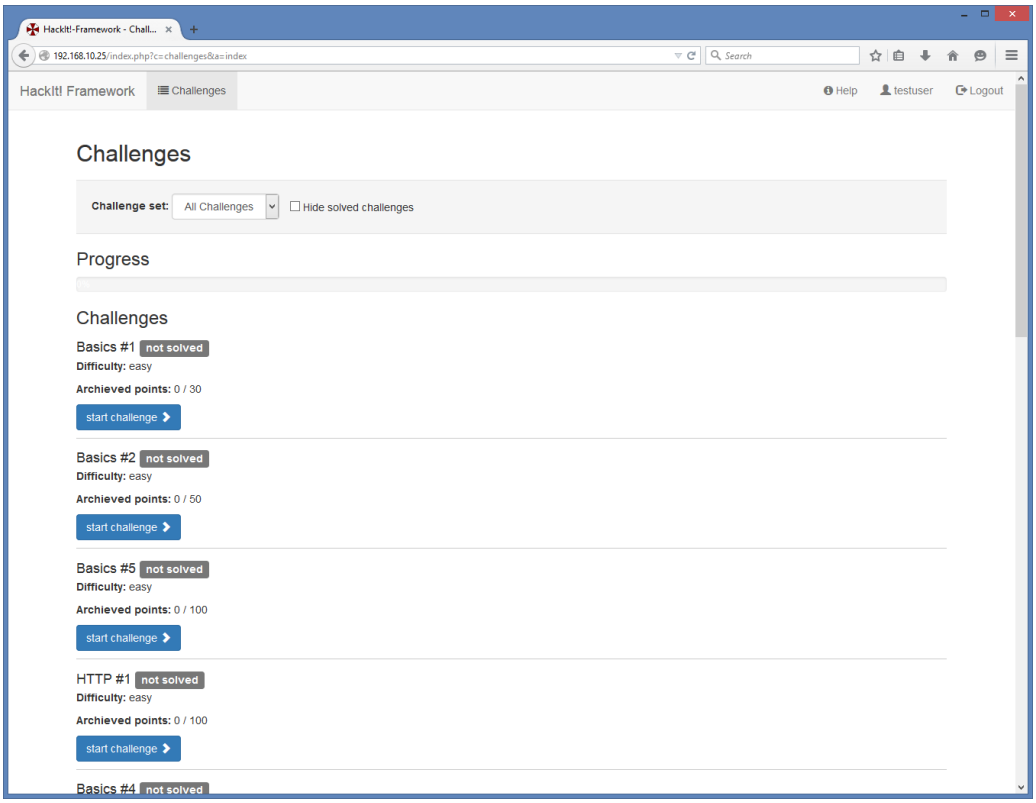

Figure 1: Challenges overview

Each box represents one challenge. When you hover over such a box with your mouse, you see details about this challenges in the box at the bottom of the page. To start a challenge, just click on the corresponding box, read the challenge story and click on the button "Go to challenge". The challenge will then open in a new tab.

In currently all challenges you have to gain access to a private area. In the example (see figure 2) you have to find out the password in order to login. All challenges provide hints to help you solve the challenges. Use the menu at the top of the page.

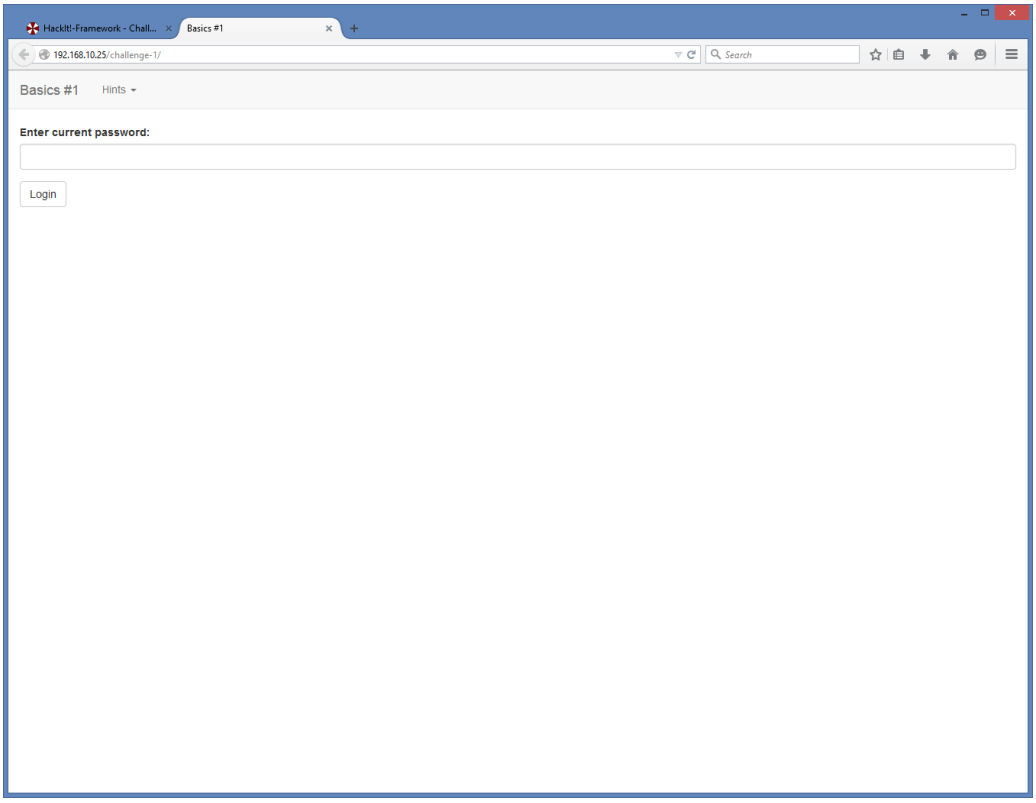

Figure 2: Challenge

If you figured out how to solve the challenge and you were right, the challenge is marked as solved and you see a message saying that you did it. You also see how many points you gained with solving the challenge. Some challenges provide additional information about the problem the challenge was about. It is recommended to read this text so you can prevent those problems in your own applications.

You can now close the tab with the challenge inside. When you return to the challenge overview, you will notice that the challenge you just solved is marked as solved.

4 Getting started hacking

Now as you know how to use the system in general, this guide will give you some basics. In currently all challenges it is a good idea to take a look at the source code of the challenge to learn about the structure and behavior of the challenge.

Hint

In order to show the source code of a website, press Ctrl + U or cmd + U (Mac)

As the challenge page also contains a menu, which is not challenge-specific, the developer decided to indicate the challenge-specific parts:

```
<!-- CHALLENGE BEGIN -->
```

```

    <!-- CHALLENGE END -->
```

Everything that belongs to the challenge is in between these comments (note: there might be more than only one challenge part).

4.1 Using the developer tools

Modern browsers like Google Chrome and Mozilla Firefox provide very useful tools for developers. Hit F12 (or fn + F12 on Mac) to start them. In Mozilla Firefox, the HTML inspector will open (see figure 3). This tool will display the HTML source code in a tree view for easy navigation through the code.

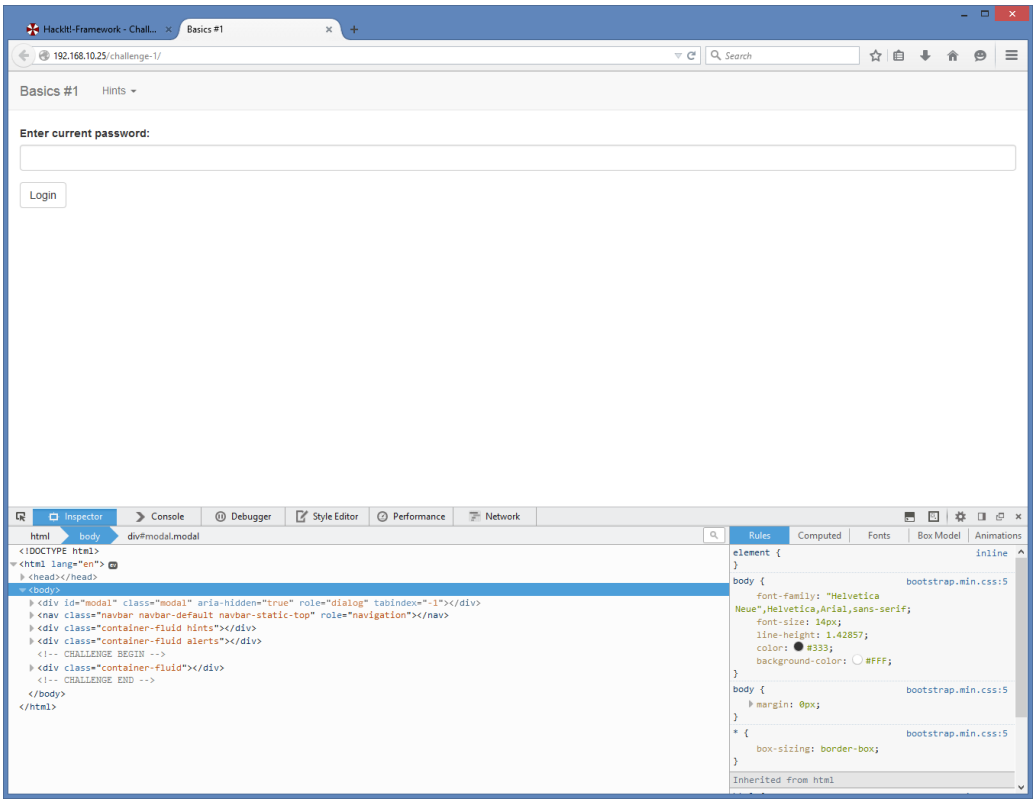

Figure 3: HTML inspector

Hint

If you want to navigate to the code of a specific element on the page, click and then click at the element you want to view the source code of

Another very useful tool is the JavaScript console. Simple click on the "Console" button at the top of the developer tools (see figure 4).

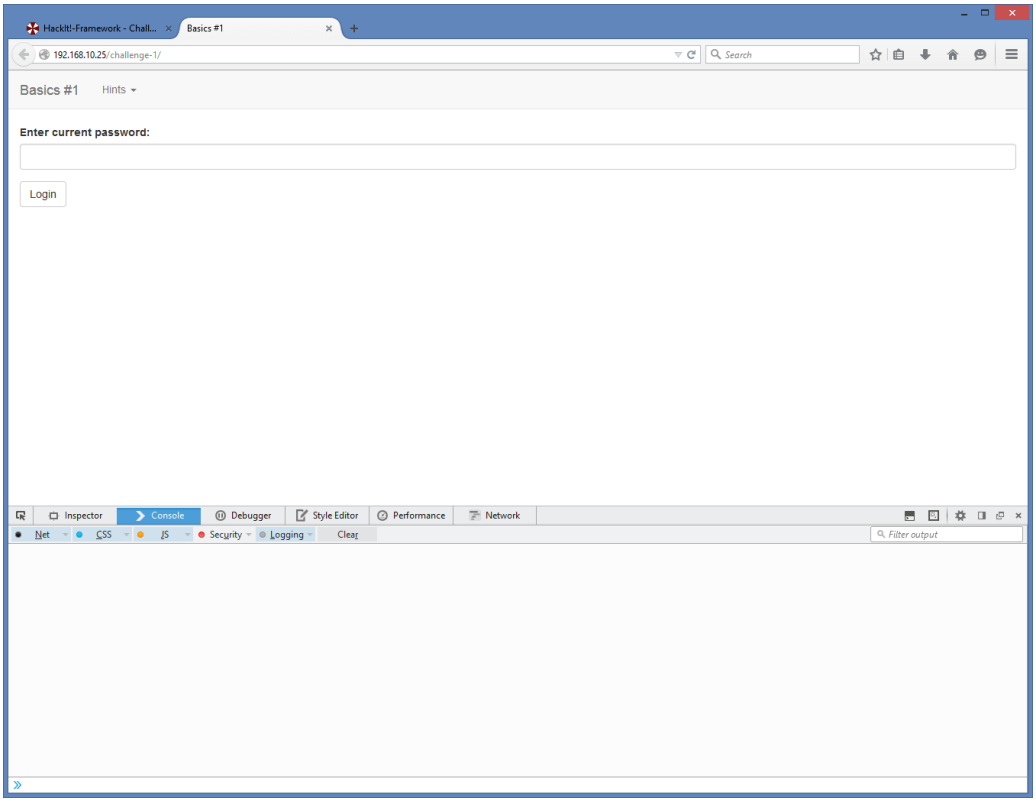

Figure 4: JavaScript console

At the bottom of this console window, there is a textfield in which you can type any JavaScript code. Hit enter to execute it. For example, type

```
alert("Hello");
```

and hit enter.

5 Hints

In every challenge you can request hints if you do not know what to do.

But be careful: you gain points whenever you finish a challenge. This number of points depends on the challenge. You can see the number of points for each challenge in the overview of challenges.

Each request of a hint will reduce the number of points you gain when you finish the challenge. But this is not a bad thing! The hints will guide you through the challenge whenever you need help.

5.1 Requesting a hint

Requesting a hint is very straight forward. In the challenge view, click on "Hints" in the upper menu.

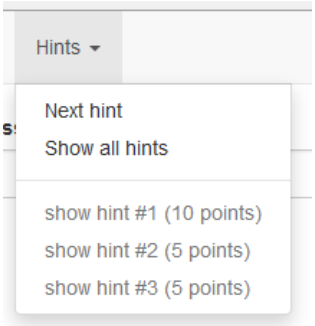

Figure 5: Hint menu

You can request a hint by clicking on "Next hint" (see figure 5).

In the menu, you can also view all requested hints at once by clicking on "Show all hints". If you do not wish to see all hints at the same time, you can also click "show hint #n" (where n is the hint number) to only see one hint.

Anhang D DVD

Erklärung

Ich versichere, dass ich die Bachelorarbeit – einschließlich beigefügter Zeichnungen, Kartenskizzen und Darstellungen – selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle deutlich als Entlehnung kenntlich gemacht.

Aachen, 09. Juni 2015 Marcel Marnitz