



universität
wien

DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

„Kekse ohne Salz schmecken nicht“

Ein innovatives Unterrichtskonzept zur Vermittlung von
Security im Webdatenbereich

verfasst von / submitted by

Simon Marik

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magister der Naturwissenschaften (Mag.rer.nat.)

Wien, 2017 / Vienna, 2017

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 190 884 313

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Lehramtsstudium UniStG
UF Informatik und Informatikmanagement UniStG - 10/2003
UF Geschichte, Sozialkunde und Polit. Bildg. UniStG - 10/2008

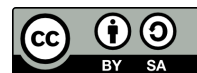
Betreut von / Supervisor:
Mitbetreut von / Co-Supervisor:

ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig
Ass.-Prof. Mag. Dr. Christian Cenker

Lizenzierung

Dieses Werk ist unter einem internationalen Creative Commons „Namensnennung - Weitergabe unter gleichen Bedingungen“ - Lizenzvertrag der Version 4.0 lizenziert (*i.e. "Attribution - ShareAlike 4.0 International CC BY-SA 4.0"*). Um eine Kopie dieser Lizenz online einzusehen, besuchen Sie bitte <http://creativecommons.org/licenses/by-sa/4.0/> oder senden Sie direkt eine schriftliche Anfrage an Creative Commons, Postfach 1866, Mountain View, CA 94042, USA.

Die dazugehörige Projektarbeit dieses Werks ist „freie Software“. Sie können die Projektdateien unter den Bedingungen der „GNU GPL“ (*i.e. "GNU General Public Licence"*), wie von der Free Software Foundation, Version 3 oder (nach Ihrer Wahl) jeder neueren veröffentlichten Version der Lizenz, weiterverbreiten und/oder modifizieren. Diese Projektarbeit wird in der Hoffnung, dass sie nützlich sein wird, aber ohne jede Gewährleistung und sogar ohne die implizite Gewährleistung der Marktfähigkeit oder der Eignung für einen bestimmten Zweck bereitgestellt. Sie sollten eine Kopie der Lizenz zusammen mit der Projektarbeit dieses Werks erhalten haben. Falls nicht, besuchen Sie bitte <https://www.gnu.org/licenses/gpl-3.0.html>.



Vorwort

Mit dieser Diplomarbeit schließe ich mein Lehramtsstudium in den Unterrichtsfächern „Informatik und Informatikmanagement“ sowie „Geschichte, Sozialkunde und Politische Bildung“ an der Universität Wien ab.

An erster Stelle möchte ich mich bei Frau ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig und Herrn Ass.-Prof. Mag. Dr. Christian Cenker für die Betreuung meiner Diplomarbeit sowie bei Frau ao. Univ. Prof. Mag. Dr. Meta Niederkorn und Herrn ao. Univ.-Prof. Dr. Hans Taeuber für die Abnahme meiner Diplomprüfung bedanken.

Mein Dank gilt auch all jenen Personen, welche mir für die zahlreichen Beratungsgespräche und Hilfestellungen im Zuge der Erstellung dieser Arbeit zur Seite gestanden sind. Bedanken möchte ich mich außerdem bei Herrn Mag. Werner Jägersberger für die Unterstützung sowie für die Teilnahme am Experteninterview.

Zum Abschluss möchte ich an dieser Stelle noch besonders meine Eltern, meine Großmutter und meine große Schwester hervorheben, welche mich im gesamten Verlauf meines Studiums in allen nur erdenklichen Lebensbereichen unermüdlich unterstützt haben und mir immer eine sehr große Stütze sowie meine größten Vorbilder waren.

Eidesstattliche Erklärung

„Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.“

Wien, am 04.02.2017

A handwritten signature in black ink, reading "Simon Maritz". The signature is written in a cursive style with a long, sweeping underline.

Abstract

Deutsch

Der hier vorliegenden Diplomarbeit liegt ein völlig neu entwickelter Zugang für die Thematik rund um Security im Webdatenbereich zugrunde, welcher für alle Lehrenden und Lernenden der Sekundarstufe II, aber auch im Studiengang Informatik als innovatives Unterrichtskonzept theoretisch sowie empirisch aufbereitet und veranschaulicht wurde.

Die Theorie umfasst dabei einerseits die aktuellen Standards der Webentwicklung (d.h. HTML5, CSS3, PHP5.6, MySQLi), eine Abgrenzung zu anderen Sicherheitstechnologien (d.h. SSL), sowie die Relevanz für den kompetenzorientierten Informatikunterricht.

Die Empirie besteht andererseits aus fünf modular aufgebauten Unterrichtsszenarien, welche sich in den Vortragsphasen jeweils auf die theoretischen Grundlagen und in den Übungsphasen auf das eigenständig entwickelte PHP Session-Login-System Musterbeispiel dieser Diplomarbeit beziehen. Mittels SWOT-Analyse und Experteninterview wird das Unterrichtskonzept abschließend noch empirisch begründet und abgerundet.

Schlagwörter: *Websecurity, PHP, Cookies, Salts, Verschlüsselung, Unterrichtsszenarien, SWOT, Interview*

English

This diploma thesis deals with a completely new approach, using security issues of the web data area. It has been theoretically and empirically illustrated as an innovative teaching concept, for teachers and students of the "Sekundarstufe II" as well as for computer science studies.

On one hand, the theory encompasses the current standards in web development (i.e. HTML5, CSS3, PHP5.6, MySQLi), other security technologies (i.e. SSL) and its relevance for the competence-oriented computer science education.

Empiricism on the other hand consists out of five modular designed teaching scenarios, which refer to the respective theoretical foundations of this diploma thesis during the phases of lecture and to the distinct developed PHP session-login-system prime example during tutorial phases. By use of the SWOT-analysis and an interview with an expert, the teaching concept gets empirically founded and blunt.

Keywords: *websecurity, PHP, cookies, salts, cryptology, teaching scenarios, SWOT, interview*

Inhaltsverzeichnis

1	Einleitung	15
1.1	Motivation	16
1.2	Zielsetzung	17
I	Theoretische Grundlagen	19
2	Standards in der Webentwicklung	21
2.1	HTML5	23
2.1.1	Metadaten	26
2.1.2	Tags	28
2.2	CSS3	31
2.2.1	Responsive Web Design	34
2.2.1.1	Media Queries	37
2.2.2	Scalable Vector Graphics	38
2.3	PHP	40
2.3.1	PHP 5.6	44
2.3.2	Externer Dateibezug	45
2.3.3	PHP-Sessions	47
2.3.3.1	Sessionvariablen	49
2.3.3.1.1	Einsatzmöglichkeiten	51
2.3.4	Cookies	52
2.3.4.1	Rechtliche Aspekte	55
2.3.4.1.1	Security	59
2.3.4.2	Probleme mit Cookies	60
2.3.4.3	Verschlüsselung von Cookies	62
2.3.4.3.1	Base64 Kodierung	64
2.3.4.3.2	SHA1 En-/Decryption	66
2.3.4.3.3	„BCrypt“ Blowfish-Algorithmus	68
2.3.4.4	Salted-Hash-Cookies	70
2.4	MySQL / MySQLi	72
2.4.1	Das SQL-Injection-Problem	75

2.4.1.1	Verschlüsselte Passwörter in Datenbanken	78
2.5	Webserver	79
3	Abgrenzung zu anderen Technologien	83
3.1	SSL / HTTPS	84
3.2	Vor- & Nachteile	87
4	Relevanz für den Unterricht	89
4.1	Rechtliche Rahmenbedingungen	90
4.1.1	SchOG - Schulorganisationsgesetz	90
4.1.2	SchUG - Schulunterrichtsgesetz	91
4.1.3	LBVO - Leistungsbeurteilungsverordnung	93
4.2	Lehrpläne Unterrichtsfach Informatik	94
4.2.1	Sekundarstufe II	95
4.2.1.1	Pflichtfach Informatik (5. Klasse)	97
4.2.1.2	Wahlpflichtfach Informatik (6. - 8. Klasse)	98
II	Empirische Arbeit	101
5	Unterrichtsszenarien	103
5.1	Pflichtfach Informatik (5. Klasse)	104
5.1.1	Szenario 1: Cookies im Webdatenbereich	104
5.1.1.1	Themenbeschreibung & Ausarbeitung	104
5.1.1.2	Planungsmatrix	106
5.1.1.3	Anwendungsbeispiele	107
5.1.2	Szenario 2: Webinterfaces anhand aktueller Standards	108
5.1.2.1	Themenbeschreibung & Ausarbeitung	108
5.1.2.2	Planungsmatrix	110
5.1.2.3	Anwendungsbeispiele	111
5.2	Wahlpflichtfach Informatik (6. - 8. Klasse)	112
5.2.1	Szenario 3: Der Umstieg von HTML auf PHP	112
5.2.1.1	Themenbeschreibung & Ausarbeitung	112
5.2.1.2	Planungsmatrix	114
5.2.1.3	Anwendungsbeispiele	115
5.2.2	Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies	116
5.2.2.1	Themenbeschreibung & Ausarbeitung	116
5.2.2.2	Planungsmatrix	119
5.2.2.3	Anwendungsbeispiele	120
5.2.3	Szenario 5: Authentifizierung anhand des Salted-Hash-Verfahrens	120
5.2.3.1	Themenbeschreibung & Ausarbeitung	120

5.2.3.2	Planungsmatrix	123
5.2.3.3	Anwendungsbeispiele	124
6	Proof of Concept	125
6.1	Experteninterview	126
6.1.1	Fragebogen	127
6.1.2	Ergebnis	127
6.2	SWOT-Analyse	130
6.2.1	Analyseergebnis	132
6.3	Adaption des Unterrichtskonzeptes	134
7	Conclusio	135
	Lebenslauf	137
	Literatur- & Quellenverzeichnis	139
	Abbildungsverzeichnis	145
	Tabellenverzeichnis	147
	Abkürzungsverzeichnis	148
A	„Index“ HTML Seitenaufbau (index.html)	153
B	„Layout“ Stylesheet (layout.css)	155
C	„Index“ PHP Seitenaufbau (index.php)	161
D	„Login“ Seitenaufbau (dashbd.php)	165
E	„Users“ Datenbank (thesis.sql)	167
F	„Connection“ Skript (scr.connect.php)	169
G	„Sign in“ Skript (scr.signin.php)	171
H	„Authentication“ Skript (scr.auth.php)	173

Kapitel 1

Einleitung

Dass Kekse ohne einer kleinen Prise Salz relativ fade bzw. nach gar nichts schmecken, wissen wir bereits seit unserer frühesten Kindheit, als wir alle noch fleißig in der Vorweihnachtszeit, gemeinsam mit unseren Müttern und Großmüttern, den Keksteig ausrollten, um ihn anschließend mit allen möglichen Formen auszusteichen, zu backen und schlussendlich kunstvoll zu verzieren. Die meisten können sich sicherlich noch sehr gut an die Belehrungen durch ihre Mütter und Großmütter erinnern, obwohl sich damals in den frühen 90er Jahren natürlich noch niemand hätte vorstellen können, dass man dieses „Knowhow“, sozusagen, auch später einmal wieder in ganz anderen Themenbereichen vorfinden und vor allem auch anwenden können wird. Aber mal ganz abgesehen vom naiven Kindesalter steckten zu dieser Zeit Computer sowie die komplette Informationstechnologiebranche sowieso gerade erstmal in ihren Kinderschuhen, wodurch selbstverständlich auch noch keine erdenklichen Parallelen zu einem vermeintlichen Internet oder gar Cookies gezogen werden konnten. Heute, fast drei Jahrzehnte später, finden wir uns plötzlich in einer komplett digitalisierten Welt wieder, in der wir – könnte man fast sagen – nur mehr von AGB, Sicherheitsbestimmungen oder Datenschutzrichtlinien geprägt sind. Um sich dieser neuen Herausforderung zu stellen, adaptierte selbst der Bildungsbereich das Thema „Datensicherheit, Datenschutz und Urheberrecht“¹ noch rechtzeitig im Curriculum, um die SchülerInnen sowie Studierenden schnellstmöglich dafür zu sensibilisieren.

Viele Kolleginnen und Kollegen aus dem Bildungsbereich haben jedoch öfters die Erfahrung gemacht, dass sich Lehrende bei diesem Themengebiet meistens nur von der Seite der „Social Networks“ aus annähernten und dabei andere wichtige Faktoren bzw. Zugänge komplett außer Acht ließen. In diesem Sinne wurde bei dieser Diplomarbeit Wert darauf gelegt, die ganze Thematik rund um „Security im Webdatenbereich“ auf Cookies aufzubauen. Diese haben nämlich ein viel breiteres Anwendungsgebiet und sind in der modernen Webentwicklung heutzutage kaum noch wegzudenken und werden den SchülerInnen einen interessanten, neuen Zugang für Sicherheit im WWW bieten. Aus diesem Grund

¹s. *Digi.Komp*: Das Kompetenzmodell Informatik 5. Klasse 2013, (Stand: 19.09.2016).

unterteilt sich diese Diplomarbeit in einen theoretischen und einen empirischen Teil, die wie folgt aufgebaut sind. Die Theorie – welche explizit für Lehrende abgestimmt wurde – beschäftigt sich einerseits mit den neuesten Webentwicklungsstandards (d.h. HTML, CSS, PHP, MySQLi), andererseits mit Abgrenzungen zu anderen Sicherheitstechnologien (d.h. SSL) sowie deren Relevanz für den Informatikunterricht. Im Zuge der Empirie wurde ein selbst entwickeltes Salted-Hash-Cookie basiertes PHP-Session-Login-System erstellt, welches als Musterbeispiel für die im Anschluss aufbereiteten Unterrichtsszenarien dienen soll. Die einzelnen Szenarien sind dabei komplett modular aufbauend gestaltet, welche sich in den Vortragsphasen wiederum auf die jeweiligen theoretischen Grundlagen dieser Diplomarbeit beziehen.

1.1 Motivation

Im Zuge verschiedenster Tätigkeiten am Grafik- und Webentwicklungssektor kommt man im Laufe der Zeit mit sehr vielen verschiedenen Internetapplikationen bzw. Webseiten für Kundinnen und Kunden diverser Branchenrichtungen in Berührung. Aber selbst als Informatiklehrkraft, oder auch im Zuge von entsprechenden Lehrveranstaltungen an der Universität, musste man bereits etliche Homepages und kleinere Content Management Systeme programmieren sowie unzählige kostenlose Produktlösungen, wie z.B. „Moodle“, „MediaWiki“, „Wordpress“, „CEWebS“ oder „CMSimple“ auf ihren Funktionsumfang und deren Konfigurationsmöglichkeiten ausgiebig testen. So standen auch stets die Benutzerfreundlichkeit der Bedienung bzw. des Interfaces sowie eine erleichterte Verständlichkeit für Personen mit eingeschränktem Informatikbezug im Mittelpunkt der Arbeit und Aufgaben. Aus diesem Grund stellen zur Zeit die neuesten Standards wie HTML5, CSS3 oder PHP5.6 essentielle Arbeitswerkzeuge dar, um einerseits „stylische“ und zugleich sichere Webprojekte zu erstellen und andererseits etwaigen Kundenwünschen sowie den eigenen Vorstellungen und Vorlieben zu entsprechen und gerecht zu werden.

Für viele Kundinnen und Kunden ist es auch oftmals ein Problem, dass man sehr stark an seinen Hosting Provider und dessen Möglichkeiten gebunden ist. Dabei kam auch oft die Problematik der verschlüsselten Verbindung mit der SQL-Datenbank bei Session basierten PHP-Login-Systemen auf. SSL verschlüsselte Serververbindungen haben manchmal – vor allem bei kleineren Backends – einen viel zu horrenden und meist auch unverständlichen Konfigurationsaufwand, wodurch Kundinnen und Kunden meist keine Zeit an diese Thematik verschwenden wollen, aber dennoch optimal geschützt sein möchten. Außerdem ist die SSL Verschlüsselung an vielen Stellen sehr unklar dokumentiert, birgt Probleme bei CORS-Operationen und ist besonders fehleranfällig bei diversen Suchmaschinenoptimierungen.² Interessanterweise wurde mehrfach berichtet, dass selbst die Erfahrung in

²s. *SSL Webhosting: SSL Proxy - Vor- und Nachteile und Einsatzmöglichkeiten*, (Stand: 19.09.2016).

punkto Professionalität und Performanz unter SSL bzw. unter HTTPS nicht dieselbe wäre wie bei einer Standardwebseite. Bei Besprechungen z.B. fielen an dieser Stelle oftmals die Termini des Salted-Hashing und des Cookies, welche eine gute Möglichkeit darstellen, um diese vielen Problematiken relativ zeitsparend zu umgehen. Für zukünftige sowie derzeitige Informatiklehrkräfte stellt diese Form der Verschlüsselung und der Webseitenprogrammierung vor allem einen spannenden Ansatz für ein innovatives Unterrichtskonzept dar, welches sehr leicht für SchülerInnen zu erlernen, anzuwenden, sowie umzusetzen ist. Dabei deckt es sehr viele verschiedene Themenbereiche des Informatikunterrichts ab und führt Kinder und Jugendliche einmal von einer komplett anderen Seite an die Thematik der Verschlüsselung von sensiblen Daten in Webprojekten heran.

1.2 Zielsetzung

Für die Bearbeitung dieser Diplomarbeit wurden neuwertige Webentwicklungskonzepte und deren Standards in Form von fünf verschiedenen Unterrichtsszenarien für das Unterrichtsfach Informatik erstellt. Die entsprechende Ausarbeitung soll in diesem Sinne Anregungen und Ideen schaffen, wie man die einzelnen Themen HTML, CSS, PHP, SQL sowie Internetsecurity in einem Unterrichtsblock zusammenfassen und sinnvoll umsetzen kann. Für diese Zusammenführung kam die Idee der Session basierten Login-Systeme auf, welche sich bereits oft in der Vergangenheit im Zuge von anderen Tätigkeitsbereichen als weit verbreiteter „State of the Art“ entpuppt hatten. Allerdings soll die Sicherheit in solchen Systemen ebenfalls gewährleistet sein, was mittels Cookies bzw. normalen Sessionvariablen oft ein Problem darstellt, da entweder herkömmliche Verschlüsselungsalgorithmen (z.B. „MD5“, „SHA1“) verwendet oder Passwörter und dergleichen gar ohne Verschlüsselung in Datenbanken abgelegt werden. Um SchülerInnen für diese Thematik zu sensibilisieren, ist es oberste Prämisse, in einem theoretischen Teil die neuesten Webstandards sowie PHP spezifische Aktionen und Möglichkeiten zu erläutern und näher zu beschreiben. Dabei soll die Theorie von normalen Sessionvariablen über herkömmliche Verschlüsselungsmethoden bis hin zum Salted-Hash-Verfahren durchbesprochen werden, um die einzelnen Unterschiede und etwaige Probleme der verschiedenen Techniken aufzuzeigen.

Zur Komplettierung der diesbezüglichen Arbeit werden bestimmte, zurechtgelegte Informatikkonzepte erläutert, um diese im Anschluss jeweils mit Beispielen und dazugehörigen Codeauszügen zu untermauern, welche sich auch im Unterricht ideal einsetzen lassen würden. Der ungefähr geschätzte Codeumfang des gesamten Projekts liegt bei ca. 900 Zeilen und umfasst dabei die Programmiersprachen PHP, HTML und CSS. Des Weiteren muss selbstverständlich auch eine Datenbank seitens der Lehrenden zur Verfügung gestellt werden, welche mittels phpMyAdmin im Browser erstellt werden kann. Hierbei ist auch unbedingt auf eine passende Kodierung wie z.B. `utf8_general_ci` zu achten.

Die Ausarbeitungen der Unterrichtskonzepte nehmen dabei allesamt Bezug auf Session basierte Login-Systeme, wodurch sich auch ein großer Pool an Themen aus der Informatik, rund um dieses Projekt ergibt. Es steht nämlich nicht nur die Anwendung neuester Webentwicklungskonzepte im Vordergrund, sondern selbstverständlich auch die sicherheitstechnischen Aspekte, welche bei solchen Projekten stets oberste Priorität haben sollten. Der Schwerpunkt dieser Diplomarbeit wird allerdings dennoch auf der Vermittlung von Informatikkonzepten und deren Einsatz im Informatikunterricht liegen. Die Forschungsfrage „Welche Vor- bzw. Nachteile birgt ein Salted-Hash-Verschlüsselungsverfahren bei Webbrowser generierten Cookies im Vergleich zu herkömmlichen Cookies bzw. Sessionvariablen und inwiefern sind diese verschiedenen Techniken im Informatikunterricht sinnvoll einsetzbar?“ wird mittels der Unterrichtsszenarien, dem darauffolgenden Experteninterview sowie der eigens dafür angefertigten SWOT-Analyse beantwortet und liefert damit einen besseren Über-, sowie Einblick in das Unterrichtskonzept „Kekse ohne Salz schmecken nicht“. Als Zielgruppe sind alle SchülerInnen der Sekundarstufe II sowie auch Studierende des Studiengangs Informatik für die Umsetzung dieses Konzeptes geeignet.

Teil I

Theoretische Grundlagen

Kapitel 2

Standards in der Webentwicklung

In den vergangenen 30 Jahren hat sich das Internet und vor allem dessen Inhalt technisch sowie optisch grundlegend verändert. Etwas, das sich allerdings nicht geändert hat, ist die altertümliche – womöglich durch Papier geprägte – Erwartungshaltung mancher NutzerInnen und WebentwicklerInnen in Bezug auf die Handhabung mit dem Internet bzw. dessen Websites, welche sich durch zwei unterschiedliche Prinzipien erklären lässt. Zum einen wird erhofft, dass AutorInnen die absolute Kontrolle über die Website haben und zum anderen erwarten sie, dass die Website auf allen Endgeräten, darunter auch allen mobilen, gleich aussieht („responsives Design“).³

John Allsopp versuchte in diesem Sinne bereits im Jahr 2000 seine Erfahrungen sowie Prinzipien des Webdesigns in seinem Aufsatz „A Dao of Web Design“ zusammenzufassen, indem er auf das uralte chinesische Weisheitsbuch „Dao De Jing“ Bezug nimmt, welches der Sage nach auf den Gelehrten Lao Tse zurückgeht. Aus diesem Buch, das mit seinen insgesamt 81 harmonischen Versen eine dennoch relativ kryptische Sammlung darstellt, kann man allerdings der folgenden Philosophie aufgrund der oben genannten Problematik sehr viel abgewinnen.⁴

„Akzeptiere, was vor dir liegt, ohne die Situation ändern zu wollen. Wenn wir genau hinschauen, werden wir sehen, dass die Arbeit schneller und leichter vorangeht, wenn wir aufhören, etwas zu wollen.“⁵

Demnach ist der Lehre des Lao Tse zu entnehmen, dass das Web flexibel ist und diese Flexibilität angenommen sowie akzeptiert werden soll, um eindrucksvolle und vor allem mediengerechte Webseiten erstellen zu können. Man muss allerdings dazu sagen, dass der Aufsatz von John Allsopp schon ein bisschen in die Jahre gekommen ist, wenn man bedenkt, dass die Wunderwaffen der Webentwicklung damals noch `<font>` sowie

³vgl. Müller (2013), S.24.

⁴s. Allsopp: A Dao of Web Design, (Stand: 24.09.2016).

⁵Müller (2013), S.24.

<table> hießen. Die Gestaltung von Webseiten mittels CSS stand zu diesem Zeitpunkt gerade erst am Anfang.⁶ Erst durch eine eigene Bewegung im Jahr 1998, welche sich für die Vereinheitlichung und Verbreitung expliziter Webstandards einsetzte, konnte es gelingen, HTML-Tabellen aus den Entwicklerstudios und somit aus Webseitendesigns zu verbannen.⁷

Grundsätzlich unterscheidet man heutzutage in der modernen Webentwicklung nur mehr zwischen „Mobile First“ und „Desktop First“ Webseiten, wobei beide Herangehensweisen Vor- und Nachteile in sich bergen. Im Internet findet man sehr gut als auch sehr schlecht programmierte Musterbeispiele beider Varianten. Wirklich wichtig ist allerdings, dass der Inhalt, auch „Content“ genannt, auf allen Geräten zugänglich ist. Genau aus diesem Grund möchte Müller hier auch den Terminus „Content First“ begründen, da er der Ansicht ist, dass Webseiten nur dazu da sind, um Inhalte an Besucher auszuliefern und somit von Anfang an im Zentrum aller Überlegungen stehen sollten. Ob sich dieser Trend in naher Zukunft noch durchsetzen wird, bleibt allerdings abzuwarten.⁸

„Bis dorthin bleibt der Weg in die Zukunft sehr steinig, da es zur Verbreitung von HTML5 und CSS3 noch einige bekannte, wie auch unbekannte Hürden zu bewältigen gibt. Auf der einen Seite ist der Internet Explorer aus dem Hause Microsoft nach wie vor einer der am meist verbreitetsten Browser weltweit, welcher mit seiner äußerst marginalen Unterstützung von HTML5 und CSS3 bis einschließlich Version 11, der Webstandardverbreitung ziemlich im Weg stand und mit dem neuen Edge-Browser unter Windows 10 nach wie vor steht. Auf der anderen Seite dürfen wir nicht ganz so einfach mit den neuen Technologien, gerade im Medieneinsatz (z.B. von Audio und Video) davongaloppieren und die Barrierefreiheit für Sehbehinderte sowie insbesondere ältere Menschen aus den Augen verlieren.“⁹

Der Internet Explorer ist natürlich nicht der einzige Browser, der in gewisser Art und Weise Probleme macht. Selbst die Varianten von Google, Apple oder Mozilla streiten sich z.B. um die Unterstützung von Audio- und Videocodecs und unterstützen schlichtweg die Formate des anderen nicht.¹⁰ Hoffentlich werden diese Unterschiede – sofern die finanziellen Interessen dieser Unternehmen jemals in den Hintergrund geraten sollten – irgendwann einmal allesamt wegfallen. Doch bis dorthin werden sich die Spezifikationen von HTML5 und CSS3 noch mehrmals weitgehend ändern und mit ihnen die Browserspezifischen HTML5-, sowie CSS3-Implementierungen von Chrome, Safari und Firefox.¹¹ Der Fokus dieses Ka-

⁶vgl. Müller (2013), S.25.

⁷s. The Web Standards Project: Working together for Standards, (Stand: 24.09.2016).

⁸vgl. Müller (2013), S.42.

⁹Hogan (2011), S.15.

¹⁰vgl. ebd., S.18.

¹¹vgl. ebd., S.19.

pitels liegt jedoch auf den aktuellen Spezifikationen der Arbeitswerkzeuge HTML5, CSS3, PHP und MySQL, welche im Nachfolgendem enger zusammengefasst und dabei näher erläutert werden.

2.1 HTML5

Wie bereits festgestellt wurde, handelt es sich bei HTML5¹² um eine reine Spezifikation, welche neue Tags, Markups sowie neue JavaScript APIs mit sich bringt. Dabei geht es keineswegs um einen neuerlichen Internethype oder gar einer Plattform, wie manche WebentwicklerInnen fälschlicherweise glauben, obwohl es das W3C selbst als solches propagiert, wie man in der folgenden Abbildung (*Abb. 2.1.1*) deutlich erkennen kann. Leider muss man diesen Sachverhalt – unter anderem deswegen – des öfteren klarstellen, da rund um diese Definition mit der Zeit so viel Verwirrung entstanden ist, dass mittlerweile KundInnen aber auch manche EntwicklerInnen CSS3-Spezifikationen wie Schatten, Verläufe und Transformationen plötzlich als HTML bezeichnen.¹³

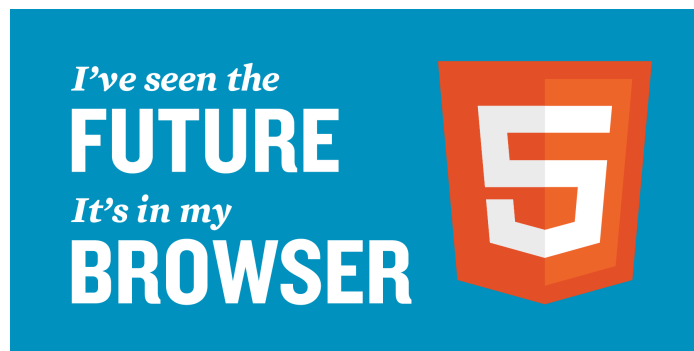


Abbildung 2.1.1: Offizieller HTML5-Sticker inkl. Logo des W3C

Semantik spielt in diesem Kontext eine wirklich große Rolle und begründet im Wesentlichen auch die fundamentale Basis von HTML5. Denn genau dieses Verlangen nach einer besseren Auszeichnung von semantischen Inhalten, wie z.B. durch Suchmaschinen, barrierefreien oder auch normalen, mobilen Webseiten, hat die Entwicklung von HTML5 seit 2007 bis heute enorm vorangetrieben. Ein weiteres großes Ziel von HTML5 besteht darin, Audio-, Video- und sonstige animierte Inhalte ohne jegliche Plug-Ins darzustellen, um auf lange Distanz Third-Party Applikationen wie z.B. den Adobe Flash Player auszumerzen. Dieser wird zwar, aufgrund erheblicher Sicherheitslücken, kaum noch auf Mobilgeräten verwendet. Allerdings würden mobile Endgeräte durch die Nutzung mehrerer Plug-Ins auch zu viele Ressourcen in Form von Akkulaufzeit oder gar Prozessorleistung einbüßen, wodurch sich die Ersparnis solcher Plug-Ins selbstverständlich bezahlt macht.¹⁴

¹²s. W3C: HTML5 Recommendation 2014, (Stand: 24.09.2016).

¹³vgl. Hogan (2011), S.4.

¹⁴vgl. Potschien (2013), S.11.

Falls man noch alte HTML4 Dokumente oder dergleichen finden sollte, kann man ebenfalls völlig beruhigt bleiben, da die meisten bisherigen Spezifikationen aus vorherigen HTML Versionen selbstverständlich beibehalten und sogar stellenweise optimiert wurden, so dass der Umstiegsaufwand auf HTML5 relativ gering ist. So können beispielsweise alte `<b>` (d.h. fetter Text) oder `<i>` (d.h. kursiver Text) Elemente genauso zu einem validen HTML5 Dokument führen, wie das bereits unter HTML4 üblich war. Das gleiche gilt natürlich auch für die Auszeichnung von Texten, Tabellen, Überschriften, Bilder und Listen. Bei der Entwicklung selbst sind viele alte Elemente, welche keine semantische Bedeutung hatten sondern ausschließlich Gestaltungszwecken dienten, übernommen worden. Eigentlich ziemlich kontraproduktiv, wenn man bedenkt, dass seit der Einführung von CSS 1994 der Usus geprägt wurde, Syntax von Design zu trennen. Dass diese alten Elemente allerdings dennoch überlebt und sogar eine passende Semantik bekommen haben, liegt auch einzig und alleine am Anpassungsaufwand der Auszeichnungen.¹⁵

HTML5 bietet außerdem einen praktischen und vor allem sicheren Workaround gegen die Browserseitigen „Cross-Site Scripting“ Einschränkungen, bei denen verhindert wird, dass man Skripte einer bestimmten Domäne dazu verwenden kann, Skripten einer anderen Domäne zu ändern bzw. auszuführen. Auch eine Unterstützung von sogenannten Websockets wird mittlerweile angeboten, bei denen eine dauerhafte Verbindung zum Server gehalten werden kann, ohne ständig vom Backend die Fortschrittsaktualisierungen abzufragen. Webstorage und WebSQL Datenbanken, welche Daten dauerhaft am clientseitigen Rechner abspeichern, sind dank der neuen Spezifikation absolut kein Tabuthema mehr. Die wohl wichtigste Neuerung stellt aber die verbesserte Barrierefreiheit z.B. bei Screenreadern dar, welche etwa für Personen mit Sehbehinderung Webseiteninhalte verarbeiten und anschließend vorlesen bzw. auf ertastbare Medien umleiten.¹⁶ Da solche Screenreader allerdings nicht mit den schnelllebigen Webbrowser mitkommen, existieren zusätzlich auch noch sogenannte Webreader, welche für Webbrowser optimiert sind.¹⁷ Dies wird durch die bessere Auszeichnung sowie durch neue Elementattribute ermöglicht, in denen sich Funktionen nun bestimmten Elementen zuordnen lassen, sodass Bildschirmlesegeräte diese leichter verarbeiten können.¹⁸ Auch das W3C hat sich dieser Thematik mittels der WCAG-Spezifikation angenähert, welche bis heute gültige Richtlinien für barrierefreie Webinhalte zusammenfasst.¹⁹

Blickt man zurück, liegen die eigentlichen Wurzeln von HTML5 in einem WHATWG-Entwurf aus dem Jahr 2005 mit dem Titel „Web Applications 1.0“, welcher die Richtung vorgab, das Web für zukünftige Webanwendungen fit zu machen. So sollte Software wie Textver-

¹⁵vgl. *Potschien* (2013), S.13.

¹⁶vgl. *Hagelkruys* (2012), S.48.

¹⁷vgl. *ebd.*, S.49 f.

¹⁸vgl. *Hogan* (2011), S.10 ff.

¹⁹s. W3C: Web Content Accessibility Guidelines (WCAG) 2.0 2009, (Stand: 24.09.2016).

arbeitung, Tabellenkalkulation, Präsentationstechnik, Bild- und Videobearbeitung und vor allem Computerspiele vom lokalen Speicherort in den Browser abwandern und das alles ganz ohne Flash bzw. Java. Aus diesem Grund wurden in dieser Diplomarbeit auch alle selbst erstellten Grafiken mittels einer Webapplikation („draw.io“)²⁰ gefertigt, um demonstrieren zu können, was zum aktuellen Stand bereits alles möglich ist. Als Musterbeispiel der Vorreiterrolle wäre hier dennoch Google mit seinen Office Anwendungen wie z.B. „Google Drive“ bzw. „Google Docs“ zu erwähnen. Bis allerdings weitere Softwareangebote im Browser funktionieren werden, müssen diese beispielsweise noch lernen, binäre Daten wie Bilder und Videos zu laden und zu bearbeiten, Programmstände offline abzuspeichern sowie sich an lokale Datenschutzgesetze zu halten, weswegen JavaScript in Zukunft eine noch wichtigere Rolle spielen wird als wie bisher.²¹

Der selbsternannte „HTML5-Erklärbar“ Peter Kröner, hat in seinem privaten Blog versucht, eine Landkarte des sogenannten „HTML5-Universums“ (*Abb. 2.1.2*) zu zeichnen, um ein für alle Mal klarzustellen, wie sich HTML5 nun definiert und was genau darüber hinausgeht.²²

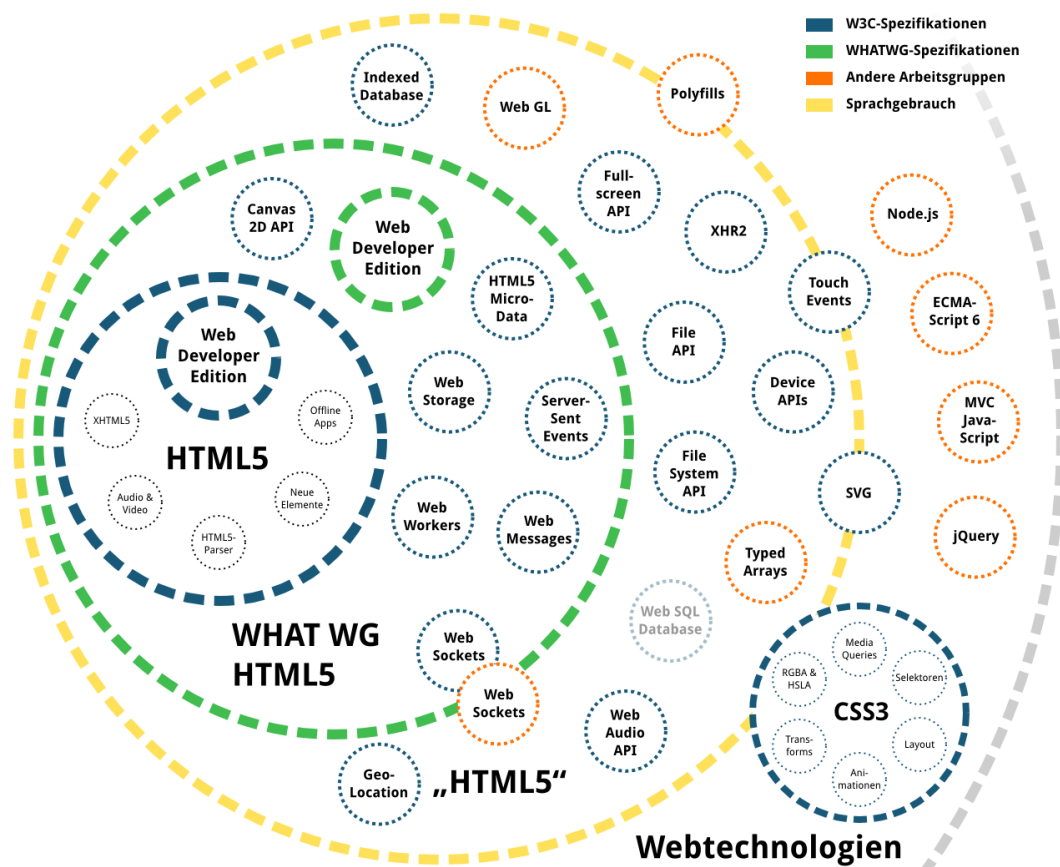


Abbildung 2.1.2: Die Karte des „HTML5-Universums“ nach Kröner

²⁰s. *Draw.io*: Flowchart Maker & Online Diagram Software, (Stand: 24.09.2016).

²¹ vgl. Müller (2013), S.62 f.

²²vgl. *ebd.*, S.63.

Der zentrale blaue Kreis links in der Mitte heißt „HTML5“ und stellt die HTML5-Spezifikation des W3C²³ dar. Die Spezifikation ist sehr umfangreich, da sie alle Details für die BrowserherstellerInnen enthält, die die Features implementieren müssen. Deshalb gibt es noch eine kompaktere Edition für WebentwicklerInnen. Der etwas größere grüne Kreis ist „HTML – The Living Standard“ von der WHATWG, der ebenfalls in einer Webdeveloper Edition existiert. Er umschließt die W3C-Spezifikation, enthält aber noch eine Menge mehr. Alles außerhalb dieser beiden Kreise ist nur noch unter Anführungsstrichen als HTML5 zu bezeichnen und sozusagen Teil des erweiterten Universums. Der kleine Kreis namens „Geolocation“ z.B. gehört weder zur W3C noch zur WHATWG, wird aber interessanterweise fast überall als Feature von HTML5 bezeichnet.²⁴ Selbiges gilt auch für Node.js oder gar jQuery, aber man sieht sehr deutlich anhand der vielen kleinen orangen Kreise, dass das Web mittlerweile sehr viel Neues zu bieten hat.²⁵

Zusammenfassend kann man also sagen, dass der Fokus bei HTML5 ganz eindeutig auf Webanwendungen liegt, der sich unter anderem folgenden zentralen Themen widmet:

- Neue Elemente für Struktur und Semantik von Webseiten wie `<header>`, `<nav>`, `<footer>`, `<main>`, `<aside>`, `<section>` und `<article>`
- Verbesserte Barrierefreiheit für Screenreader usw.
- Neue Formularelemente wie Suche, E-Mail usw.
- Multimedia Elemente und APIs für Video und Audio ohne Fremdtechnologien
- Canvas (per JavaScript programmierbare Bitmap-Grafiken für Animationen, Diagramme usw.)
- Offline-Webanwendungen, die auch ohne Internetverbindung funktionieren
- Visuelle Effekte sowie fortschrittliche Selektoren, dank der CSS3 Erweiterung²⁶

2.1.1 Metadaten

Wer glaubt, dass es von HTML5 mehrere Versionen gibt, der irrt gewaltig. Von HTML5 gibt es nämlich nur eine einzige Version, bei der man praktischerweise keinerlei Referenzen mehr auf eine bestimmte DTD angeben muss. So war es bisher nämlich vorgesehen, über den DOCTYPE die verwendete HTML Version einschließlich der verwendeten Variante (z.B. "transitional", "framset", oder "strict") anzugeben. In HTML5 wird nur mehr sehr platzsparend angegeben, dass es sich dabei um ein HTML Dokument handelt, wodurch jede HTML Vorgängerversion automatisch aufgrund der neuen und kurzen Schreibweise umgangen wird.²⁷ Die vereinfachte DOCTYPE-Deklaration (Abb. 2.1.3), welche original nur mehr aus `<!DOCTYPE html>` besteht, ist nämlich ebenfalls problemlos

²³ s. W3C: HTML5 Recommendation 2014, (Stand: 25.09.2016).

²⁴ vgl. Müller (2013), S.63 f.

²⁵ s. Kröner: Die Karte des HTML5-Universums, (Stand: 25.09.2016).

²⁶ vgl. Müller (2013), S.65.

²⁷ vgl. Potschien (2013), S.14 f.

dazu in der Lage, Dokumente im Browser standardkonform darzustellen. Selbst in XML geschriebene HTML5 Dokumente benötigen keine DOCTYPE-Angabe mehr. Stattdessen können sie eine XML-Deklaration erhalten, bei dem der MIME-Typ "application/xhtml+xml" bzw. "application/xml" verwendet werden sollte.²⁸

```

1  <!DOCTYPE html>
2  <!-- optionale Kommentare -->
3  <html lang="de">
4    <head>
5      <meta name="author" content="Simon Marik">
6      <meta name="description" content="Thesis Projekt">
7      <meta name="viewport" content="width=device-width,
      initial-scale=1.0, user-scalable=no">
8      <title>Thesis Projekt</title>
9      <link rel="stylesheet" type="text/css"
      href="css/thesis.css">
10     </head>
11     <body>Webseiten Inhalt...</body>
12  </html>

```

Abbildung 2.1.3: Minimalbeispiel der neuen HTML5 DOCTYPE-Deklaration

Wenn man bedenkt, dass die HTML4 DOCTYPE-Deklaration damals aus einer schier unendlich langen Zeichenkette bestand, die man nur in den seltensten Fällen auswendig wusste und sich meistens von irgendwo anders kopierte, war es schon äußerst sinnvoll, an dieser bei der Spezifikationsaktualisierung einzusparen. Dabei bleibt es auch vollkommen egal, ob man `<!DOCTYPE html>` oder `<!doctype html>` schreibt, was für viele, vor allem für englischsprachige NutzerInnen, extrem komfortabel sein sollte. Interessant daran ist, dass früher selbst die Browser diese endlos lange Zeichenkette nicht einmal gelesen haben. Es war und ist nach wie vor nur wichtig, dass der DOCTYPE in der ersten Zeile steht, um den Browser in den Standardmodus zu versetzen. In der zweiten Zeile geht es dann direkt mit dem Stammelement `<html>` weiter, welches dem Browser den Dokumententyp mitteilt. Sollte die Standardsprache der zu erstellenden Website nicht Englisch sein, so schreibt man z.B. für eine deutsche Seite `<html lang="de">`.²⁹

Die Angabe des verwendeten Zeichensatzes erfolgt bei HTML5 nicht mehr über die Angabe des "http-equiv" Attributs, sondern wird über das neue "charset" Attribut innerhalb eines `<meta>` Elements angegeben. Bei der Angabe von "http-equiv", musste man früher nämlich auch noch den MIME-Typ angeben, wodurch der Eintrag erheblich länger wurde.³⁰ Als Zeichensatz ist es empfehlenswert UTF-8 zu verwenden,

²⁸vgl. *Niederst Robbins* (2014), S.7.

²⁹vgl. *Müller* (2013), S.72.

³⁰vgl. *Potschien* (2013), S.15.

da es eigentlich – außer bei alten Datenbanken mit ISO Zeichensatz – keine guten Gründe gibt, diesen nicht zu verwenden (Default = UTF-8). Der Seitentitel `<title>` sowie die Seitenbeschreibung `<meta name="description" content="...">` haben sich seit damals nicht verändert. Diese beiden Elemente beschreiben unsichtbar den Inhalt einer Seite und sind vor allem für Suchmaschinen wichtig. Man kann das etwa mit einem Eintrag bei Google vergleichen. Der Seitentitel repräsentiert dabei die Überschrift, welche kurz und bündig sein sollte und die Seitenbeschreibung in max. 150 Zeichen die zweizeilige Beschreibung unterhalb der grünen URL abdeckt. Auch die Angabe der AutorIn `<meta name="author" content="...">` ist bei der Umstellung weitgehend gleich geblieben. Einzig und allein das Element für den Viewport namens "meta-viewport" kam neu in der `<meta>`-Welt dazu, obwohl es eigentlich nur sehr wenig mit HTML5 zu tun hat. Dieses Element ist nämlich eine Erfindung von Apple und hat sich mit der Zeit zu einer Art „Semistandard“ weiterentwickelt. Als im Jahr 2007 das iPhone auf den Markt kam, gab es noch keine für Smartphones optimierten Webseiten, wodurch die iOS-Entwickler gezwungen waren, Webseiten in mobilen Browsern soweit zu verkleinern, dass sie vollständig auf die kleinen Displays passten. Dabei wurden Seiten auf die gedachte Breite von 980 Pixel (beim iPhone) bzw. 800 Pixel (bei Android) gerendert und anschließend auf die tatsächliche Displaybreite von 320 Pixel heruntergezoomt. Diese Technik findet heutzutage nach wie vor bei so ziemlich allen mobilen Webbrowsern auf fast allen mobilen Betriebssystemen Anwendung, außer das CSS der Website wurde durch Media Queries optimiert. Der optionale Content des Viewport Elements "width=device-width, initial-scale=1.0" definiert die Gerätebreite als eigentliches Breitenmaß und verweigert letzten Endes den BenutzerInnen das Verkleinern der Website. Abschließend lässt sich noch sagen, dass man bei „Mobile First“ Webseiten immer den Meta-Viewport verbauen sollte, ihn aber bei „Desktop First“ Varianten getrost weglassen kann.³¹

2.1.2 Tags

Im direkten Zusammenhang mit HTML5 steht, wie vorhin bereits erwähnt, die sogenannte Semantik, welche semantische Elemente für die Auszeichnung des Seiteninhalts verantwortlich sind. Als sich im Jahr 2004 die WHATWG zusammenfand, kamen CSS-basierte Layouts gerade erst richtig in Schwung und mit ihnen verbreitete sich auch das sogenannte `<div>`-Element. Erweitert um diverse IDs wie `<div id="container">` oder Klassen wie `<div class="nav">`, war es eine unglaubliche Bereicherung für alle WebseitenautorInnen, ihre Seiten zu strukturieren und per CSS zu stylen. In HTML5 ist das `<div>`-Element ebenfalls kein Tabu, allerdings gibt es eine Reihe neuer semantischer Elemente (Tab. 2.1.1), welche in vielen Situationen oft die bessere Wahl sind.³²

³¹vgl. Müller (2013), S.73 ff.

³²vgl. ebd., S.76 f.

Neue HTML5-Elemente:		
<article>	<figcaption>	<output>
<aside>	<figure>	<progress>
<audio>	<footer>	<rp>
<bdi>	<header>	<rt>
<canvas>	<hgroup>*	<ruby>
<command>*	<keygen>	<section>
<data>**	<main>**	<source>
<datalist>	<mark>	<time>
<details>	<menuitem>**	<track>
<dialog>**	<meter>	<video>
<embed>	<nav>	<wbr>

* Nicht mehr Teil von HTML5

** Nur WHATWG und HTML5

Tabelle 2.1.1: Ein Auszug der neu unterstützten HTML5-Elemente ³³

Selbstverständlich kann man nach wie vor anstatt einer `<h1>`-Überschrift ein `<div>`-Element mit selbigen Layout basteln. Die normalen BenutzerInnen würde das im Quelltext sicherlich kaum stören, aber Suchmaschinen würden das `<div>`-Element keinesfalls als `<h1>`-Überschrift erkennen. Die neue Semantik hilft also, wie man sieht, vor allem Suchmaschinen, Screenreadern, Browsern aber auch so manche EntwicklerIn. Vor allem bei Layoutbereichen wie der Navigationsleiste oder der Fußzeile muss man nicht mehr auf zahlreiche `<div>`-Elemente mit unzähligen Klassen oder IDs zurückgreifen, sondern kann sich einfach aus der neuen HTML5 Tagsammlung bedienen (Abb. 2.1.4). Ähnlich wie beim Überschriftenbeispiel haben die neuen Elemente innerhalb des Quelltexts auch ihre eigene Bedeutung, wodurch Suchmaschinen beispielsweise in der Lage sind, Suchbegriffe im Inhaltsbereich `<main>` höher zu bewerten als in Bereichen wie z.B. `<nav>` oder `<footer>`.³⁴

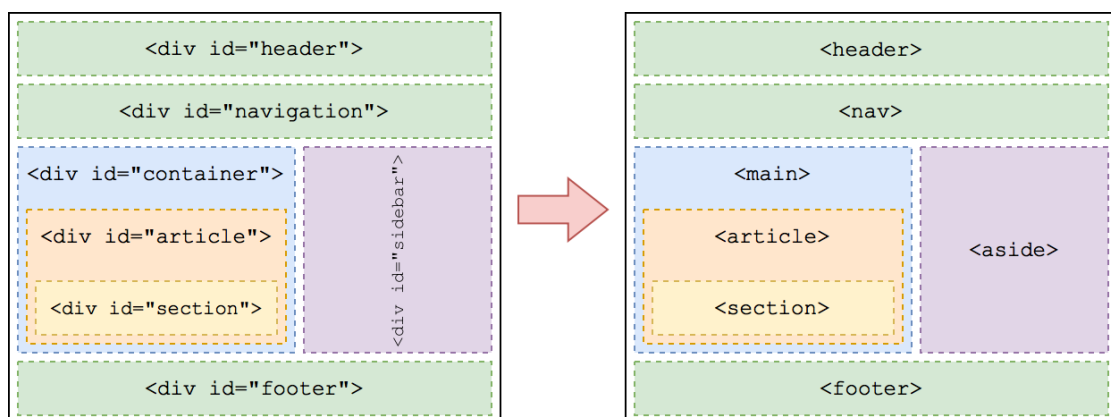


Abbildung 2.1.4: Auszeichnung der Layoutbereiche von HTML4 nach HTML5

³³ Niederst Robbins (2014), S.4.

³⁴ vgl. Müller (2013), S.79 ff.

Mit der neuen Spezifikation wird auch eine Reihe neuer Eingabetypen für Inputelemente in Formularen definiert, welche innerhalb des `<input>`-Elements als Wert für das "type" Attribut angegeben werden. Dabei sind Werte wie "color", "date", "datetime", "time", "email", "week", "month", "number", "range", "search" oder "tel" möglich. Um auch der wachsenden Nachfrage nach interaktiven Webinhalten nachzukommen, wurden in HTML5 viele neue APIs eingeführt, welche Tasks proprietärer Plugins standardisieren. Für einige dieser APIs existieren sogar eigene Markup-Bausteine wie beispielsweise für audio, video und canvas. Andere verwenden schlichtweg JavaScript oder serverseitige Komponenten. Folgende APIs sind beispielsweise unter anderem Teil der HTML5-Spezifikation des W3C:

- Media API zum Abspielen von Video- und Audiodateien inkl. Multimedia-Synchronisierung.
- TextTrack zum Einbinden von zeitgesteuerten Untertiteln bei Videoelementen
- Session History API für die Arbeit mit dem Browserverlauf
- Editing API (enthält das neue globale Attribut "contenteditable")
- Drag and Drop API (enthält das neue Attribut "draggable")
- Web Workers API (ermöglicht das Ausführen von Skripts im Hintergrund)
- File API (ermöglicht den Zugriff auf per Formular hochgeladene Dateien)³⁵

Es wurden aber auch im Vergleich zu HTML4 einige gebräuchliche Elemente wieder abgeschafft. Einige der Tags wie z.B. `<basefont>`, `<big>`, `<center>`, `<font>`, `<s>`, `<strike>`, `<tt>` oder `<u>` sind bereits extrem veraltet und werden unter HTML5 nicht mehr unterstützt. Es finden sich im WWW aber nach wie vor viele Seiten, welche mit visuellen Editoren wie Adobe Dreamweaver editiert werden und somit noch immer eine Menge an `<font>` und `<center>` Tags enthalten. Selbst die Unterstützung von Frames, welche bei sogenannten Enterprise-Webanwendungen (z.B. Microsoft Outlook Web Access) immer sehr beliebt waren, wurden restlos entfernt. Dies geschah aber aus gutem Grund, da Frames jeher extreme Schwierigkeiten bei der Benutzerfreundlichkeit und auch bei der Barrierefreiheit verursachten und somit einfach nicht tragbar waren. Mit den Frames verschwanden selbstverständlich auch die Elemente `<frame>`, `<frameset>` sowie `<noframes>` und gestalterische Frametools wie `<acronym>` wurden z.B. durch bessere Optionen wie `<abbr>` ersetzt. So viele Attribute wie auch in der neuen Spezifikation dazugekommen sind, von so vielen hat man sich auch beim Umstieg zur neuen Version wiederum verabschiedet. So gehören Attribute wie beispielsweise "align", "link", "bgcolor", "height", "width", "scrolling", "hspace", "vspace", "valign", "cellpadding", "cellspacing" oder "border" mittlerweile der Vergangenheit an. Aber auch ein paar wichtige Tags wie "target", "profile" oder "longdesc" mussten leider den Weg für HTML5 ebnen und räumen. Viele Entwickler rebellierten bei

³⁵vgl. *Niederst Robbins* (2014), S.5 f.

"longdesc", da dieses Attribut die einzige Möglichkeit darstellte, BenutzerInnen mit Bildschirmlesegeräten zusätzliche beschreibende Informationen anzubieten.³⁶

Josh Duck hat in seinem privaten Blog – um wieder etwas Licht in das Dickicht der Tags zu bringen – ein komplettes Periodensystem nach der aktuellen HTML5-Spezifikation des W3C angefertigt (Abb. 2.1.5), aus dem alle funktionierenden Elemente nun endlich eindeutig hervorgehen sollten.³⁷

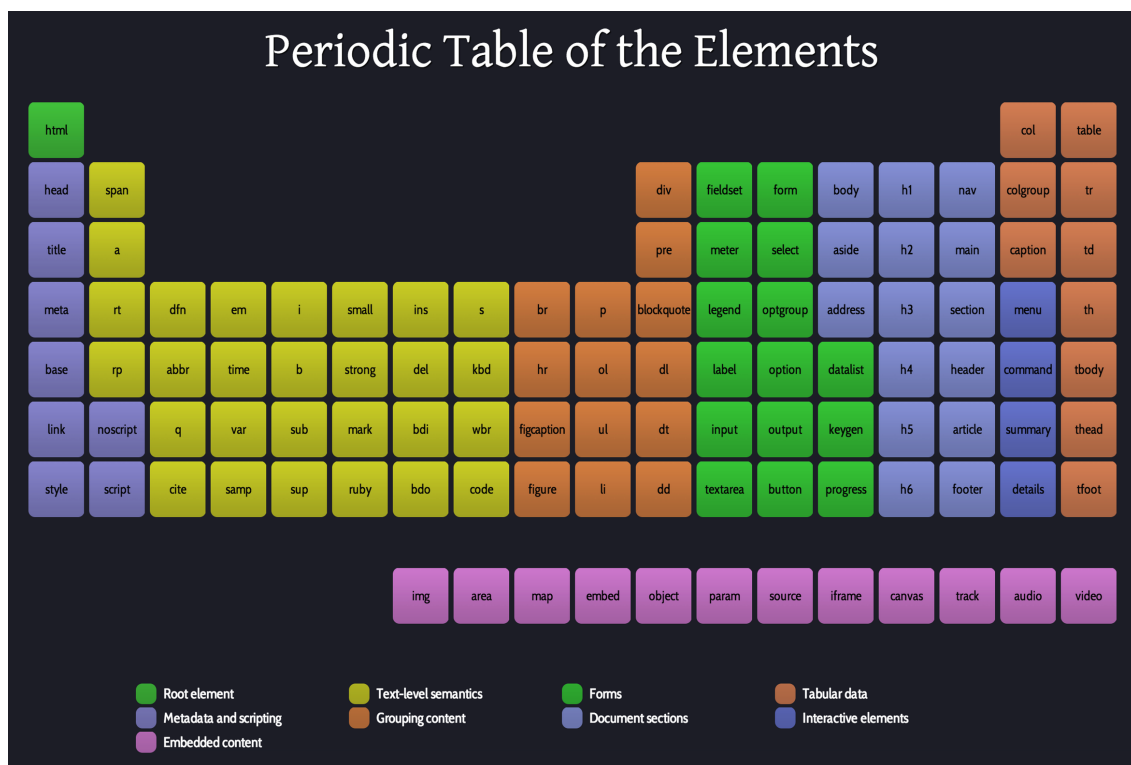


Abbildung 2.1.5: Das Periodensystem der HTML5-Elemente nach Duck

2.2 CSS3

CSS stellt eine Ergänzung zu HTML dar, welche vom W3C als eigener Standard innerhalb des HTML-Universums definiert und somit bereits auch propagiert wurde (Abb. 2.2.1). Der CSS-Standard dient einerseits dazu, BrowserherstellerInnen genauestens anzugeben, wie eine CSS-Eigenschaft dargestellt werden soll, dient aber auch andererseits als zuverlässiges Referenzmaterial für WebentwicklerInnen. In einer solchen Referenz kann z.B. nachgelesen werden, mit welchen CSS-Eigenschaften was gesteuert werden kann, welche möglichen Parameter dabei eingesetzt werden können und welcher Effekt damit schlussendlich erzielt werden kann. Es ist interessant, dass CSS2.1 erst seit Juni 2011

³⁶vgl. Hogan (2011), S.16 ff.

³⁷s. Mozilla Developer Network: Liste der HTML5-Elemente, (Stand: 25.09.2016).

vom W3C offiziell empfohlen wurde, obwohl dieser Standard bereits über Jahre hinweg erfolgreich in der Praxis Anwendung fand und CSS3 seit dem Jahr 2000 in Entwicklung war. Angesichts dessen wird es sicherlich noch einige Zeit bis zum Release von CSS3 dauern, obwohl eigentlich nur auf bereits Bestehendem aufgebaut wird.³⁸ CSS2.1 stellt nämlich die fundamentale Basis von CSS3 dar, was bedeuten soll, dass CSS3 – wenn es offiziell erscheint – rund alle 115 Eigenschaften des 2.1 Standards umfassen und dabei sogar auf ca. 300 Eigenschaften im eigenen Repertoire aufstocken wird. Für die Grundlagen bedeutet das, dass CSS3 nicht anders aufgebaut ist als die Version 2.1 des Standards und daher – ganz im Gegensatz zu unterschiedlichen HTML Versionen – die Schreibweise komplett ident geblieben ist.³⁹

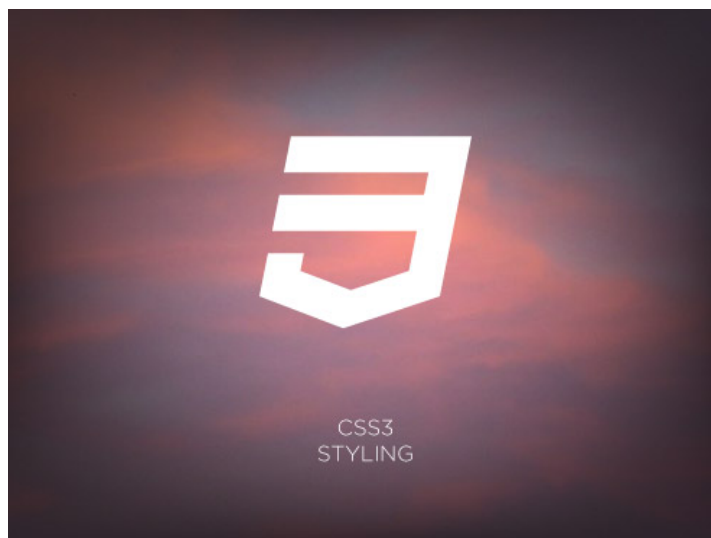


Abbildung 2.2.1: Offizielles CSS3-Klassenbild inkl. Logo des W3C

Wie man sieht ist CSS3 ähnlich wie HTML5 ebenfalls noch nicht abgeschlossen. Es gibt selbst noch einige Features welche zwar in CSS3 vorgesehen sind, die allerdings noch von keinem einzigen Browser unterstützt werden. Die meisten Eigenschaften sind bereits in allen gängigen Browsern implementiert – teilweise sogar bereits in mehreren Versionen. Da sich solche Eigenschaften bereits etabliert haben, rät das W3C auch zur Verwendung dieser, da sie ältere Browser einfach ignorieren würden. Allerdings geht jeder Browser gerade bei der Einführung neuer Eigenschaften seinen eigenen Weg. Dadurch kann es beispielsweise passieren, dass sich Webseiten in unterschiedlichen Browsern komplett anders verhalten bzw. anders aussehen. Aus diesem Grund hat man die sogenannten „Vendor-Präfixe“ eingeführt, wodurch jede Browser-Engine sein eigenes Präfix inkl. eigener Eigenschaft hat, durch die es möglich ist, Eigenschaften für jeden Browser individuell einzusetzen. Selbstverständlich wird man bei den bereits etablierten

³⁸s. W3C: CSS Current Status, (Stand: 26.09.2016).

³⁹vgl. Heller (2012), S.234 f.

CSS-Eigenschaften keine inkorrekte Darstellung bzw. Fehlverhalten mehr vorfinden, aber aufgrund der zahlreichen Nutzung älterer Browser zählt es sich nach wie vor aus – genauso wie bei JavaScript Eigenschaften, die Vendor-Präfixe der eigentlichen Eigenschaft voran zu stellen.⁴⁰ Um dies abschließend besser veranschaulichen zu können, listet die folgende Abbildung (Abb. 2.2.2) alle gängigen Vendor-Präfixe der Browser Safari & Chrome (`-webkit`), Firefox & Linux-Browser (`-moz`), Internet Explorer & Edge (`-ms`) und Opera (`-o`) anhand der Eigenschaft `border-radius` auf.

```
1  .testclass {  
2      -webkit-border-radius: 0.3em;  
3      -moz-border-radius: 0.3em;  
4      -ms-morder-radius: 0.3em;  
5      -o-border-radius: 0.3em;  
6      border-radius: 0.3em;  
7  }
```

Abbildung 2.2.2: Alle gängigen Vendor-Präfixe für die CSS-Eigenschaft `border-radius`

Zu den Neuerungen von CSS3 zählen unter anderem die vielen neuen, vorhin bereits erwähnten Eigenschaften, sowie die extremen funktionellen Erweiterungen. Während in CSS2.1 noch Farben, Abstände, Schrifteigenschaften und weitgehend statisches, zweidimensionales Aussehen dominierte, liefert CSS3 inzwischen Eigenschaften, mit denen HTML-Elemente dynamisch bewegt bzw. animiert werden können. So könnte man meinen, dass all das, was prinzipiell einst JavaScript vorbehalten war, sich nun auf den Funktionsumfang von CSS3 verlagerte.⁴¹ Der neue Standard führte aber auch komplett neue Features wie Sprachkonstrukte (z.B. Media Queries), verbesserte Selektoren, sogenannte `@font-face` Regeln, sowie ein komplett überarbeitetes Box-Modell (Abb. 2.2.3) ein, welches von innen nach außen die `content-box`, `padding-box`, `border-box` und die `margin-box` abdeckt.⁴²

⁴⁰vgl. Potschien (2013), S.125 f.

⁴¹vgl. Heller (2012), S.235.

⁴²s. SelfHTML Wiki: CSS/Box-Modell, (Stand: 26.09.2016).

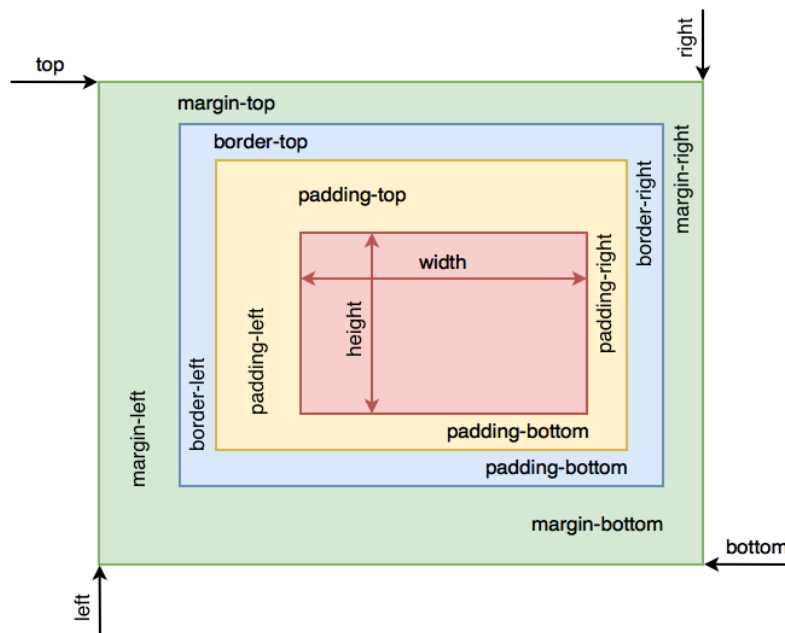


Abbildung 2.2.3: Neu konzipiertes Box-Modell des CSS3-Standards

Natürlich sind das nicht alle Neuerungen des Standards. Es gibt noch unzählige weitere Eigenschaften, welche CSS3 auszeichnen und einer eigenen Erklärung bedürfen würden. Um das Ganze aber nicht unnötig ausschweifen zu lassen, stellt die folgende Aufzählungsliste die wohl wichtigsten neuen Möglichkeiten in der Eigenschaftswelt von CSS3 dar:

- Abgerundete Ecken
- Schatten
- Transparenz mit RGBA (Rot, Grün, Blau und Alpha - ein Format um Farbinformationen zu definieren, mischt die drei Grundfarben entsprechend von Dezimalzahlwerten zwischen 0 und 256.)
- Transforms (zwei- oder dreidimensionale Transformationen von HTML-Elementen)
- Transitions (Animation von HTML-Elementen)
- Animations (Transitions kennen nur einen Start- und einen Stop-Zustand des Objektes. Für komplexere Animationen wird eine sogenannte „Keyframe-Animation“ benötigt, die mehrere Zustände des Elements im zeitlichen Verlauf abbilden kann.)⁴³

2.2.1 Responsive Web Design

Ursprünglich angedachte „960-Grid-Frameworks“ verloren in der Webentwicklung zunehmend an Bedeutung, als im Jahre 2007 das iPhone von Apple präsentiert wurde. Durch das iPhone, sowie drei Jahre später durch das iPad, aber auch durch die große Vielfalt anderer verschiedener Gerätegrößen am Markt, entstanden völlig neue Wege der

⁴³vgl. Ebner (2013), S.5 ff.

mobilen Internetnutzung, womit auch ein Paradigmenwechsel einherging. Die Annahme, dass Besucher von Webseiten vor einem Schreibtisch bei einem Computer sitzen und die Seite mit einer Maus und Tastatur bedienen, schwindet nach und nach immer mehr. WebentwicklerInnen können aus diesem Grund heute nicht mehr sicher sein, auf welchen Endgeräten ihr Quelltext angezeigt wird. Dabei können die Bildschirmdiagonalen zwischen 3 Zoll und 50 Zoll oder auch mehr variieren, wenn man bedenkt, dass ein modernes Gerätespektrum vom kleinen Smartphone bis zum großen Fernseher alles abdeckt. Die französische Grafik- und Webdesignerin Stéphanie Walter griff das einstige Zitat des Kampfkünstlers und Schauspielers Bruce Lee – *“Empty your mind, be formless, shapeless – like water. [...] Be water, my friend.”*⁴⁴ – auf, um die erforderliche harmonische Anpassungsfähigkeit einer jeden Webentwicklerin in der heutigen Zeit zu demonstrieren (Abb. 2.2.4). Aber ganz abgesehen von den Geräten hat sich auch das Umfeld bzw. die Umgebung geändert, in welcher Menschen heutzutage Webseiten aufrufen. So surft man beispielsweise bereits in der Küche, im Bad, im Schlafzimmer oder einfach ganz bequem auf dem Sofa. So gesehen besteht die einzige Sicherheit darin, dass der Quelltext von einem Browser gerendert wird. Alles andere sind bloß Annahmen.⁴⁵



Abbildung 2.2.4: Inhalt ist wie Wasser (i.e. “Content is like water”) nach Walter

Der Autor Ethan Marcotte beschrieb bereits im März 2009 die Vorteile sogenannter „fluid grids“, welche allerdings erst im Mai 2010⁴⁶ unter dem verbesserten Titel „Responsive Webdesign“, sowie unter der Ergänzung von Media Queries und flexiblen Elementen zu einer absolut bahnbrechenden Vision innerhalb der Webentwicklung wurden. Dabei beschrieb er drei verschiedene Techniken, mit denen seiner Meinung nach mediengerechte

⁴⁴ s. Wikiquote: Bruce Lee - Quotes, (Stand: 27.09.2016).

⁴⁵ vgl. Müller (2013), S.28 f.

⁴⁶ vgl. Kadlec (2013), S.13.

Webseiten gebaut werden sollten und responsives Webdesign ermöglichen:

- Prozentbasierte Gridlayouts („fluid grids“)
- Flexible Bilder (z.B. SVG Grafiken)
- Media Queries⁴⁷

„Das ist unser Weg nach vorne. Statt voneinander unabhängige Designs auf eine ständig wachsende Zahl von Geräten zuzuschneiden, können wir sie als unterschiedliche Facetten derselben User-Experience behandeln. Wir können für eine optimale Darstellung gestalten, aber gleichzeitig standardbasierte Technologien in unsere Designs einbetten, damit sie nicht nur flexibler werden, sondern sich auch besser an das jeweilige Medium anpassen, auf dem sie dargestellt werden.“⁴⁸

Aufgrund solcher Artikel wie jener von Marcotte, aber auch durch Bücher und Publikationen anderer AutorInnen wurden die traditionellen Grundfeste des Webdesigns des Öfteren erschüttert, wodurch sich auch in weiterer Folge viel Verwirrung darum bildete. Die Grundidee des Responsive Webdesigns ist es, den HTML-Quellcode mittels Media Queries unterschiedlich innerhalb des CSS zu steuern. Marcotte verwendete nämlich, ganz im Gegensatz zu vielen anderen nach ihm, ein prozentbasiertes Raster für sein Layout, wodurch sich mit der Zeit eine Unterscheidung zwischen „responsive layouts“ und „adaptive layouts“ herauskristallisierte. Während das responsive Layout alle drei von Marcotte genannten Komponenten enthält, verzichtet das adaptive Layout auf schwer umzusetzende fluide Raster und arbeitet anstatt dessen mit festen Breiten innerhalb der Media Queries. Man kann aber getrost sagen, dass beide Methoden gleichberechtigte Formen des responsiven Webdesigns darstellen und somit original zwei unterschiedliche Wege zum selben Ziel sind.⁴⁹

Abschließend sollen nun die wichtigsten Layoutoptionen angeführt werden, welche für die moderne Webentwicklung absolut essentiell, und in beinahe jedem professionell entwickelten Webprojekt vorzufinden sind:

- Feste Breite (Breite der Website wird durch ein Pixelmaß bestimmt - meistens 960 Pixel wegen der Primfaktorzerlegung: $960=2^6 \cdot 3 \cdot 5$)
- Fluid Layouts (Breite der Website wird durch eine Prozentangabe bestimmt - bei weitem flexibler z.B. 60% des Containers)
- Elastische Layouts (sind Fluid Layouts sehr ähnlich, jedoch wird die Begrenzung über die Schriftgröße „em“ vorgenommen - 1em entspricht dabei der Höhe der aktuellen Schriftgröße)

⁴⁷vgl. Müller (2013), S.30 f.

⁴⁸s. Marcotte: Responsive Web Design, (Stand: 27.09.2016).

⁴⁹vgl. Müller (2013), S.32 ff.

- Hybride Layouts (eine Kombination aus fluiden und elastischen Layouts)
- Schriftgrößen:
 - Pixel (zwar die bevorzugte Wahl bei jeder Webentwicklerin, haben aber den Nachteil, dass sie nicht kaskadierungsfähig sind, keine definierte Größe haben und Probleme bei der Barrierefreiheit bereiten)
 - em (passen sich dynamisch an und können Schriftgrößen kaskadieren, können allerdings den Kontext ungewollt verändern)
 - Prozent (relativ ähnlich wie em, deswegen ist em auch die bevorzugtere Methode)⁵⁰

2.2.1.1 Media Queries

Die wohl wichtigste Neuerung bei CSS3 sind die sogenannten Media Queries, wie vorhin bereits erwähnt wurde. Damit können CSS-Angaben in Relation zu den Eigenschaften des Ausgabegerätes vorgenommen werden. Prinzipiell werden Medienabfragen am häufigsten eingesetzt, um CSS-Statements für kleinere Displaygrößen zu setzen. Man kann sie aber auch in Kombinationen mit anderen Techniken benutzen. So ist es beispielsweise möglich, über Media Queries die feinen Unterschiede zwischen den vielen verschiedenen mobilen Ausgabegeräten zu berücksichtigen, um Inhalte bestmöglich darstellen zu können (Abb. 2.2.5). Aber auch Webapplikationen können mittels dieser Abfragen eine angepasste Ausgabe erzeugen und zwar je nachdem, ob das Gerät im sogenannten „Portrait-View“- oder „Landscape-View“-Modus ist.⁵¹

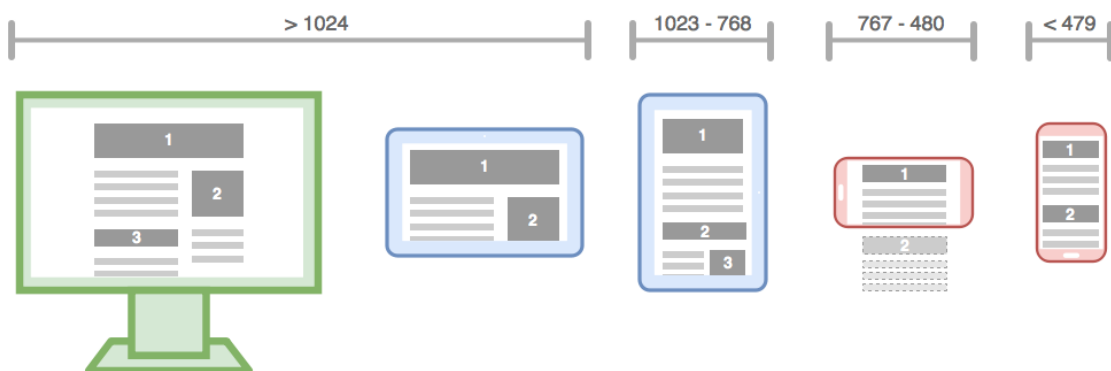


Abbildung 2.2.5: Anordnungsmöglichkeiten durch verschiedene Media Queries nach Pixelbreite

Die Angabe eines Medientyps ist die einfachste Form einer Medienabfrage und gleichzeitig Grundlage für komplexere Abfragen. Ein Medientyp ist ein Schlüsselwort, dem verschiedene gleichartige Ausgabegeräte zugeordnet werden können. Allerdings gibt es keine feste Definition, welche Geräte einem Medientyp zuzuordnen sind (Tab. 2.2.1). Die

⁵⁰vgl. Kadlec (2013), S.25 ff.

⁵¹vgl. Maurice (2012), S.115.

Beschreibung der einzelnen Medientypen wird dabei ausdrücklich als Richtbeschreibung bezeichnet, da ein Browser nämlich nur einen einzigen Medientyp berücksichtigen darf. Ein Dokument kann also nicht von zwei Stylesheets, welchen unterschiedliche Medientypen zugeordnet wurden, gleichzeitig beeinflusst werden.⁵²

Medientyp:	Richtbeschreibung:	Unterstützung:
all	alle Ausgabemedien (Standardwert)	- - -
print	für Drucker, die Inhalte sichtbar auf Papier drucken	Android, Chrome, Firefox, Internet Explorer, Opera, Safari
screen	(Computer-) Bildschirme	Android, Chrome, Firefox, Internet Explorer, Opera, Safari
speech	für Sprachsynthesizer, reserviert für ein zukünftiges CSS-Modul (ersetzt den Medientyp aural)	CSS2.1, Opera

Tabelle 2.2.1: Übersicht der Medientypen in CSS

„Die Definition der Ausgabemedien und die der Media Queries erfolgt mittels der *@media* Anweisung innerhalb der einzelnen Cascading Stylesheets, welche wiederum andere Styles umfasst. Die öffnende Zeile steht dabei vor dem Style und die schließende geschwungene Klammer danach (Abb. 2.2.6). Die Anweisung *@media* kann in einem Stylesheet mehrmals auftreten, darf aber keinesfalls verschachtelt werden. Bevor das zweite *@media* beginnt, muss das vorherige demnach geschlossen werden.“⁵³

```

1  @media print {
2
3      /* Ausgabedefinition per Media Query */
4
5  }
6
7  /* oder via meta-viewport Angabe im head: */
8
9  <link rel="stylesheet" href="thesis.css"
    media="screen and (min-width: 960px)">

```

Abbildung 2.2.6: Beschränkung der Styles auf die Druckausgabe per Media Query bzw. meta-viewport

2.2.2 Scalable Vector Graphics

Grafiken transportieren Informationen oft viel schneller und exakter als ein geschriebener Text. Sie rufen einen enormen Aufmerksamkeitsgrad, begleitet durch eine erhöhte kognitive Wahrnehmung, außer bei Blinden, hervor und verdoppeln somit – in Kombination mit dem Text – den allgemeinen Informationsfluss. Dass dabei die Speicherfähigkeit

⁵² s. SelfHTML Wiki: CSS/Media Queries, (Stand: 27.09.2016).

⁵³ Müller (2013), S.170 f.

im menschlichen Gehirn signifikant angeregt wird und viele Emotionen und Assoziationen dabei geweckt werden, wissen kommerzielle Webseitenbetreiber mittlerweile gut zu nützen.⁵⁴ Damit dies im weiteren Verlauf aber auch erfolgen kann, existieren bestimmte Anforderungen an Bilddateiformate innerhalb des Internets, die wie folgt lauten:

- flexible und erweiterbare Dateiformate
- schnellere Verbreitung der Akzeptanz durch Standardisierung
- schnellstmögliche Übertragbarkeit im Internet
- Indizierungsmöglichkeiten für spätere Recherchezwecke⁵⁵

Obwohl Webseiten eigentlich seit Anbeginn weitgehend pixelbasierten Grafiken ausgeliefert waren, wurde mit HTML5 endlich das neue SVG-Format unterstützt, welches man nativ in ein Dokument einbinden kann.⁵⁶ Im Grunde genommen ist SVG eine weitere Spezifikation des W3C (*Abb. 2.2.7*), welche für zweidimensionale Vektorgrafiken, sowie grafische Anwendungen in XML entwickelt wurde. Dadurch, dass SVG ebenfalls eine XML-basierte Sprache darstellt, profitiert sie extrem vom offenen Standard, der Plattformunabhängigkeit und dem enormen Funktionsumfang.⁵⁷



Abbildung 2.2.7: Offizielles Scalable Vector Graphics Logo des W3C

Im Allgemeinen bestehen Vektorgrafiken aus vielen geometrischen Formen, welche mathematisch durch die Koordinaten der Stützpunkte und Art der Linienverbindung (z.B. Kreis, Kurve oder Gerade) definiert sind. Ordinäre Rastergrafiken bestehen beispielsweise nur aus Pixelinformationen, die einerseits einen höheren Speicherplatz aufweisen und vor allem beim Vergrößern Qualitätsverluste mit sich bringen. SVG hingegen nutzt die volle Druckerauflösung und ist sogar für den Bildschirm stets richtig skaliert.⁵⁸ Einige weitere Unterschiede dazu sind in der folgenden Tabelle (*Tab. 2.2.2*) angeführt.

⁵⁴vgl. Hammer (2008), S.35 ff.

⁵⁵vgl. Mahle (2004), S.2 f.

⁵⁶vgl. Potschien (2013), S.101.

⁵⁷vgl. Pomaska (2005), S.184.

⁵⁸vgl. ebd.

Merkmal:	Vektorgrafik:	Pixelgrafik:
Verwendung	exakte Zeichnung	Scans, digitale Fotos
kleinste Informationseinheit	zwei- oder dreidimensionale Objekte in variabler Form	zweidimensionale Pixel in fester Form
Skalierung	stufenlos (ohne Qualitätsverlust)	stufenlos (mit Qualitätsverlust)
Dateigröße	klein bis mittel und variabel (je nach Objektmenge)	groß und fest
Komprimierung	Flash nicht, SVG schon	JPEG, GIF, PNG
Ausgabegröße	beliebig (wird immer neu gerastert)	vorgegeben (bei Erhöhung der Größe erfolgt Pixelwirkung)
Internet / Formate	Flash, SVG, WebCGM, VML	JPEG, GIF, PNG
benötigte Rechenleistung	hoch 3	sehr gering
Eignung für realistische Darstellung	gering (jede Datei wird als Vektor beschrieben)	hoch (jedes Pixel einzeln beschreibbar)

Tabelle 2.2.2: Unterscheidungsmerkmale von Vektor- & Pixelgrafiken ⁵⁹

Moderne Grafikprogramme sind mitunter aus diesem Grund bereits in der Lage, Dateien im SVG-Format abzuspeichern. Wenn der Webbrowser dann noch HTML5 unterstützt, können die Vektorgrafiken, wie alle anderen Bildformate auch, mittels `<img>`-Element im Quellcode eingebunden werden. Es ist aber auch möglich, die SVG-Dateien per `<object>`- oder `<embed>`-Tag im Quelltext einzubetten, da sich Vektorgrafiken schlussendlich wie normale Dokumente behandeln lassen. Über die Attribute "width" und "height" kann außerdem mittels CSS eine beliebige Größe aufgrund der besseren Skalierbarkeit eingestellt werden. Der wohl wichtigste Vorteil einer jeden SVG-Grafik steckt allerdings in der definierten Auszeichnungssprache, ähnlich wie bei HTML-Dateien. Somit kann dieses Dateiformat in einem einfachen Editor bearbeitet werden, wodurch separate Grafikprogramme wie für andere Vektorgrafikformate überflüssig werden.⁶⁰

2.3 PHP

Obwohl PHP erst seit einigen Jahren existiert, war man in der Webentwicklung bereits vor seiner Erscheinung sehr aktiv. Anstatt der neuartigen Skriptsprache wurden damals noch serverseitige Skripte mittels C oder Perl geschrieben, welche im weiteren Verlauf für Webanwendungen angepasst wurden. Die erste aller PHP-Versionen wurde im Juni 1995 von Rasmus Lerdorf entwickelt, unter dem Vorwand, häufig vorkommende Webprogrammierungsaufgaben einfacher und weniger repetitiv zu gestalten. Interessanterweise stand der Name damals noch nicht für das rekursive Akronym, unter welchem wir es heute kennen (Abb. 2.3.1), sondern für „Personal Home Page“. Durch das vorgegebene Ziel, Codieraufwand weit möglichst zu reduzieren, führte dies zu einer starken HTML-Orientierung, in der PHP-Code schlichtweg in HTML eingebettet wurde und nach wie vor wird. Erst die zweite Version namens „PHP/FI 2.0“, sowie die dritte Version „PHP 3“,

⁵⁹ Mahle (2004), S.7.

⁶⁰ vgl. Potschien (2013), S.101.

konnten einen Wandel verzeichnen. Letztere wurde hauptsächlich von Zeev Suraski und Andi Gutmans entwickelt, welche PHP von Grund auf neu schrieben und damit die bekannten Parsing-Probleme der zweiten Version behoben. Ab dieser Version wurde es auch viel einfacher, die Sprache für Dritte zu erweitern, sowie eigene Module dafür zu schreiben, was den Kernfunktionsumfang erheblich erweiterte.⁶¹



Abbildung 2.3.1: Offizielles PHP Hypertext Preprocessor Logo von PHP.net

Zwar wuchs die Verbreitung von PHP exponentiell, doch anfänglich war es fast unmöglich die Skriptsprache, vor allem für große kommerzielle Webanwendungen, zu nutzen. Entwickler, welche Perl, ColdFusion oder gar andere Sprachen verwendeten, wollten nämlich nicht mehr über die Fähigkeiten von PHP lernen und Java-Entwicklern fehlte es zunehmend an Frameworks, Objektorientierung sowie statischer Typisierung. Heute existiert selbstverständlich keiner dieser beiden Standpunkte mehr, da PHP mittlerweile keine Hilfssprache mehr darstellt.⁶² Der Nachfolger „PHP 4“ brachte dann schlussendlich, neben vielen weiteren Änderungen, den Wechsel zur sogenannten „Zend Engine“ und befriedigte damit sehr viele einstige Gegner. Zend war ein weiteres Vorhaben von Suraski und Gutmans, mit dem die beiden PHP im Unternehmenssektor verbreiten wollten. Dabei übernahm Zend das Herzstück von PHP und erweiterte es um Features wie z.B. Reference Counting, Webserver-Abstraktion und einer komplett überarbeiteten Ausführungsweise des PHP-Codes selbst. Komplettiert wurde PHP mit dem Sprung von „PHP 4“ auf „PHP 5“, obwohl dieser nicht mehr so signifikant war wie jener von der dritten auf die vierte Version. Dennoch ist es nennenswert, dass mit „PHP 5“ eine verbesserte Objektorientierung, die Aufnahme von Exceptions mit try/catch-Fehlerbehandlung, SimpleXML, sowie SQLite eingeführt wurde, wodurch vieles erleichtert und die moderne Webentwicklung nachhaltig umgemodelt wurde.⁶³

Mittlerweile ermöglicht PHP WebentwicklerInnen die Erzeugung dynamischer Webseiten, mit denen auch Webapplikationen, wie z.B. E-Commerce-Systeme, Chats oder Foren

⁶¹vgl. Hudson (2005), S.1 f.

⁶²vgl. Schlossnagle (2006), S.23 f.

⁶³vgl. Hudson (2005), S.2.

erstellt werden können. Ganz im Gegensatz zu statischen Internetseiten kann sich hier der Inhalt aufgrund von unterschiedlichsten BenutzerInnenaktionen bzw. neuer Basisinformationen (z.B. aus der Datenbank) weitgehend verändern. Insbesondere die vereinfachte Auswertung von Formulardaten, mit denen es möglich ist, Daten an einen Webserver zu senden, wird endlich für eine verbesserte Zusammenarbeit zwischen unterschiedlichen DBMS, von PHP unterstützt.⁶⁴ Im Vergleich zu anderen Programmiersprachen bietet PHP sehr viele Vorteile, die unter anderem wie folgt lauten:

- dient zur Entwicklung von Internetanwendungen
- ermöglicht eine einfache Entwicklung von Programmen
- unterstützt verschiedene Plattformen
- lässt sich sehr einfach in den freien Apache-Webserver integrieren
- ist frei verfügbar und sehr flexibel
- ist leicht erlernbar und ideal für den Bildungseinsatz geeignet⁶⁵

Einer der wohl wichtigsten Vorteile von PHP ist die Interpretation des Codes. PHP kompiliert nämlich jedes Mal vorab im Hintergrund den Code in eine Folge von Anweisungen, welche auch „Opcodes“ genannt werden. Diese Anweisungen werden anschließend der Reihe nach ausgeführt, bis das Skript beendet wird. Das unterscheidet PHP, genauso wie auch Java, von herkömmlichen kompilierten Sprachen wie C++, welche den Code in eine ausführbare Datei übersetzen und diese jedes Mal ausführen, wenn dieses Segment wieder benötigt wird. Die ständige Neukompilierung führt dabei zu enormen Performanceeinbußen, welche unter PHP allesamt wegfallen. Andererseits können PHP-Skripte länger zum Kompilieren als zum Ausführen brauchen, wodurch eine größere Zeitverzögerung entsteht. Diese kann aber ganz einfach mittels sogenannter serverseitiger „PHP-Code-Caches“ verhindert werden.⁶⁶

Wie vorhin bereits erwähnt wird PHP-Code in ein bestehendes HTML-Dokument eingebettet. Dazu wird in den meisten PHP-Applikationen die Methode `<?php ... ?>` verwendet, welche im folgenden Minimalbeispiel (Abb. 2.3.2) nochmals vollständig einschließlich des HTML-Codes abgebildet ist.

⁶⁴vgl. Theis (2014), S.18.

⁶⁵vgl. ebd., S.18 f.

⁶⁶vgl. Hudson (2005), S.3.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Thesis Project</title>
5    </head>
6    <body>
7      <p>Erste Zeile in HTML</p><br>
8      <?php echo "Zweite Zeile in PHP <br>"; ?>
9      <p>Dritte Zeile in HTML</p><br>
10     <?php
11       echo "Vierte Zeile in PHP <br>";
12       echo "Fünfte Zeile in PHP <br>";
13     ?>
14   </body>
15 </html>

```

Abbildung 2.3.2: Minimalbeispiel zur Verdeutlichung der Einbettung von PHP-Code in HTML

Das Element „<?php“ kann dabei, wie man sieht, eine einzelne Anweisung oder auch einen ganzen Block an Anweisungen in das HTML-Dokument einfließen lassen. Dies geschieht solange bis der Endtag „?>“ erreicht wird. Demnach können PHP-Blöcke im ganzen Dokument verstreut bzw. auch zwischen HTML-Elementen untergebracht werden, da der Code nur von oben nach unten abgearbeitet wird. Die Anweisung `echo` gibt wiederum den angegebenen Text auf dem Bildschirm aus. Dabei ist es wichtig darauf zu achten, dass der Text entweder in doppelten Anführungsstrichen bzw. in einfachen Hochkommata geschrieben wird. HTML-Elemente wie z.B. das Zeilenumbruchselement `<br>` können einfach in den PHP-Code eingearbeitet werden, um dabei automatisch als valide HTML-Sprache erkannt zu werden.⁶⁷ Sogenannter „Whitespace“, also Leerzeichen innerhalb des Codes, haben selbstverständlich keinerlei Auswirkungen auf die Funktionalität, sondern dienen eher der Übersichtlichkeit.⁶⁸ Zu den wichtigsten Stammfunktionen von PHP zählen:

- Erstellung von PHP-Skripte innerhalb normaler Editoren wie Notepad oder vim
- Kennzeichnung von Variablen durch ein Dollarzeichen vor dem eigentlichen Variablennamen (z.B. `$var`)
- direkte Ersetzung der Daten von Variablen in String-Konstanten
- Zeichencodierung für jede 8-Bit-Sequenz – also jedes Zeichen – muss bekannt sein
- Bedingungen und Verzweigungen wie `if/else`-Anweisungen oder `switch`-Anweisungen, sowie Schleifensteuerung durch `for`- oder `while`-Schleifen
- Möglichkeit der Erstellung und Verwendung von verschiedenen Datentypen, Konstanten und Funktionen⁶⁹

⁶⁷vgl. Theis (2014), S.21 f.

⁶⁸vgl. Williams & Lane (2005), S.19.

⁶⁹vgl. ebd., S.20 ff.

2.3.1 PHP 5.6

Als im Juli 2004 „PHP 5“ erschien konnte zu dieser Zeit noch keiner ahnen, dass die Entwicklungszeit bis zur Version 5.6 ganze weitere zehn Jahre dauern würde. Während Version 5.3 im Juni 2009 ihr Debut feierte, konnten die PHP-EntwicklerInnen erst gegen März 2012 den Nachfolger 5.4 und erst ein weiteres Jahr später Version 5.5 auf den Markt bringen. Von PHP 5.5 bis zur heutigen „finalen“ Version 5.6 (ziemlich relativ, da zum heutigen Stand „PHP 7“ bereits angekündigt und teilweise sogar schon verbreitet wurde)⁷⁰ verging dann erneut ein Jahr und somit wurde PHP 5.6 erst im August 2014 präsentiert.⁷¹

Wie man sieht, musste man sehr lange auf eine neue Version von PHP warten, da sehr viele neue Features und Verbesserungen eingebaut sowie andere Komponenten entfernt bzw. als veraltet markiert wurden. Dabei sind aber auch einige Features geändert worden, welche stellenweise keine Abwärtskompatibilität mehr gewährleisten (z.B. die Methoden `json_decode()`, `mdecrypt_encrypt()`, `mdecrypt_decrypt()` oder GNU Multiple Precision Ressourcen), obwohl das meiste von alten PHP 5 Codes unter 5.6 noch verwendet werden kann. Zusammenfassend beinhaltet PHP 5.6 folgende neue Features:

- konstante Skalare Ausdrücke, welche Zahlen und Strings beinhalten können
- Variable Funktionen mittels neuem `[...]` Operator anstatt `func_get_args()`
- neuer Potenzierungsoperator `[**]`
- Methode `_debuginfo()` (kann Eigenschaften und Werte von Objekten ändern, nachdem sie mit `var_dump()` ausgegeben wurden)
- Default Character Encoding (zum Einstellen einer allgemeinen Zeichenkodierung)
- größerer Dateiupload als bisher möglich (größer als 2 Gigabyte)⁷²
- Parameter entpacken mittels neuem `[...]` Operator⁷³
- Spiegeln von Bildern durch neue Methode `imageflip()`⁷⁴
- Namensräume (ermöglichen Elemente gleichen Namens zu definieren, zu nutzen und eindeutig zuzuordnen mittels `use`, `global` und `class`)⁷⁵

Durchforstet man ein bisschen die offizielle PHP.net Website, findet man sogar ein kleines Logbuch über alle Funktionen, welche für die Version 5.6 geändert wurden. Prinzipiell könnte man meinen, dass diese auch einfach in der vorhin angeführten Aufzählungsliste Platz finden würden, doch die geänderten Kernfunktionen von PHP 5.6 sind so essentiell wichtig, dass sie sich sogar einen eigenen Absatz verdient haben. Unter diesen sogenannten Kernfunktionen wurde nämlich unter anderem auch die Methode `crypt()` verändert,

⁷⁰s. *PHP.net*: Migrating from PHP 5.6.x to PHP 7.0.x, (Stand: 28.09.2016).

⁷¹vgl. *Maurice* (2015), S.5.

⁷²s. *Envatotuts Plus*: PHP 5.6 What's New, (Stand: 28.09.2016).

⁷³vgl. *Theis* (2014), S.100.

⁷⁴vgl. *ebd.*, S.505.

⁷⁵vgl. *ebd.*, S.281 f.

welche im weiteren Sinne die Grundfunktion eines jeden Salted-Hash verschlüsselten Login-Systems darstellt. Ab sofort wird ein sogenannter `E_NOTICE` Error (d.h. Laufzeitfehler) ausgegeben, falls der `salt` Parameter innerhalb der Methode fehlen sollte.⁷⁶ Man sieht also, dass selbst in den Entwicklerstudios von PHP die Salted-Hash-Methode bereits Einzug genommen hat und höchstwahrscheinlich eher weiterentwickelt wird, anstatt jemals entfernt bzw. als veraltet markiert zu werden.

2.3.2 Externer Dateibezug

In PHP ist es möglich, einzelne Funktionen sowie komplette Skripte in externe Dateien auszulagern, damit sie an verschiedenen Stellen erneut verwendet werden können. Durch die verbesserte Objektorientierung seit „PHP 5“ erfreut sich diese Methodik unter Entwicklern mittlerweile großer Beliebtheit. Hierfür stellt PHP die Anweisungen `include()`, `include_once()`, `require()` und `require_once()` zur Verfügung, welche mehrere direkte sowie indirekte Angriffspunkte aufweisen. Selbstverständlich sind diese Schwachstellen keinesfalls PHP zuzuordnen, sondern eher einem sorglosen Umgang mit Dateien und der Programmiersprache selbst. Dabei treten häufig Fallen in Verbindung mit sogenannten „Include-Dateien“ auf, die wie folgt lauten:

- Dateiendungen der inkludierten Dateien werden nicht vom Webserver geparkt
- Dateien liegen in einem öffentlich zugänglichen Verzeichnis
- Dateinamen, welche von außen als Parameter kommen, werden für `include()` und `require()` verwendet⁷⁷

Aufgrund der hohen Flexibilität der Sprache spielt es lediglich eine Rolle, wie der Dateiname der zu inkludierenden Datei zum Zeitpunkt der Interpretation lautet. Ist diese Datei dann auch noch existent und eine für PHP valide Dateiform, wird diese ohne Nachfragen im Code eingebunden. Dieses Verhalten birgt aber eine wesentliche Gefahr in sich: Wird der Wert des Parameters durch externe Daten gespeist, können bei ungenügender Konfiguration des Webserver sensible Daten ausgelesen bzw. schadhafter Code am System ausgeführt werden. Dies wird ermöglicht, da PHP in der Lage ist, bereits entfernte Dateien in ein Skript einzubinden. Das bedeutet, dass nicht zwingend auf eine lokale Datei verwiesen werden muss, sondern Dateien auch per HTTP oder FTP nachgeladen werden können. Der Zugriff kann nur erfolgen, wenn die Option `allow_url_fopen` der PHP-Konfigurationsdatei (`php.ini`) auf 1 steht. Eine allgemeine Deaktivierung dieser Problematik ist nicht möglich.⁷⁸ Wie man sieht, gibt es immer einen Weg, Manipulation über einen Webserver zu betreiben – auch wenn es sich dabei nur um Include-Dateien handelt. Man kann dieser Problematik allerdings einen Riegel vorschieben, indem man

⁷⁶s. *PHP.net*: PHP 5.6 Changed Functions, (Stand: 28.09.2016).

⁷⁷vgl. *Wassermann* (2007), S.67.

⁷⁸vgl. *ebd.*, S.76 f.

eine Aufrufkombination aus `header()` und dem Auslesen bzw. dem Ausgeben einer solchen Datei verwendet. Mittels `header()` wird nämlich der sogenannte „Content-Type“ festgestellt, damit der Client stets weiß, um welche Datei es sich gerade handelt und wie er damit umzugehen hat.⁷⁹

Des Weiteren ist es möglich, in einer einzigen PHP-Datei gleich mehrere Include-Dateien einzubinden und zwar ganz unabhängig davon, an welcher Stelle sich diese im Code befinden. Prinzipiell können die oben genannten Anweisungen als eine Art „Einfügen“-Befehl gesehen und verstanden werden, bei dem der Aufruf der Funktion durch den Inhalt des Skripts ersetzt wird. So können beispielsweise Variablen für Datenbankverbindungsparameter bzw. einfache Header- und Footer-Dateien sinnvoll ausgelagert werden, anstatt diese jedes Mal erneut aufzurufen und damit eine Mehrfacheinbindung zu riskieren (Abb. 2.3.3). Diese Methodik dient vor allem zur Codeorganisation und ist bei größeren Webprojekten kaum auszuschlagen.⁸⁰

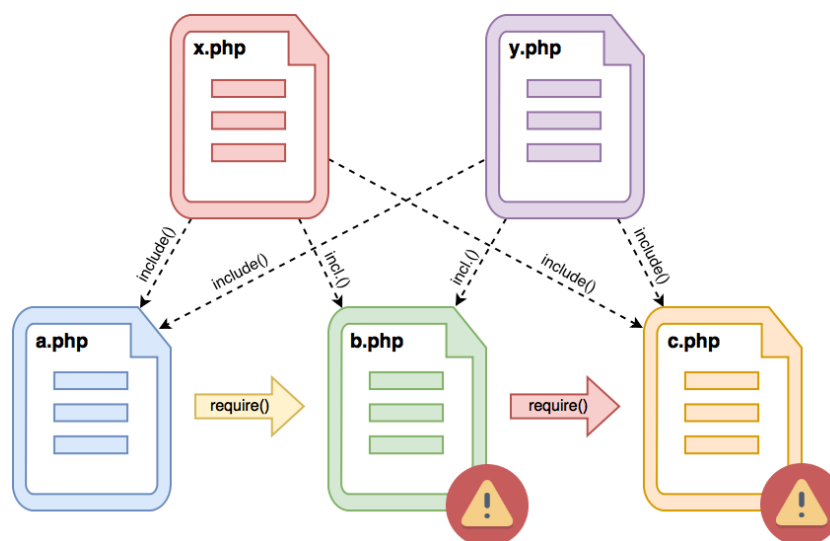


Abbildung 2.3.3: Mehrfacheinbindung von PHP-Skripte durch *include* oder *require*

Um die eben abgebildete Mehrfacheinbindung zu verhindern, zählt es sich aus, die Befehle `include_once()` bzw. `require_once()` anstatt `include()` oder `require()` zu verwenden. Obwohl alle Befehle prinzipiell gleich funktionieren, werden die angeforderten Dateien durch den Zusatz `_once` nur einmal in die Ursprungsdatei geladen (auch bei mehrfacher Anforderung), sofern diese noch nicht in der Liste der bereits eingefügten Dateien stehen.⁸¹ Wichtig dabei ist, dass hier ganz penibelst auf Groß- und Kleinschreibung geachtet wird und man demnach bei der Namensverteilung besonders gut aufpassen muss. Wird versucht eine nicht existierende Datei einzufügen, wird entweder eine Warn-

⁷⁹vgl. Wassermann (2007), S.81.

⁸⁰vgl. Beighley & Morrison (2009), S.255 f.

⁸¹s. PHPnet: Funktion `require_once`, (Stand: 29.09.2016).

meldung ausgegeben, bzw. unter `require()` oder `require_once()` ein Fatal Error, welcher wiederum das Skript abbricht. In älteren PHP-Versionen war das allerdings noch nicht so. Der Befehl `require()` stellte nämlich ein unbedingtes Äquivalent zu `include()` dar, wodurch PHP versuchte, die angegebene Datei zu laden, selbst wenn die Zeile, in der `require()` stand, nie ausgeführt wurde. Mittlerweile überprüft PHP zuerst das angegebene Verzeichnis, in welchem das Skript läuft. Wird die Engine dort nicht fündig, wird der Include-Pfad unter der Direktive `include_path` in der `php.ini` Datei überprüft.⁸²

2.3.3 PHP-Sessions

Ein grundlegendes Erkennungsmerkmal des Internets ist die zustandslose Interaktion zwischen Browsern und Webservern, da das verwendete HTTP-Protokoll ebenfalls zustandslos ist. Dadurch ist kein HTTP-Request vom jeweils anderen abhängig, was für kleinere Internetanwendungen vollkommen ausreicht. Die zustandslose Natur ist allerdings bei Anwendungen, welche eine komplexere Interaktion mit der BenutzerIn erfordern wie z.B. bei einem Onlinewarenkorb, ein regelrechtes Problem, da man nicht mehrere zusammenhangslose sowie zustandslose Webseiten aneinanderhängen kann. Für solche zustandsbehafteten Webdatenbankanwendungen eignet sich aber der Einsatz von sogenannten „Sessions“. Eine Session verwaltet die entstehenden Interaktionen zwischen Webbrowsern und Webservern. Dadurch wird es möglich, bestimmte Informationen, welche durch Benutzeraktionen entstehen, festzustellen und zu verwenden. Wird eine Session gestartet, erhält die BrowsernutzerIn eine eindeutige Session-ID, welche bei zukünftigen Serveranfragen als eine Art Cookie mitverwendet wird. Führt der Browser dann demnach eine Anfrage aus, nutzt der Server die gegebene Session-ID, um etwaige Sessionvariablen zu finden, zu lesen oder zu beschreiben. Bei jedem HTTP-Request und jedem HTTP-Response wird also die Session-ID zwischen dem Webbrowser und dem Webserver transferiert, obwohl die Sessiondaten selbst lediglich am Server abgespeichert werden. Die Speicherung passiert nur serverseitig, da es von hier aus einfacher ist, ungewollte Änderungen der Zustandsinformation noch rechtzeitig zu verhindern.⁸³

Zu den wichtigsten Elementen des PHP Session-Managements zählen unter anderem die Identifikation von Sessions, das Speichern von Sessionvariablen sowie das Bereinigen alter Sessions, welche allesamt vom PHP Session-Handler abgearbeitet werden. All diese Aufgaben werden bereits von der PHP-Bibliothek direkt für das allgemeine Session-Management abgedeckt. Die Standardkonfiguration des Session-Managements nutzt dabei vor allem lokal gespeicherte Dateien, um die Sessionvariablen abzuspeichern. Dateien als Sessionspeicher zu verwenden ist aber nur für jene Webanwendungen kein Problem, bei denen die Anzahl der gleichzeitigen Zugriffe beschränkt ist. Eine bessere

⁸²vgl. Hudson (2005), S.41 f.

⁸³vgl. Williams & Lane (2005), S.363 ff.

und vor allem skalierbare Lösung wäre der Einsatz von MySQL-Datenbanken als Speicherort für Sessioninformationen.⁸⁴ Der Session-Handler kümmert sich dabei um eine reibungslose Ablaufsteuerung aller Aktionen während einer einzigen Operation. Dabei initialisiert er seine Daten nach dem Start, führt eine Speicherplatzbereinigung durch und liest dabei die Sessiondaten der BenutzerIn aus. Die Seitenlogik wird dabei nach dem Befehl `session_start` ausgeführt, wodurch das Skript das Array `$_SESSION` nach Belieben modifizieren und verwenden kann. Wird die Session dann schlussendlich über die Funktion `session_destroy()` geschlossen, werden die Informationen wieder auf die Festplatte geschrieben und die internen Sessiondaten gelöscht (Abb. 2.3.4).⁸⁵

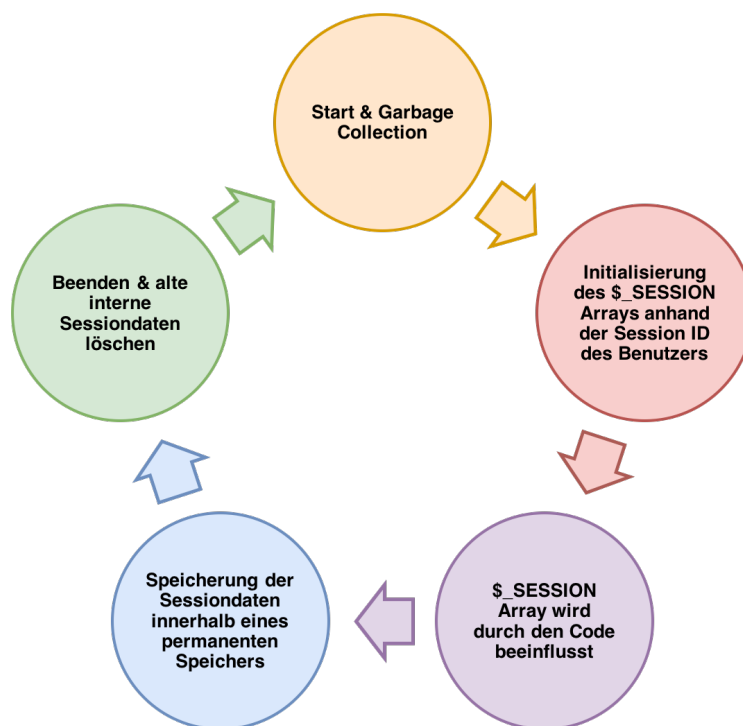


Abbildung 2.3.4: Ablaufplan eines PHP Session-Handlers

Prinzipiell wird im Session-Management zwischen serverseitigen und clientseitigen Sessions unterschieden. Serverseitige Sessions sind Verfahren, bei denen geringe Datenmengen zwischen Server und Client transferiert werden. Dabei wird lediglich die Session-ID übertragen und anschließend dessen Status in einer Art Sessioncache am Server abgelegt. Allerdings lassen sich serverseitige Sessientechniken nur schwer an eine verteilte Architektur anpassen, wodurch clientseitige Sessions oftmals bevorzugt werden, obwohl sie eigentlich einen schlechten Ruf haben (bei jeder Anfrage werden nämlich alle Sessiondaten zwischen Client und Server ausgetauscht). EntwicklerInnen gehen oftmals viel zu kompliziert an eine Problemlösung heran, wodurch z.B. Applikationsserver oder

⁸⁴vgl. Williams & Lane (2005), S.366.

⁸⁵vgl. Schlossnagle (2006), S.406 f.

gar intensive Session-Verwaltungstechniken zum Einsatz kommen. Sie sind aber dennoch sehr beliebt, da sie beispielsweise für eine große Anzahl von Clients sehr leicht skalierbar sind und sich einfach in verteilten Systemen einbetten lassen.⁸⁶

Dadurch, dass Sessions temporäre Daten über ihre Besucher abspeichern, eignen sie sich besonders gut, wenn diese Daten außerhalb des Servers nicht zugänglich sein sollten. Dies wird durch die automatische Umschreibung von URLs ermöglicht, mit der PHP ganz einfach Session-IDs zum Client überträgt. Damit stellen sie eine sehr beliebte Alternative zu Cookies dar, falls die BenutzerIn die Verwendung von Cookies im Webbrowser deaktiviert haben sollte. Demnach stellt eine Session eine Kombination aus einer abgespeicherten serverseitigen Datei, sowie einem clientseitigen Cookie dar, in dem eine Referenz zu den Serverdaten abgespeichert wird. Durch die Funktion `session_start()` im PHP-Code wird sowohl die Serverdatei, als auch das lokale Cookie erzeugt, wobei keinerlei Parameter der Funktion übergeben werden. Diese informiert original den Server, dass ab sofort Sessions zum Einsatz kommen und überprüft, ob die BenutzerIn ein Session-Cookie gesendet hat. Falls dies nicht zutrifft, wird eine neue Sessiondatei am Server erzeugt und eine neue ID an die BenutzerIn gesendet. Anderenfalls sendet PHP einfach die Sessiondaten weiter. Erst die Funktion `session_start()` ermöglicht den Zugriff auf nachfolgende spezifische Sessionvariablen der BenutzerInnen, da diese eindeutig benannt irgendwo am Server weggesperrt liegen und zuerst zugänglich gemacht werden müssen. Das ist auch der Grund, weshalb `session_start()` immer vor dem `<body>`-Tag in einem HTML-Dokument stehen muss.⁸⁷

2.3.3.1 Sessionvariablen

Eine Session besteht, wie vorhin bereits erwähnt, aus zwei verschiedenen Komponenten: Einer Session-ID sowie einer selbst erzeugten Sessionvariablen. Sogenannte Sessionvariablen können Zustandsinformationen enthalten, welche auf Interaktionen der NutzerIn bzw. der verwendeten Anwendung beruhen. Als prädestiniertes Beispiel wäre an dieser Stelle wieder der Online-Warenkorb zu nennen. So können hier beispielsweise Sessionvariablen die verschiedenen Artikel, die Stückzahlen, die Preise sowie andere Artikel speichern, welche sich die BenutzerIn im Zuge seines Einkaufs sonst noch angesehen hat. Dafür werden die Sessionvariablen am Webserver bzw. in der DB gespeichert, wo sie mit Hilfe der Session-ID wieder abgerufen werden können. In der Praxis werden Sessionvariablen normalerweise vom Webserver in einer Datei bzw. vom Datenbankserver in einer Tabelle gespeichert. Allerdings wird es erst durch die vergebene Session-ID möglich, die aktuelle Benutzersession von anderen Sessiondaten zu unterscheiden.

⁸⁶vgl. *Schlossnagle* (2006), S.395 f.

⁸⁷vgl. *Hudson* (2005), S.184.

Die folgende Abbildung (Abb. 2.3.5) zeigt dabei auf, wie Sessionvariablen in der Webserverumgebung gespeichert und indiziert werden, sodass keine Verwechslung oder ähnliches dabei auftreten kann.⁸⁸

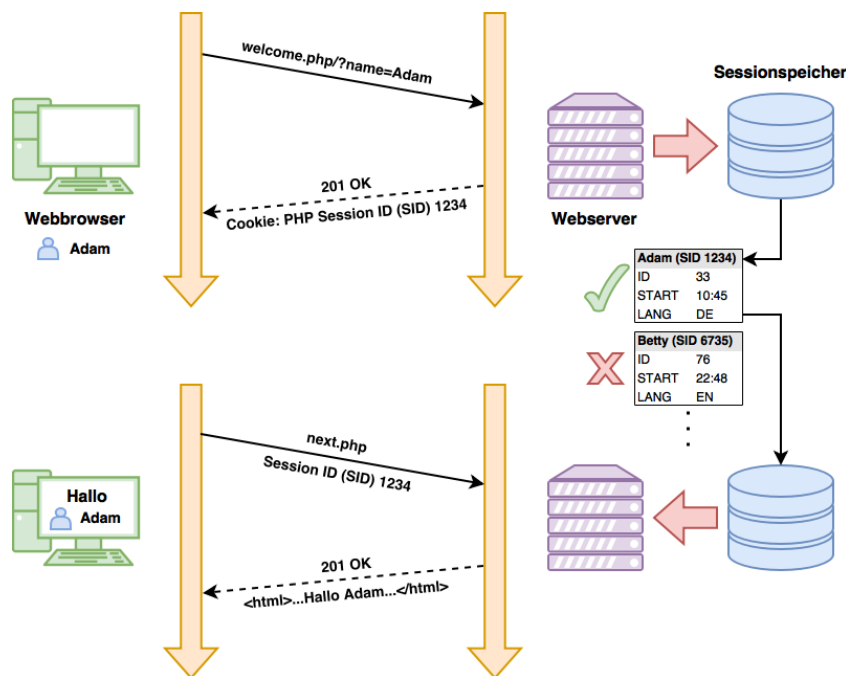


Abbildung 2.3.5: Speicherung & Indizierung von Sessionvariablen anhand der Session-ID

Prinzipiell werden alle Sessiondaten innerhalb des Superglobal-Arrays `$_SESSION` gespeichert. Das bedeutet also, dass jede Sessionvariable zusammen mit ihrem Wert, ein Element dieses globalen Arrays darstellt. NutzerInnen können diesem Array, genauso wie bei anderen, einfachen Variablen hinzufügen, welche selbst nach dem erneuten Laden einer Seite noch vorhanden sind.⁸⁹ Um eine Sessionvariable anschließend wieder zu entfernen, verwendet man die Funktion `unset()` bzw. löscht das gesamte Array mittels des Befehls `array()` oder den gesamten Sessioninhalt via `session_unset()`. Sessionvariablen können dabei jeden Datentyp annehmen, welche auch von PHP selbst unterstützt werden. Allerdings ist es ratsam, sofern man Objekte als Sessionvariablen speichern möchte, die Klassendefinition dieser Objekte in jedem Skript einzuschließen, welches über `session_start()` aufgerufen wird. Dabei ist es vollkommen gleichgültig, ob das aufgerufene Skript diese Objekte auch tatsächlich verwendet oder nicht. Für PHP ist es nur wichtig, die Objekte korrekt aus dem Sessionspeicher zu lesen und später wieder richtig hineinzuschreiben.⁹⁰ Um diese Variablenoperationen besser zu verstehen, sind in der folgenden Abbildung (Abb. 2.3.6) nochmals alle in Codeform dargestellt.

⁸⁸vgl. Williams & Lane (2005), S.364.

⁸⁹vgl. Hudson (2005), S.184.

⁹⁰vgl. Williams & Lane (2005), S.369.

```

1  /* Sessionvariablen mit verschiedenen Datentypen hinzufügen */
2  $_SESSION['thesis-session'] = true;
3  $_SESSION['user'] = $id;
4  $_SESSION['start'] = time();
5  $_SESSION['expire'] = $_SESSION['start'] + (30 * 60);
6
7  /* Sessionvariablen entfernen */
8  $_SESSION['thesis-session'] = false;
9  unset($_SESSION['user']);
10 session_unset();
11
12 /* Sessionvariablen mit Objekten und Klassen hinzufügen */
13 $_SESSION['ein-objekt'] = new eine_klasse();

```

Abbildung 2.3.6: Minimalbeispiel zu den einzelnen Variablenoperationen innerhalb von Sessions

Möchte man Sessionvariablen auf dem Webserver verwalten, müssen diese für jede einzelne Browsersession gespeichert werden. Dabei stellt sich die Frage, wie lange man solche Daten am besten speichern sollte. Dadurch, dass HTTP zustandslos ist, gibt es keine Möglichkeit festzustellen, wann eine BenutzerIn eine Session nicht mehr benötigt. Im Idealfall loggt sich die BenutzerIn über den Aufruf eines Logout-Skripts, welches die Session zerstört, aus. Der Server kann sich aber dennoch nie sicher sein, ob die BenutzerIn immer noch da ist, wodurch er gezwungen ist, alte Sessions zu löschen, welche schon länger nicht mehr benutzt wurden. Diese stellen nämlich ein ziemliches Sicherheitsrisiko dar und verbrauchen noch nebenbei unnötige Serverressourcen. Zusammenfassend lassen sich in diesem Konsens drei unterschiedliche Erkennungsmerkmale feststellen, welche im weitesten Sinne Sessions innerhalb von Webanwendungen ausmachen:

- Informationen und Zustände, welche über mehrere HTTP-Requests erhalten bleiben sollen, müssen immer in Sessionvariablen gespeichert werden
- Jeder HTTP-Request muss eine eindeutige Session-ID mit sich führen
- Sessions müssen Verfallszeiten in Form von Sessionvariablen besitzen⁹¹

2.3.3.1.1 Einsatzmöglichkeiten

Sessionvariablen müssen nicht immer direkt etwas mit benutzerspezifischen Daten sowie aktionsbedingten Informationen zu tun haben. Dadurch, dass sich nämlich alle Datentypen innerhalb solcher Variablen problemlos einsetzen lassen, können Sessionvariablen in sehr vielen unterschiedlichen Einsatzgebieten angewendet werden. Im vorherigen Unterkapitel wurde bereits das Beispiel rund um den Online-Warenkorb durchgespielt, welches verdeutlicht hat, inwiefern Sessiondaten gespeichert sowie schlussendlich wieder

⁹¹vgl. Williams & Lane (2005), S.365 f.

indiziert werden können. Im Regelfall handelt es sich bei solchen Daten um sehr sensible Benutzerinformationen, welche normalerweise mit einer Verschlüsselungsmethode wie z.B. MD5 oder SHA1 gesichert sind. Man kann aber auch ganz einfache Systemdaten in solche Sessionvariablen schleusen, welche im weiteren Sinne Systemaufgaben anstatt Benutzeraktionen betreuen.

In der Praxis sind beispielsweise multilinguale Sprachunterstützungen zum Ändern der Sprache bzw. Captchafunktionen, welche eine starke Barriere gegen Internetbots darstellen, beliebte Tools, welche ebenfalls innerhalb von Sessionvariablen implementiert werden können. Verbindet man nämlich die Sessionvariablen mit sogenannten `$_GET`-Variablen, kann der Status, welcher über eine `$_GET`-Variable verschickt wurde, abgefragt und mit der aktuellen Sessionvariable abgeglichen werden. So lässt sich beispielsweise beim multilingualen Support die Sessionvariable `$_SESSION['lang'] = $lang`; durch die Variable `$lang = $_GET['lang'];` beeinflussen, welche bei jedem Aufruf den entsprechenden Wert innerhalb der Variable `$lang` ändert. Für Captchafunktionen gilt prinzipiell genau das gleiche. Hier wird der zufallsgenerierte Captchawert mittels der PHP Funktion `imagestring()` in ein PNG- oder JPEG-Bild geschrieben, wodurch sich der Text nicht mehr markieren bzw. kopieren lässt und damit gezwungenermaßen von Hand eingegeben werden muss. Dabei wird der Captchawert ebenfalls in einer Sessionvariable gespeichert, damit der Webserver – beispielsweise nach dem Absenden eines Formulars – nach wie vor den Captchawert kennt und diesen mit der Benutzereingabe abgleichen kann.

2.3.4 Cookies

Dadurch, dass HTTP ein zustandsloses Protokoll ist, geraten Daten spätestens dann in Vergessenheit, sobald die Website an den Client gesendet und die Verbindung wieder getrennt wurde. Im Jahr 1994 implementierte Netscape im hauseigenen Browser schlussendlich das Cookie, welches nur von jener Webseite gelesen werden konnte, von der es auch geschrieben wurde. Somit stellte es eine sichere Art dar, um Informationen dauerhaft über Seiten hinweg zu speichern, obwohl es anfänglich eher einem schlechten Ruf unterlag, da der Host darüber feststellen kann, wie oft eine BenutzerIn die jeweilige Webseite aufruft und was dort gemacht wird. Viele Menschen hatten plötzlich große Sorge um ihre Privatsphäre innerhalb des Internets, wodurch unter anderem auch zahlreiche Gerüchte entstanden. Beispielsweise wurde Cookies nachgesagt, dass sie jede Information auf der Festplatte lesen könnten, wodurch viele Zeitschriften und Blogs auf diesen Zug aufsprangen und sogar zur Anwendungsdeaktivierung von Cookies im Webbrowser rieten. Mittlerweile hat sich die vorerst angespannte Situation wieder beruhigt und Cookies werden im Allgemeinen von der Mehrheit akzeptiert.⁹²

⁹²vgl. Hudson (2005), S.181.

Prinzipiell stellen Cookies lediglich reine Textinformationen dar, welche auf Aufforderung eines Webserverns durch den Internetbrowser auf der Festplatte des Clients abgespeichert werden.⁹³ Wenn die gleiche Webseite oder eine Seite der gleichen Domäne erneut aufgerufen wird, so schickt der Browser die im Cookie enthaltenen Textinformationen an den Webserver zurück. Der Webserver selbst ist aber lediglich in der Lage, ihm bereits bekannte Informationen im Cookie abzuspeichern, sodass auch nur solche Informationen vom Webbrowser an ihn zurückgesandt werden. Die oben genannte Einschränkung kann aber auch soweit technisch umgangen werden, dass andere Domänen ebenfalls Gebrauch von einem abgespeicherten Cookie machen können. In diesem Fall spricht man allerdings von Cookies von Drittanbietern (*Abb. 2.3.7*).⁹⁴ Selbstverständlich besteht auch die Möglichkeit, in aktuellen Webbrowsern wie Safari, Chrome, Firefox oder Opera, die Speicherung von Cookies komplett zu deaktivieren. Diese Option kann aber unter Umständen dazu führen, dass der ordnungsgemäße Aufbau einer Seite, sowie ein reibungsloses Surfen an sich, nicht mehr gewährleistet werden kann.⁹⁵ Grundsätzlich kann man also sagen, dass es sich bei Cookies im weiteren Sinne um eine Art Cachespeicher handelt, durch den eine bestimmte Anzahl von Informationen gespeichert wird, wodurch eine BenutzerIn vom Webserver wiedererkannt werden kann.⁹⁶



Abbildung 2.3.7: Abfrage von Cookieinformationen durch Drittanbieter anderer Domänen

Als WebentwicklerIn kann man sowohl auf Cookies als auch auf Sessions problemlos zugreifen. Es ist schlussendlich mit beiden Techniken möglich, Daten einer Seite über mehrere andere hinweg zu speichern, wobei diese auch einige Unterschiede aufweisen. Cookies können nämlich – ganz im Unterschied zu Sessions – auf eine lange Lebensdauer eingestellt werden, wodurch Daten innerhalb eines Cookies über Monate bzw.

⁹³ s. *Cookie Central: The Cookie Concept*, (Stand: 04.10.2016).

⁹⁴ vgl. *Peyton* (2002), S.53 f.

⁹⁵ vgl. *Christl* (2014), S.21.

⁹⁶ vgl. *Voss* (2003), S.229.

Jahre gespeichert werden können. Da Cookiedaten clientseitig gespeichert werden, ist die Speicherung sensibler Daten allerdings nur dann sinnvoll, wenn z.B. in einem Cluster mehrerer Webserver gearbeitet wird. Sessions wären hier nicht nützlich, da nach der ersten fertig abgearbeiteten Anfrage eines Webserver die Informationen im Cluster schlichtweg nicht mehr verfügbar wären. Außerdem weisen viele moderne Webbrowser eine Speicherplatzbegrenzung bei der Speicherung von Cookies auf, um schädliche Internetseiten davor zu hindern, immens viel Speicherplatz in Form von Cookies auf der Festplatte zu verschwenden. Schlussendlich beruht die Entscheidung, ob man in seinem Webprojekt nun Cookies oder doch lieber Sessions verwenden sollte darauf, ob die Daten noch am nächsten Tag für die BenutzerIn zur Verfügung stehen sollen. Falls dem so sein sollte, kommt man um Cookies nicht herum. Falls sensible Daten gespeichert werden sollen, ist es empfehlenswert, diese in einer gesicherten Datenbank abzulegen und lediglich eine Referenz zur ID im Cookie zu speichern. Dies deshalb, da Cookies dadurch, dass sie am Computer gespeichert werden, von BenutzerInnen sehr leicht bearbeitet werden können. So könnte man beispielsweise die Benutzer-ID, welche zur automatischen Anmeldung bei einer Webseite benötigt wird, umschreiben und somit in den Account der jeweils geänderten Benutzer-ID einsteigen. Genau deshalb werden heutzutage auch oft Sicherheitstoken innerhalb solcher Login-Systeme als eine Art zusätzliche Hackerbarriere implementiert, welche sich aus verschiedenen Benutzerinformationen zusammensetzen und mittels der Benutzer-ID und sonstigen Daten einen eindeutig feststellbaren Hashtag ergeben.⁹⁷

Verlagert man nun die gesamte Thematik zurück zu PHP, muss unbedingt erwähnt werden, dass der Aufruf der Methode `setcookie()` immer vor dem HTML-Formular stehen muss, da HTTP alle Headerinformationen vor dem `<body>`-Tag sendet. In den Kopfzeilen werden Informationen wie beispielsweise der Webservertyp (z.B. Apache), die Seitengröße in Bytes sowie andere wichtige Daten übertragen. Cookies gehören ebenfalls zu diesen Headerinformationen, was heißt, dass nach jedem `setcookie()` der Webserver eine Zeile in den Header schreibt. Die `setcookie`-Funktion kann dabei drei Hauptparameter annehmen, nämlich den Cookienamen, den Wert sowie das Datum wann das Cookie wieder verfallen soll (*Abb. 2.3.8*).⁹⁸ Im einfachen Fall wird in einem Cookie genau ein Wert gespeichert (max. 4 Kilobyte). PHP bietet aber auch die Möglichkeit, eine Arrayvariable als Cookie abzuspeichern, wodurch es ermöglicht wird, die Werte in einer Schleife abzuarbeiten. Für den Fall, dass noch komplexere Daten in einem Cookie gespeichert werden sollen, gibt es die Möglichkeit, den Variableninhalt mittels der Methode `serialize()` zu serialisieren, indem eine Zeichenkette aus der komplexen Variable erstellt wird.⁹⁹

⁹⁷vgl. Hudson (2005), S.181 f.

⁹⁸vgl. *ebd.*, S.182 f.

⁹⁹vgl. Kofler & Öggl (2010), S.171.

```

1  /* Erstellung eines Cookies namens "thesis-cookie" */
2  /* Ablaufdatum: aktuelle Zeit + 5000 Tage */
3  setcookie('thesis-cookie', $data, time() + (86400 * 5000), "/");
4
5  /* Entfernung des Cookies "thesis-cookie" */
6  /* Inhalt wird gelöscht / Ablaufdatum: aktuelle Zeit */
7  setcookie('thesis-cookie', '', time() - (86400 * 5000), "/");

```

Abbildung 2.3.8: Minimalbeispiel zur Erstellung und Entfernung eines Cookies

Der größte Nachteil von Cookies liegt sicherlich darin, dass man sich einfach nicht auf sie verlassen kann. Es lässt sich nämlich nicht mit Gewissheit sagen, ob ein Browser auch wirklich ein Cookie nach einer Sitzung gelöscht hat, oder ob sie eine BenutzerIn aufgrund von Sicherheitsbedenken eigenhändig entfernte bzw. ob sie sogar deaktiviert wurden. Darum ist es generell keine gute Idee, gerade bei sicherheitskritischen Bereichen wie z.B. Anmeldesystemen, Cookies zu verwenden, außer man rüstet sein Login-System mit einem Sicherheitstoken aus. Ein weiterer großer Nachteil von Cookies ist, dass man sie nur vor dem Ausliefern eines Seiteninhalts erstellen bzw. bearbeiten kann. Diese Einschränkung ist innerhalb der Spezifikation von HTTP geregelt und ist nicht PHP bedingt. Des Weiteren werden Cookies nach dem Aufruf der Methode `setcookie()` lediglich erstellt. Erst nach dem Laden der nächsten Seite wird das Cookie auch im Webbrowser als aktiv angezeigt, was ein häufig auftauchender „Fehler“ in so manchem Forum ist.¹⁰⁰

2.3.4.1 Rechtliche Aspekte

Jedes europäische Mitgliedsland und jeder Drittstaat regelt den rechtlichen Umgang mit Cookies auf seine eigene Art und Weise. So unterliegen Cookies beispielsweise innerhalb Deutschlands dem TMG (d.h. Telemediengesetz), sowie dem BDSG (d.h. Bundesdatenschutzgesetz).¹⁰¹ In Österreich werden die Rechtsbestimmungen für Cookies und darüber hinaus wiederum nur im DSG geregelt. Allerdings gibt es auch eine flächendeckende EU-Richtlinie, welche im sogenannten TKG (d.h. Telekommunikationsgesetz) geregelt ist und somit alle EU-Mitgliedsstaaten betrifft. Eigentlich könnte man fast vor lauter Gesetzesbestimmungen und Verhaltensrichtlinien meinen, dass die viele Bürokratie hier den eigentlichen Sinn verfehlt hätte, aber gerade heute, in einer Zeit in der eine Unzahl an Cookies auf den Benutzerrechnern abgespeichert werden, sind diese wichtiger denn je.

Prinzipiell wird in jedem Gesetzestext – wie auch im DSG – zwischen Cookies mit personenbezogenem Inhalt (z.B. Namen), indirekt personenbezogenem Inhalt (z.B. Benutzer-IDs oder auch IP-Adressen) und nicht personenbezogenem Inhalt (z.B. multi-

¹⁰⁰vgl. Kofler & Öggel (2010), S.171 f.

¹⁰¹s. *Meine-Cookies: Rechtliche Aspekte*, (Stand: 05.10.2016).

linguale Sprachunterstützungsdaten) unterschieden.¹⁰² Aus datenschutzrechtlicher Sicht kann sich allerdings nur dann ein Problem ergeben, wenn Cookies zumindest indirekt personenbezogene Daten zum Inhalt haben, bei denen ein besonderer Schutz zur Geheimhaltung gegeben werden soll. Demnach fallen Cookies, welche nicht personenbezogene Daten speichern und vor dem Setzen akzeptiert wurden, auch nicht unter das DSG. Obwohl Cookies auf den ersten Blick nicht wirklich viel über BenutzerInnen preisgeben, verbergen sich in den Buchstaben- und Zahlenkombinationen oft wichtige und vor allem sensible Benutzerdaten. Der Webseitenbetreiber ist durch das Abrufen solcher Cookies aber dennoch nur in der Lage, einen reinen Maschinenbezug dazu herzustellen. Nach §4 (Abs. 1) DSG sind personenbezogene Daten nämlich nur jene, in denen Angaben über einen Betroffenen gemacht werden, dessen Identität bestimmt oder bestimmbar ist.¹⁰³ Das bedeutet, dass die AnwenderInnen über den bloßen Maschinenbezug hinauskommen und einen direkten Personenbezug herstellen müssen, bevor das DSG beim Setzen und Auslesen solcher Cookies greifen kann. Indirekt personenbezogene Daten innerhalb von Cookies entstehen dann, wenn der Ersteller der Cookies die Identität des Betroffenen nicht bestimmen kann, da z.B. die Verwendung von Cookies im Webbrowser durch die BenutzerIn deaktiviert wurde. Für einen „Anderen“, welcher im DSG beispielsweise als Systemadministrator auftritt, würden diese Daten dennoch wieder direkt personenbezogene Daten darstellen, da er dadurch trotzdem in der Lage wäre, die Identität des Betroffenen offen zu legen.¹⁰⁴ Damit man im österreichischen Paragraphendschungel allerdings nicht die Orientierung verliert, werden in der nachfolgenden Tabelle (Tab. 2.3.1) die wichtigsten Paragraphen und Absätze zum DSG¹⁰⁵ aufgelistet und näher erläutert.

Paragraphentitel:	regelt folgende Absätze:
§6 Grundsätzliche Verwendung von Daten	(Abs. 1) Was Daten lediglich dürfen (Abs. 2) Wo der Auftraggeber die Verantwortung trägt (Abs. 3) Vertretungsregelung ausländischer Auftraggeber (Abs. 4) Verwendung von Daten nach Treu und Glauben
§7 Zulässigkeit der Verwendung von Daten	(Abs. 1) Wann Daten lediglich verarbeitet werden dürfen (Abs. 2) Wann Daten übermittelt werden dürfen (Abs. 3) Grundsätze des §6 werden vorausgesetzt
§8 Schutzwürdige Geheimhaltungsinteressen bei nicht-sensiblen Daten	(Abs. 1) Wann diese bei nicht-sensiblen Daten nicht verletzt sind (Abs. 2) Keine Verletzung bei publiquen Daten oder indirekt personenbezogenen Daten (Abs. 3) Wann diese besonders nicht verletzt sind (Abs. 4) Verwendung von Daten strafbarer Handlungen oder Unterlassungen
§9 Schutzwürdige Geheimhaltungsinteressen bei sensiblen Daten	(Abs. 1) Wann diese bei sensiblen Daten nicht verletzt sind

¹⁰² s. *Meine-Cookies: Rechtliche Aspekte*, (Stand: 05.10.2016).

¹⁰³ s. *Rechtsinformationssystem: Gesamte Rechtsvorschrift für Datenschutzgesetz 2000*, (Stand: 05.10.2016).

¹⁰⁴ vgl. *Christl* (2014), S.101 ff.

¹⁰⁵ s. *Rechtsinformationssystem: Gesamte Rechtsvorschrift für Datenschutzgesetz 2000*, (Stand: 05.10.2016).

§12 Genehmigungsfreie Übermittlung und Überlassung von Daten in das Ausland	(Abs. 1) Bei Datenverkehr mit Vertragsstaaten des Europäischen Wirtschaftsraumes (Abs. 2) Bei Datenverkehr mit Empfängern in Drittstaaten mit angemessenem Datenschutz (Abs. 3) Wo der Datenverkehr ins Ausland noch genehmigungsfrei ist (Abs. 4) Wo der Datenverkehr ins Ausland sonst noch genehmigungsfrei ist (Abs. 5) Rechtmäßigkeit der Datenanwendung des §7 wird vorausgesetzt
§13 Genehmigungspflichtige Übermittlung und Überlassung von Daten ins Ausland	(Abs. 1) Wann eine Genehmigung der Datenschutzbehörde einzuholen ist (Abs. 2) Wann die Genehmigung der Datenschutzbehörde zu erteilen ist (Abs. 3) Wann die Genehmigung bei meldepflichtigen Anwendungen zu erteilen ist (Abs. 4) Wie inländische Dienstnehmer eine Genehmigung beantragen können (Abs. 5) Übermittlung von Daten an ausländische Vertretungsbehörden (Abs. 6) Wann der Bundeskanzler zur Einholung einer Genehmigung verpflichtet ist
§24 Informationspflicht des Auftraggebers	(Abs. 1) Welche Informationspflichten Auftraggeber einer Datenanwendung haben (Abs. 2) Verwendung von Daten nach Treu und Glauben (Abs. 3) Wann Informationen entfallen dürfen, wenn Betroffene nicht befragt werden (Abs. 4) Keine Informationspflicht bei Anwendungen, die nicht meldepflichtig sind
§25 Pflicht zur Offenlegung der Identität des Auftraggebers	(Abs. 1) Wann der Auftraggeber seine Identität in geeigneter Weise offenzulegen hat (Abs. 2) Wann die Identität der betroffenen Person und des Auftraggebers anzugeben ist

Tabelle 2.3.1: Wichtige Paragraphen zum DSG über den Schutz personenbezogener Daten

Wie man sieht, darf der Eingriff in das Grundrecht des Datenschutzes durch verschiedenste Datenübermittlungen nur im erforderlichen Ausmaß und mit besonderer Sorgfalt geschehen, wobei stets auf die Einhaltung der Grundsätze in §6 geachtet werden muss. So ist es zum Glück auch kein Problem mehr, Daten über Staatsgrenzen hinaus zu transportieren, welche durch eine globalisierte Welt und einem internationalen Handel auch dafür prädestiniert waren. Nichts desto trotz sind die Seitenanbieter dazu angehalten, Informationen an die BenutzerInnen auszuhändigen, ob und inwiefern Cookies auf deren Webseite zum Einsatz kommen und welche Daten diese umfassen. Die Verwendung von Cookies ist jedoch nur solange zulässig, bis die vertraglichen Verpflichtungen beider Seiten erfüllt wurden. Von diesem Gesichtspunkt her ist eine Verwendung der Daten innerhalb solcher Cookies kaum geregelt. In der Praxis passiert es nämlich sehr häufig, dass jene Daten, die eigentlich nach Vertragsende gelöscht werden sollten, nach wie vor auf den Servern der Webseitenanbieter gespeichert bleiben.¹⁰⁶ Als Beispiel lässt sich hier vergleichsweise ein Onlinebestellvorgang bei Amazon nennen. Dort werden Cookies verwendet um beispielsweise den Vertragsabschluss (d.h. Warenkorberstellung, Zahlungsoptionen, Lieferungsart, usw.) zu ermöglichen bzw. zu erleichtern. In diesem Fall wird das Setzen von Cookies bis zum Abschicken der Bestellung als zulässige Handlung gestattet.¹⁰⁷ Aber auch Lernportale, welche beispielsweise zur Übermittlung von Lehrinhalten und der Kommunikation eines geschlossenen Nutzerkreises dienen, kommen heutzutage ohne Cookies einfach nicht mehr aus und unterliegen somit ebenfalls dem DSG. So sind aber auch öffentlich gespeicherte Profilinformationen genauso wie Profilbilder innerhalb solcher Lernplattformen wichtige Streitthemen, welche neben dem Einsatz von Cookies nochmals explizit im TKG der Europäischen Union geregelt wurden.¹⁰⁸

¹⁰⁶vgl. Christl (2014), S.106 ff.¹⁰⁷vgl. Jähnel (2001), S.88.¹⁰⁸vgl. Horn (2015), S.76 f.

Aufgrund einiger Richtlinien der EU sowie der Datenschutzrichtlinie für elektronische Kommunikation, wurde in Österreich das TKG im Jahre 2003 erlassen. Obwohl Cookies in dieser Richtlinie gar nicht namentlich genannt werden, können dennoch einige Parallelen dazu gezogen werden. So werden beispielsweise unter Abschnitt 12 „Kommunikationsgeheimnis, Datenschutz“ im §95 die Datensicherheitsmaßnahmen geregelt, welche Betreiber eines öffentlichen Kommunikationsdienstes zu folgenden drei Punkten verpflichten:

- Sicherstellung, dass nur ermächtigte Personen für rechtlich zulässige Zwecke Zugang zu personenbezogenen Daten erhalten
- Schutz der gespeicherten oder übermittelten personenbezogenen Daten vor unbeabsichtigter oder unrechtmäßiger Zerstörung, unbeabsichtigtem Verlust oder unbeabsichtigter Veränderung und unbefugter oder unrechtmäßiger Speicherung oder Verarbeitung, unbefugtem oder unberechtigtem Zugang oder unbefugter oder unrechtmäßiger Weitergabe
- Umsetzung eines Sicherheitskonzepts für die Verarbeitung personenbezogener Daten¹⁰⁹

„Diese Regelung ist jedoch nicht nur alleine auf Cookies anzuwenden, sondern auch auf alle anderen Webinstrumente, welche Daten auf einem Computer speichern wie z.B. Web-Bugs oder Spyware.“¹¹⁰

So führt einem die unklare Formulierung bezüglich der „Speicherung von Informationen“ zu der Frage, ob diese Regelung nun für sämtliche Cookies oder lediglich für solche mit indirektem Personenbezug gilt.¹¹¹ Würde nämlich die Informationspflicht auch bei nicht personenbezogenen Daten bestehen, wäre das ein Widerspruch zu den Regelungen im DSG, da diese Art der Daten nach den Bestimmungen nach nicht schutzwürdig wären.¹¹² Nichts desto trotz sind nach wie vor manche EU-Mitgliedsstaaten wie z.B. Deutschland bei der Umsetzung der europäischen Cookie-Richtlinie sehr zaghaft, wodurch nationale Interessen nach wie vor über globale gestellt werden, was sich in naher Zukunft auch höchstwahrscheinlich kaum ändern wird. Der wohl größte Kritikpunkt dieser Rechtsbestimmung liegt aber dennoch in der Unverständlichkeit mancher Begriffsregelungen wie z.B. „Opt-In“ und „Opt-Out“. In diesen wird nämlich geregelt, ob eine BenutzerIn seine Zustimmung („Opt-In“) bzw. die Ablehnung eines solchen Services („Opt-Out“) bekannt geben muss. Obwohl sich die meisten Länder, darunter auch Österreich, für die „Opt-In“ Methode entschieden haben, genügen in anderen europäischen Ländern wie Finnland oder Portugal einfache „Opt-Out“ Lösungen.¹¹³

¹⁰⁹s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Telekommunikationsgesetz 2003, (Stand: 05.10.2016).

¹¹⁰Jahnel (2004), S.339.

¹¹¹vgl. Jahnel (2003), S.109.

¹¹²vgl. Christl (2014), S.113.

¹¹³s. *Artworkz*: Die Cookie-Richtlinie der EU, (Stand: 05.10.2016).

2.3.4.1.1 Security

Zu Beginn des Internets spielte unter WebentwicklerInnen oder gar bei BenutzerInnen die Websicherheit kaum eine Rolle, da aufgrund einfacher statischer Websites wie z.B. durch HTML- bzw. Textdokumente, keine potentielle Angriffsfläche gegeben war, um komplexe Anweisungen an den Webserver oder an Subsysteme zu schicken. Zudem kam hinzu, dass anfangs das Internet hauptsächlich von Forschern benutzt wurde, welche sich untereinander alle bestens kannten und somit einen ordentlichen Umgang mit dem neuen Tool pflegten. Allerdings entwickelte sich die Websecurity nach dem Aufkommen größerer Webapplikationen und der damit einhergehenden Kommerzialisierung in den 1990er Jahren zu einem wichtigen Thema innerhalb der IT-Branche. Durch die frische Sensibilisierung in diesem Segment wurde mit der Zeit selbst den meisten Unternehmen bewusst, dass ihre Webanwendungen durch zahlreiche Firewalls oder verschlüsselte Kommunikationskanäle lediglich auf der Transportebene, aber nicht auf der Applikationsebene gesichert waren. Somit kam es auch rund um das Millennium zu zahlreichen Zwischenfällen und Hackerangriffen, wodurch ein allgemeines Umdenken stattfand. Den Menschen wurde allmählich klar, dass Gefahren selbst im Web drohen können und diese oft einen geringen Aufwand für Angreifer darstellen aber einen horrenden Schaden verursachen können.¹¹⁴ Wenn man bedenkt, dass es nach wie vor eine Unzahl an Webseiten gibt, die entweder mit versteckten HTML-Formularfeldern, welche Informationen in Form von XML-Dateien oder ähnlichem clientseitig speichern oder sensiblen Benutzerdaten innerhalb von URLs bzw. Textdokumenten bearbeiten, kann dieser Paradigmenwechsel auch gar nicht schnell genug von statten gehen.¹¹⁵

So existierten anfänglich nicht einmal ausreichend verbindliche Sicherheitsstandards innerhalb des WWW. Anstatt dessen wurden sogenannte RFC (d.h. „Request for Comment“) oder BCP (d.h. „Best Current Practice“) Hinweise ausgeteilt, welche in gewisser Weise eine Vorbildwirkung für spätere Internetstandards (STD) hatten. Erst im Jahr 2001 wurde durch die Gründung des OWASP (d.h. „Open Web Application Security Project“) im Detail über die Sicherheit von Webapplikationen diskutiert, welches bis heute eine der entscheidenden Säulen im Bereich der Websicherheit darstellt. Dieser Diskurs war auch bereits zwingend notwendig geworden, da manch einer zwischenzeitlich schon glaubte, beispielsweise durch sogenannte „Web-Application-Firewalls“, welche ähnlich wie Netzwerkfirewalls funktionieren, Sicherheitslücken innerhalb des Webs schließen zu können, was sich später natürlich als absoluter Irrglaube herausstellte. Die neuesten Tendenzen in Sachen Websecurity gehen heutzutage viel mehr in Richtung sogenannter „Bug-Bounty-Programme“, in denen Sicherheitsexperten bezahlt Jagd auf Sicherheitslücken machen, wodurch sich Unternehmen auch im weiteren Sinne erhoffen,

¹¹⁴vgl. Schäfers (2016), S.28 f.

¹¹⁵vgl. Splaine (2002), S.105.

den boomenden Hackerschwarzmarkt damit eindämmen zu können.¹¹⁶ Bezieht man nun die gesamte Thematik der Sicherheitsstandards auf Cookies, so fällt einem auf, dass hier ebenfalls ein viel zu geringes Angebot existiert. Die meisten Verhaltensregeln, wie beispielsweise das Setzen oder das Löschen von Cookies, sind allesamt in der SOP (d.h. „Same-Origin-Policy“) geregelt, welche einen eigenständigen Standard für Cookies darstellt.¹¹⁷

Um jene Informationen, welche innerhalb von Cookies gespeichert werden, weitgehend zu standardisieren, hat das W3C eine eigene Spezifikation, namens „P3P 1.0“ (d.h. „Platform for Privacy Preferences“) herausgegeben, welche sich bis heute als aktueller Standard im Umgang mit Cookies behaupten kann. Vor allem bei der Speicherung direkt personenbezogener Daten greift der Standard in jedem Belangen und sollte von jeder Webentwicklerin genauestens beachtet werden, da z.B. für die Verwendung von Third-party Cookies eigene P3P-Informationen innerhalb des Webprojekts zur Verfügung gestellt werden müssen.¹¹⁸

2.3.4.2 Probleme mit Cookies

Obwohl Cookies eigentlich der Erleichterung im Umgang mit dem Internet dienen sollten, können diese einem auch ganz einfach zum Verhängnis werden. Die meisten Probleme damit sind jedoch sicherlich auf einen eher leichtsinnigen Benutzerumgang zurückzuführen, obwohl es auch andere Mittel und Wege für potentielle Angriffe gibt. Ursprünglich wurde angedacht, Cookies überhaupt nur für die Dauer der aktuellen Verbindung am Nutzerrechner zu speichern und danach sofort wieder zu löschen. Allerdings verfolgte die Praxis einen komplett eigensinnigen Weg und so kam es dazu, dass Cookies auf den Computern überhaupt nicht mehr von Webseitenbetreibern gelöscht wurden. Ganz im Gegensatz zum Ursprungsgedanken wurden sogar im weiteren Verlauf die Zugriffsrechte der Cookies ausgedehnt, sodass es Drittanbietern ermöglicht wurde, ebenfalls Gebrauch von diesen Cookies zu machen.¹¹⁹ Die wohl größte Gefahr dieses neuen Usus steckt allerdings nicht an den erweiterten Zugriffsrechten für Dritte, sondern viel eher in der fremden Auswertung, wodurch Personen im schlimmsten Fall identifiziert und auf deren Vorlieben bzw. Interessen Rückschlüsse gezogen werden können. Dies ist auch jener Grund, weshalb wir tagtäglich mit personalisierter Werbung innerhalb von E-Mails oder gar Banneranzeigen im Web konfrontiert sind.¹²⁰ Die so gesammelten Datenprofile werden in der Praxis auch häufig verkauft bzw. vermietet, wofür sich bereits ein florierender Markt entwickelt hat.¹²¹

¹¹⁶vgl. *Schäfers* (2016), S.30.

¹¹⁷vgl. *ebd.*, S.43.

¹¹⁸vgl. *Kofler & Öggel* (2010), S.172.

¹¹⁹vgl. *Brandl & Mayer-Schönberger* (1999), S.368.

¹²⁰vgl. *Jahnel* (2001), S.87.

¹²¹vgl. *Christl* (2014), S.100.

Es stellt an und für sich ein relativ kontroverses Problem dar, dass man Cookies als eine Art Verfolgungstool missbrauchen kann. Dabei kann jeder erdenkliche Schritt, den BenutzerInnen innerhalb des Internets setzen, mitgeloggt werden und somit oft sehr pikante Details von BenutzerInnen offenlegen. So nutzen diese Informationen auch gezielt Unternehmen wie beispielsweise DoubleClick, Globaltrack oder ADSmart, welche Werbebanner auf diversesten Seiten hosten und unter Synonymen wie „Advertising-Rings“ bzw. „Tracking-Networks“¹²² im Internet zu finden sind. Dabei kommt die Ring-Bezeichnung nicht von irgendwo, da das Geschäftsmodell einem einzigen Kreislauf ähnelt. Die Kunden solcher Werberinge platzieren einen `<img>` Tag auf ihrer HTML-Seite, welcher auf eine URL des Werberingwebserver verweist. Sieht ein Webbrowser nun ein solches `<img>` Element, kontaktiert er den Webserver des jeweiligen Werbeanbieters, um die Bildressource zu erhalten. Wenn dieser Werbebanner das erste Mal heruntergeladen wurde, wird ein Cookie am Computer abgespeichert, welches eine zufällig generierte ID enthält. Ab diesem Zeitpunkt wird dann jedes Mal beim Aufrufen einer Seite auf der solche Werbebanner zum Einsatz kommen, dieses Cookie an den Webserver der Werbeagentur geschickt. Dadurch wird den Werbeanbietern ermöglicht, personalisierte Benutzerprofile zu erstellen, in denen genau festgelegt ist, wie sich das Surfverhalten bzw. einzelne Interessen einer BenutzerIn definieren (Abb. 2.3.7). Aus diesem Grund verwenden moderne Webapplikationen auch meist kodierte bzw. verschlüsselte Cookies, um Dateneinsicht oder Datenraub etwas Einhalt gebieten zu können. Allerdings werden dabei oft nur herkömmliche Verschlüsselungsmethoden ohne jeglicher Authentifizierungsmöglichkeit angewendet, welche sehr leicht zu dechiffrieren sind und somit wieder keinen besonders effektiven Schutz darstellen.¹²³

Selbstverständlich stellt das bei weitem noch nicht alle Cookie bezogenen Problemstellungen dar. Wenn man bedenkt, dass Cookies auch oft gerne dazu verwendet werden, um sensible Benutzerdaten wie beispielsweise Benutzernamen oder Passwörter zu speichern, muss eine gewisse Sicherheitsbarriere wie z.B. ein Salted-Hash-Verfahren eingebaut werden, um stets die korrekte Authentifizierung einer BenutzerIn gewährleisten zu können. Fehlt diese gänzlich, wäre jeder Angreifer dazu in der Lage, die Identität einer anderen BenutzerIn zu übernehmen und mittels eines nachgebildeten Cookies in dessen Account oder dergleichen einzusteigen.¹²⁴ Diese Art des Angriffs wird im Fachjargon als „Cookie-Replay-Attack“ bezeichnet, da sie im Wesentlichen die Nachahmung eines fremden Benutzercookies durch einen Angreifer beschreibt. Bei einem solchen Angriff wird die zuvor ausspionierte Session-ID (durch „Sniffing“ oder Ausprobieren) verwendet, um sich gegenüber der Webanwendung ausweisen zu können. Sofern man es schafft eine gültige Session-ID zu replizieren, kann man mit dieser Information der Anwendung vorgaukeln,

¹²²vgl. Knuckles & Yuen (2005), S.502.

¹²³vgl. Khu-smith & Mitchell (2002), S.134 f.

¹²⁴vgl. ebd., S.133.

eine andere BenutzerIn zu sein und somit auf den kompletten Funktionalitätsumfang zugreifen. Dabei spielen natürlich lange Verfallszeiten von Cookies den Angreifern in die Hände, da beispielsweise nach dem Absturz eines Computers oder dergleichen, Sessions nach wie vor gültig und aktiv sein können. Eine andere Angriffsmethode wäre noch die sogenannte „Cookie-Steal-Attack“, in welcher der Angreifer durch Versenden einer schadhafte URL inkl. implementiertem Skript ein Cookie des Opfers kopieren und sich ganz bequem zurücksenden lassen kann (Abb. 2.3.9). Natürlich kann man auch ganz ohne Cookies im WWW angegriffen werden, wenn man z.B. das sogenannte „Session-Hijacking“ bedenkt. Hier wird nämlich ganz im Gegensatz zur „Cookie-Replay-Attack“, die Session-ID bereits während der Applikationsnutzung ausspioniert und nicht erst im Nachhinein, wodurch ganz unabhängig von etwaigen schlecht implementierten Cookie-Verfahren die aktuelle Nutzersitzung recht komfortabel übernommen werden kann.¹²⁵

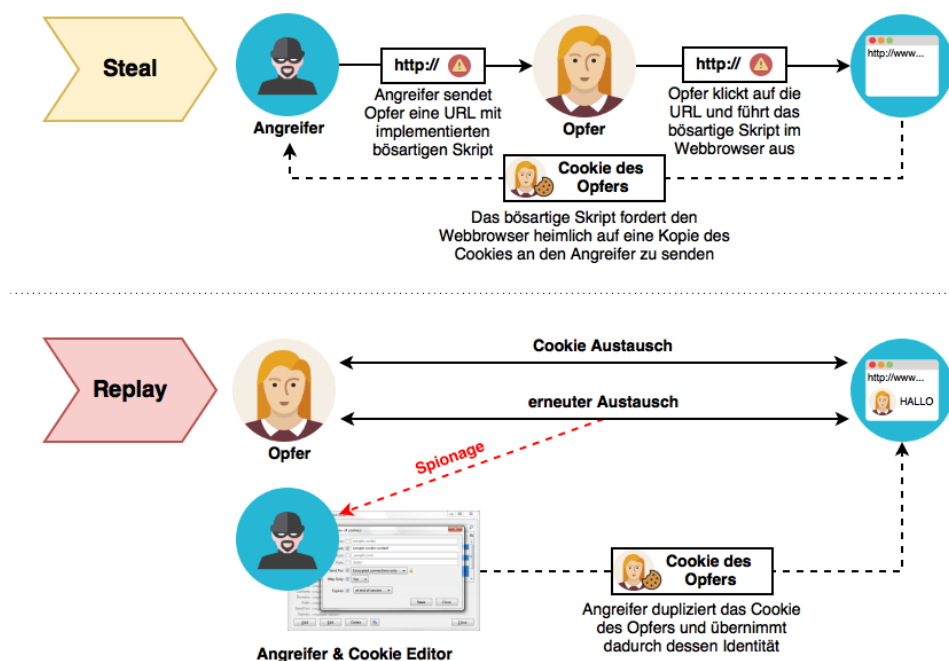


Abbildung 2.3.9: Vorgehensweise von Angreifern bei einer „Cookie-Steal“- bzw. „Cookie-Replay-Attack“

2.3.4.3 Verschlüsselung von Cookies

Grundsätzlich sollte bei jeder Webanwendung eine Verschlüsselung zum Einsatz kommen, um die Möglichkeit eines potentiellen Angriffs zumindest reduzieren zu können. Normalerweise wird an dieser Stelle das HTTPS-Protokoll bzw. ein „HTTP Strict Transport Security“-Protokoll (HSTS) verwendet, um der NutzerIn eine direkte verschlüsselte Verbindung zum Webserver anbieten zu können, wodurch Session-Übernahmen und

¹²⁵vgl. Schäfers (2016), S.92 f.

dergleichen verhindert werden. Sind die HTTPS-Funktionen am Webserver allerdings nicht konfiguriert, muss man zu anderen Verschlüsselungsmethoden zurückgreifen, um zumindest die abgespeicherten Cookies bestmöglich vor Fremden zu schützen. Verschlüsselung bedeutet aber im Wesentlichen nur, dass während einer Verbindung beispielsweise die Session-ID ausgelesen werden kann, was bedeuten soll, dass kein Schutz vor vorhersagbaren Session-IDs oder anderen Angriffsvektoren gegeben ist.¹²⁶ Eine Webseite kann darüber hinaus noch die sogenannte „Secure Cookie“ Option nutzen, wodurch überprüft wird, ob ein Cookie vor seiner Übertragung auch wirklich verschlüsselt ist. Diese Option ist vor allem dann sehr hilfreich, wenn ein Webprojekt aus einer Kombination von mehreren verschlüsselten sowie unverschlüsselten Seiten besteht.¹²⁷

Möchte man jedoch keine Zeit mit SSL verschwenden, kann durch das Zusammenspiel von Cookie-Integrität und Cookie-Authentifizierung innerhalb moderner Webprojekte eine vollkommen gleichwertige Cookie-Verschlüsselung und die damit verbundene Discretion geboten werden. Die notwendige Authentifizierung wäre dabei z.B. durch einen Salt, ein Zertifikat oder einer digitalen Signatur gegeben, wodurch in weiterer Folge „Cookie-Replay-Attacks“ keine Chance mehr hätten. Ein weiterer Vorteil davon wäre beispielsweise der Schutz beim Transfer, sowie bei der Speicherung solcher Cookies, da verschlüsselte Kanäle wie beispielsweise HTTPS lediglich die bloße Übertragung absichern. Ein nennenswerter Nachteil solcher Verschlüsselungsmethoden besteht allerdings im serverseitigen Schlüsselmanagement der einzelnen „Keys“, welche für die Entschlüsselung benötigt werden.¹²⁸ Im Wesentlichen wird zwischen zwei großen Cookie-Verschlüsselungsmethoden unterschieden:

- Serverseitig verwaltete Cookie-Verschlüsselung: Bei diesem Zugang arbeitet der Webserver mit einer Kombination aus unterschiedlichen „Secure Cookies“ wie z.B. einem Username-Cookie (speichert den Benutzernamen für die Authentifizierung), einem Key-Cookie (für den Schlüssel) und einem Sicherheitstoken, welcher die Integrität durch einen zufällig gewählten Hashwert aufrechterhält. Gestohlene „Secure Cookies“ können allerdings nach wie vor an den Webserver gesendet werden, weswegen diese Methode keinen Schutz vor „Cookie-Replay-Attacks“ bietet.
- Clientseitig verwaltete Cookie-Verschlüsselung: Hier hat die BenutzerIn die Kontrolle über sämtliche Sicherheitsmechanismen seiner Cookies und kann Verschlüsselungs- bzw. Kodierverfahren wie z.B. Base64, SHA1 oder den Blowfish Algorithmus einbauen. Demnach ist aber auch das Schlüsselmanagement Aufgabe der BenutzerIn, was im weiteren Sinne ein wesentliches Sicherheitsrisiko darstellt, da die „Keys“ clientseitig gespeichert werden müssen. Um diese weitgehend abzusichern, eignen sich die folgenden zwei Möglichkeiten:

¹²⁶vgl. Schäfers (2016), S.101 f.

¹²⁷vgl. Splaine (2002), S.108.

¹²⁸vgl. Khu-smith & Mitchell (2002), S.137.

- Symmetrische Verschlüsselung: Die NutzerIn aktiviert die Cookie-Verschlüsselung mittels einer Anfrage an den Webserver
- Asymmetrische Verschlüsselung: Die NutzerIn benötigt ein gültiges Zertifikat und ein asymmetrisches Schlüsselpaar¹²⁹

Innerhalb von PHP ist das Verschlüsseln und Überprüfen von sensiblen Cookie-Daten mittels der Methoden `password_hash()` und `password_verify()` möglich. Die Methode `password_hash()` erstellt dabei einen neuen Hashwert aus den eingefügten Stammdaten, indem ein sehr starker Einweg-Hash-Algorithmus benutzt wird. An dieser Stelle ist auf die Methode `crypt()` zu verweisen, welche prinzipiell dasselbe macht, aber weniger Konfigurationsmöglichkeiten, z.B. bei der Wahl des Hash-Algorithmus, bietet. Außerdem werden verschiedene Optionen unterstützt, wie das manuelle Einfügen eines Salt-Parameters oder die Angabe der algorithmischen Kosten, welche stark von der Hardware abhängig sind.¹³⁰ Die Methode `password_verify()` verifiziert im Anschluss den übergebenen Hashwert mit den vorliegenden sensiblen Daten. Dabei übergibt die Methode den gewählten Algorithmus, die Kosten sowie den Salt-Parameter um Serverressourcen einzusparen und ist durch das Retournieren eines herkömmlichen Booleschen Werts vollkommen vor Laufzeitangriffen geschützt.¹³¹

2.3.4.3.1 Base64 Kodierung

Das „Base64-Encoding“ einer Datei ermöglicht die Kodierung der darin liegenden Zeichensätze in 64 Zeichen des ASCII-Codes. Dabei werden aus drei 8-Bit Sequenzen (je 3 Bytes) vier 6-Bit Sequenzen generiert, wodurch insgesamt 256 mögliche Zustände eines Bytes in die 64 möglichen Zustände der Base64 Kodierung umgewandelt werden können (Abb. 2.3.10).¹³² Der Kodierungsstandard wird bis heute, vor allem im E-Mailversand, eingesetzt, da er im „Simple Mail Transfer Protocol“ (SMTP) implementiert ist. So wird praktisch jede E-Mail und auch jeder Anhang wie beispielsweise Archiv- oder Bilddateien, vor dem Versand in Base64 umgewandelt. Bei der Kodierung eines herkömmlichen Klartextes vergrößert sich die Menge um etwa 33%, wobei Füllbytes in Form von Gleichheitszeichen am Ende entstehen können. Base64 kommt in den meisten Webanwendungen bei Cookies oder sonstigen Parametern zum Einsatz, da der Kodierungsstandard fest in PHP eingebettet ist und mittels der Methoden `base64_encode()` und `base64_decode()` gesteuert werden kann. Es ist jedoch möglich, durch entsprechende Tools, welche auch online zu finden sind, sowie durch bloße Umkehrung der Kodierung den kompletten Inhalt zu ermitteln, weswegen die Base64 Kodierung bei sensiblen Daten nicht eingesetzt werden sollte.¹³³

¹²⁹vgl. *Khu-smith & Mitchell* (2002), S.138 ff.

¹³⁰s. *PHP.net*: Funktion `password_hash`, (Stand: 10.10.2016).

¹³¹s. *PHP.net*: Funktion `password_verify`, (Stand: 10.10.2016).

¹³²s. *Brünner*: Der Kodierungsstandard Base64, (Stand: 11.10.2016).

¹³³vgl. *Schäfers* (2016), S.87.

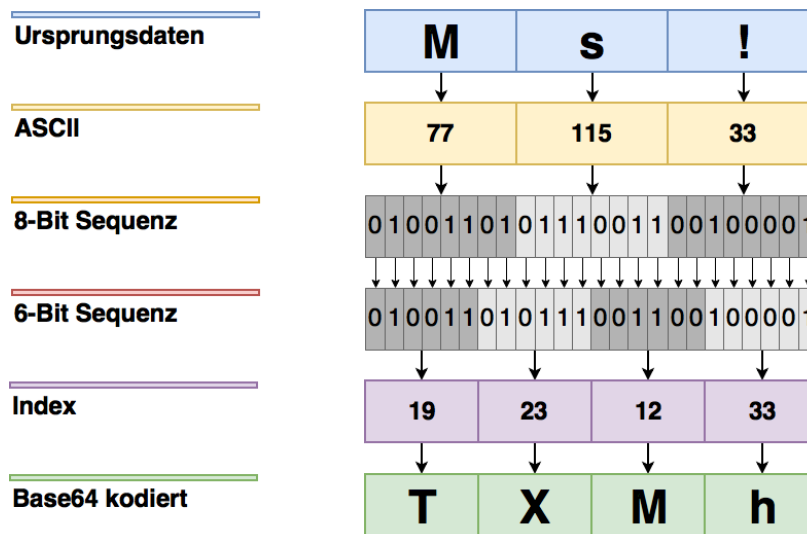


Abbildung 2.3.10: Kodierung einer Zeichenkette durch Base64

Dadurch, dass mit 6-Bit eine maximale Anzahl von 64 unterschiedlichen Werten dargestellt werden kann, wurde das Alphabet so konstruiert, dass es lediglich Zeichen des ASCII Zeichensatzes enthält und somit jederzeit ohne jeglichen Konvertierungsaufwand leicht übertragbar ist. Die folgende Tabelle (Tab. 2.3.2) listet das gesamte Base64 Alphabet und die dazugehörigen Indexwerte jedes Zeichens auf.¹³⁴

Wert:	Kodierung:	Wert:	Kodierung:	Wert:	Kodierung:	Wert:	Kodierung:
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						64	== (Füllbits)

Tabelle 2.3.2: Das komplette Base64 Alphabet¹³⁵¹³⁴vgl. Fuchs (2001), S.109 f.¹³⁵ebd., S.110.

Wie man sieht, muss lediglich eine Sonderregelung für den Schluss einer jeden Zeichenkette getroffen werden, da nicht garantiert werden kann, dass am Schluss auch genau 24-Bit überbleiben. Da ein Byte immer aus 8 Bit besteht und somit kein anderer Fall existieren kann, unterscheidet man generell zwischen drei unterschiedlichen Fällen:

- Es bleiben 24 Bit über: nichts wird unternommen (bereits vier 6-Bit Sequenzen)
- Es bleiben 8 Bit über: vier 0-Bit Sequenzen werden angehängt (daraus entstehen zwei 6-Bit Sequenzen - fehlende 12 Bit werden durch „=“ Füllbits ersetzt)
- Es bleiben 16 Bit über: zwei 0-Bit Sequenzen werden angehängt (daraus entstehen zwei 3-Bit Sequenzen - fehlende 8 Bit werden durch „=“ Füllbits ersetzt)¹³⁶

2.3.4.3.2 SHA1 En-/Decryption

Der „Secure Hash Algorithm“ (SHA) beschreibt einen äußerst sicheren Verschlüsselungsalgorithmus, welcher vom „National Institute of Standards and Technology“ (NIST) in Verbindung mit der „National Security Agency“ (NSA) entwickelt wurde und innerhalb des „Digital Signature Algorithm“ (DSA) für digitale Signaturen verwendet wird. Dieser wurde im Jahr 1993 als ein sogenannter „Federal Information Processing Standard“ namens „FIPS 180“ erstmalig herausgegeben und zwei Jahre später überarbeitet (FIPS 180-1), wodurch sich einige Eckpfeiler rund um SHA1 entwickeln konnten:

- Funktioniert bei Nachrichtenlängen mit bis zu 2^{64} Zeichen
- In PHP über die Methode `sha1()` aufrufbar
- Erzeugt einen 160-Bit Hashwert
- Füllt eine Nachricht um ein Vielfaches von 512 Bits auf (Länge = $448 \bmod 512$)
- Arbeitet achtzig verschiedene Operationen alleine im Hauptalgorithmus ab
- Führt je eine nichtlineare Operation an drei der fünf Variablen A, B, C, D und E pro Operation aus
- Nutzt eine Sequenz logischer Funktionen $f_0, f_1, \dots, f_{79}$, in der jede Funktion f_t , bei der $0 \leq t \leq 79$ gilt, an drei 32-Bit Wörtern arbeitet und ein neues 32-Bit Wort ausgibt¹³⁷

Wie im FIPS 180-1 bzw. im RFC 3174¹³⁸ beschrieben, verarbeitet der SHA-Algorithmus Nachrichten mit beinahe unendlicher Länge, wobei die einzelnen Blöcke eine maximale Länge von 512 Bit zählen. Um die maximale Blocklänge zu erreichen, füllt SHA1 genau gleich wie Base64 die Leerstellen mit Füllbits oder sogenannten „Paddingbits“ auf, damit die Eingabedaten ein Vielfaches von 512 Bit betragen. Daraufhin folgt eine Zusammenkettung der einzelnen Blöcke, gefolgt von einer Aufteilung in 32-Bit Teile und einer anschließenden neuen Verkettung (Abb. 2.3.11). SHA1 operiert mittels eines Eingabeblocks

¹³⁶vgl. Fuchs (2001), S.110.

¹³⁷vgl. Mogollon (2007), S.131.

¹³⁸s. MySQL: Encryption Functions, (Stand: 11.10.2016).

von 512 Bit Länge, einer Prüfsumme von 160 Bit und 80 Schritten pro Datenblock. Dazu kommen noch einige nichtlineare Funktionen sowie mehrere Additionskonstanten hinzu. Als Nachfolger von SHA1 hat die NIST SHA2 standardisiert. Unter diese Bezeichnung fallen Algorithmen wie SHA224, SHA256, SHA384 oder SHA512, welche zu den sichersten Hash-Algorithmen überhaupt gehören und allesamt von PHP unterstützt werden.¹³⁹

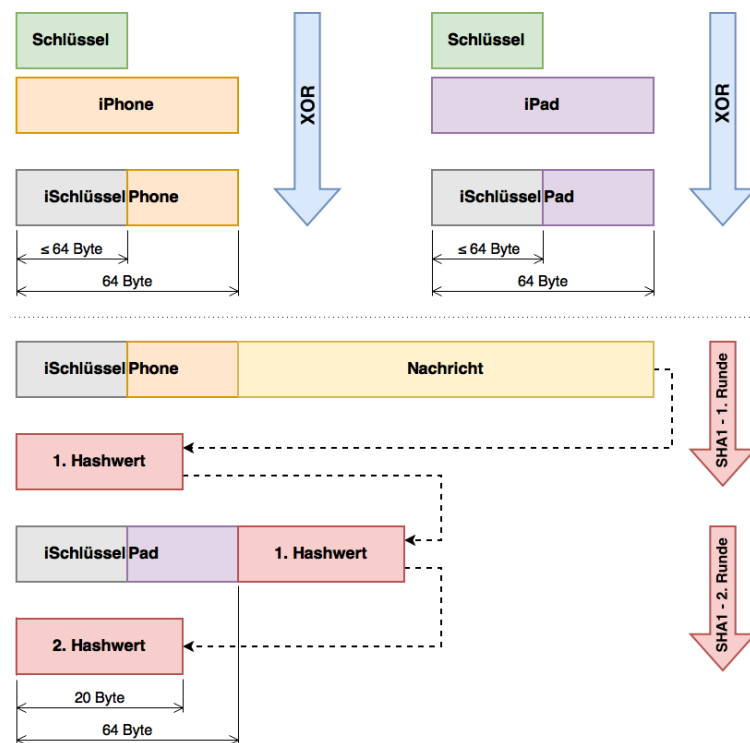


Abbildung 2.3.11: Schematische Darstellung eines SHA1 Verschlüsselungszyklus

Das Auffüllen der Nachrichten passiert im Wesentlichen deswegen, damit am Schluss eine Nachricht mit 512 Bit Länge finalisiert werden kann. Dafür wird als erstes eine 1 angeheftet, welcher im Anschluss so viele Nullen wie nötig folgen, um es auf 64 Bit eines Vielfachen von 512 Bit zu kürzen. Schlussendlich wird ein 64-Bit Integer am Ende der Nullen angehängt, um die finale Länge von $n * 512$ Bits zu erreichen. Dabei repräsentiert der Wert der Integer die Länge der ursprünglichen Nachricht. SHA1 teilt dabei den Input sequentiell in 512-Bit Blöcke auf und arbeitet diese der Reihe nach ab, währenddessen der Hashwert abschließend erstellt wird.¹⁴⁰ Zuvor muss die Nachricht allerdings noch in n M-Bit Blöcke gepart werden bevor die Hashgenerierung starten kann. Bei SHA1 wird die aufgefüllte Nachricht demnach in n 512-Bit Blöcke $M^{(1)}, M^{(2)}, \dots, M^{(n)}$ gepart. Durch die 512 Bit des Eingabeblocks lässt sich auf insgesamt sechzehn 32-Bit Wörter schließen, wobei die ersten 32 Bit des Nachrichtenblocks i als M_0^i gekennzeichnet sind, die nächsten 32 Bit als M_1^i , bis hin zu M_{15}^i . Um den Hashwert zu berechnen, verwendet SHA1 zwei

¹³⁹s. ITWissen: SHA (Secure Hash Algorithm), (Stand: 11.10.2016).

¹⁴⁰vgl. Rhee (2003), S.149.

unterschiedliche Puffer, welche je aus fünf 32-Bit Wörtern und einer Sequenz von achtzig 32-Bit Wörtern besteht. Die Wörter des ersten Puffers werden als H_0, H_1, H_2, H_3 und H_4 bezeichnet, während die Wörter des zweiten Puffers a, b, c, d und e genannt werden und die Elemente der letzten achtziger Sequenz als $W_0, W_1, \dots, W_{79}$ gekennzeichnet sind.¹⁴¹

2.3.4.3.3 „BCrypt“ Blowfish-Algorithmus

Im Jahr 1993 wurde der Blowfish-Verschlüsselungsalgorithmus von Bruce Schneier entwickelt, welcher erst im darauffolgenden Jahr publiziert wurde und ab diesem Zeitpunkt frei verfügbar war. Ursprünglich wurde der Algorithmus hauptsächlich in der Mikrocontrollerelektronik verwendet, obwohl er sich auch bestens für die Verschlüsselung sensibler Daten – z.B. in Datenbanken bzw. für sichere Kommunikationskanäle – mit seltenem Schlüsselwechsel eignet. Blowfish verwendet, im Gegensatz zu SHA, 64-Bit Blöcke und kann Schlüssel mit einer Länge von bis zu 448 Bit einsetzen. Prinzipiell besteht der Algorithmus aus zwei Teilen, wobei lediglich der zweite Teil den Verschlüsselungsprozess umfasst. Der erste Teil löst den Schlüssel in verschiedene Teilschlüssel auf, sodass sich eine Gesamtlänge von 4168 Bit ergibt. Im zweiten Teil wird dann der 64-Bit Datenblock aufgeteilt und innerhalb von 16 Iterationen jeweils schlüsselabhängige Permutationen sowie schlüssel- und datenabhängige Substitutionen unterzogen. Die einzelnen Hälften werden danach iterativ durch mehrere S-Boxen (d.h. nichtlineare Substitutionsoperationen) geschickt, wobei die Substitutionsparameter dieser Boxen durch den Schlüssel generiert werden. Das ist auch einer der Hauptgründe, weshalb Blowfish so unempfindlich gegen kryptanalytische Angriffe ist.¹⁴²

Beim Blowfish-Algorithmus wird ein 64-Bit Eingangsblock, welcher mit Ursprungsdaten befüllt ist, zu allererst in zwei 32-Bit Blöcke aufgeteilt. Die linken 32 Bit durchlaufen einen XOR-Gatter mit dem ersten Element eines P-Arrays, um eine Variable P' zu generieren. Anschließend läuft die neu erstellte Variable P' durch eine Transformationsfunktion S , wodurch sie anschließend, mit den rechten 32-Bit der Ursprungsdaten, erneut durch ein XOR-Gatter muss und dabei eine neue Variable S' erstellt. Dann wird der linke 32-Bit Block der Ursprungsdaten durch S' und der rechte 32-Bit Block durch P' ersetzt. Dieser Prozess wird schlussendlich 15 Iterationen lang mit gültigen Elementen des P-Arrays fortgesetzt. Die daraus resultierenden Variablen S' und P' werden dann noch ein allerletztes Mal mit den beiden letzten Elementen des P-Arrays (17 & 18) durch ein XOR-Gatter geschickt, bevor sie wieder zu einem 64-Bit Block zusammengefügt werden, welcher den verschlüsselten Text repräsentiert (Abb. 2.3.12).¹⁴³

¹⁴¹vgl. Mogollon (2007), S.132.

¹⁴²vgl. Lassmann (2006), S.390 f.

¹⁴³s. Gatliff: Encrypting Data with the Blowfish Algorithm, (Stand: 12.10.2016).

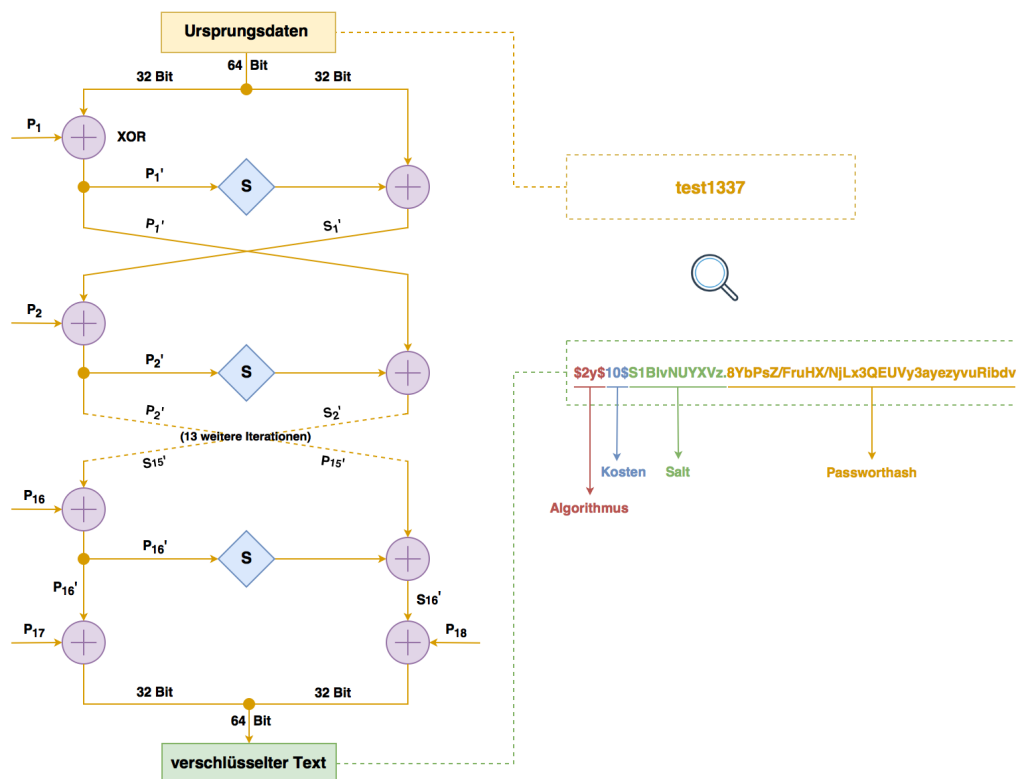


Abbildung 2.3.12: Ablaufdiagramm eines Verschlüsselungszyklus des Blowfish-Algorithmus

Herkömmliche Hashalgorithmen wie MD5 oder SHA1 sind einzig und allein für Performance optimiert, was zwar einerseits gut ist, um Datentransfers oder Zertifikate zu überprüfen, andererseits aber sehr leicht umkehrbar ist und sich somit eher weniger für die Verschlüsselung von Passwörtern eignet. Selbstverständlich kann man an dieser Stelle auch ohne Blowfish-Algorithmus mit ganz einfachen Salts arbeiten, allerdings eignet sich dieser perfekt aufgrund der datenbankseitigen Speicherung. Blowfish ist nämlich sehr langsam und das ist ideal, um Hackerangriffen vorzubeugen. Der Hashwert beinhaltet dabei einen Wert für die Iterationen – die sogenannten Kosten (Aufwand für die Berechnung) – sowie für einen individuell gewählten Salt. In PHP ist „BCrypt“ über die `crypt()` Funktion seit der Version 5.3 integriert und mittels einer der Kennungen `$2a$`, `$2x$` oder `$2y$` am Anfang jedes Hashwerts deutlich erkennbar (Abb. 2.3.12).¹⁴⁴ Es gibt jedoch einige Funktionen, welche ärgerlicherweise im Blowfish Algorithmus nicht vorgesehen sind. So ist es z.B. nicht möglich, einen geheimen Salt innerhalb einer Webanwendung zu hinterlegen oder den Hashwert von zusätzlichen Faktoren wie Benutzernamen oder User-IDs abhängig zu machen.¹⁴⁵

Einige namhafte Kryptographen wie Sarge Vaudenay oder Vincent Rijmen haben Blowfish auf etwaige Schwachstellen und Fehler untersucht und sind dabei auf zwei

¹⁴⁴s. *PHP.net*: Funktion `crypt`, (Stand: 12.10.2016).

¹⁴⁵s. *PHP Gangsta*: Schöner hashen mit BCrypt, (Stand: 12.10.2016).

Erkenntnisse gestoßen. Einerseits können bei schwachen Schlüsseln, bei denen z.B. mehrere Werte gleichzeitig vorkommen, bestimmte Teile ermittelt werden, sofern weniger als 14 Iterationen durchlaufen werden. Des Weiteren wurde sogar von Rijmen mittels einer sogenannten „Second-Order-Differential-Attack“ versucht, den Blowfish-Algorithmus zu knacken. Dieser Angriff war bis zur vierten Iteration erfolgreich, schlug allerdings – und das betrifft die meisten Vorhaben in diesen Belangen – bei jeder weiteren fehl, wodurch der Blowfish-Algorithmus bis heute noch nicht geknackt wurde. Gerade deswegen erfreut sich der Blowfish-Algorithmus unter Entwicklern so großer Beliebtheit und kommt seit 1994 in zahlreichen kommerziellen Web- sowie Softwareanwendungen wie z.B. „OpenBSD“, „TrueCrypt & VeraCrypt“, oder „BestCrypt“ zum Einsatz, in denen sensible Daten verschlüsselt werden sollen.¹⁴⁶

2.3.4.4 Salted-Hash-Cookies

Ab der PHP Version 5.6 existierte durch `crypt()` eine überarbeitete Passwort-API, welche mittels `password_hash()` dazu im Stande ist, aus dem Stegreif sehr sichere Hashwerte für sensible Daten innerhalb von Cookies oder Passwörtern zu generieren. Diese API war von langer Hand geplant, da jeder nur erdenkliche Hashwert herkömmlicher Hashalgorithmen, wie z.B. von MD5, mittlerweile allesamt durch Brute Force (d.h. Ausprobieren aller möglichen Kombinationen) kompromittiert wurden und die entsprechenden Rainbow- bzw. Lookup-Tables (Datenstrukturen, die vorberechnete Hashpaare enthalten, um das Performanceproblem von Brute Force Attacken zu umgehen) sogar mit Google gefunden werden können.¹⁴⁷ Selbstverständlich gibt es solche Tabellen auch für komplexere Algorithmen, da durch den ewig wachsenden Bestand an Cloudfarmen auch immer mehr Rechenleistung zum Knacken solcher Algorithmen vorhanden ist. Um die selbst generierten Hashwerte nicht innerhalb diversester Suchmaschinen wiederzufinden, glauben manche Hobbyentwickler, dass das Problem durch eine krude Vermischung von MD5 mit Base64 gelöst wäre. Dem ist natürlich nicht so, da selbst für solche Kombinationsmöglichkeiten Rainbow-Tables existieren. Das einzige was hilfreich ist, sind sogenannte Salted-Hash-Verfahren, welche einen zufällig generierten Salt (d.h. Salz) in den Hashwert miteinberechnen, so dass dieser nicht mehr durch normale Tabellen auffindbar wird. Genauer gesagt wird der zufällig generierte Salt an das Ende des Strings angeheftet und muss somit bei jeder Validierung genau dort vorkommen. Dadurch, dass jeder Hashwert einen eigenen Salt besitzt, wäre der zu investierende Aufwand, den Hashwert zu knacken, enorm hoch und mit aktuellen CPU-Leistungen gar nicht realisierbar. Aktuell wird in PHP zur Verschlüsselung automatisch „BCrypt“ eingesetzt, da er zurzeit zu den sichersten Hashalgorithmen überhaupt zählt (Abb. 2.3.13).¹⁴⁸

¹⁴⁶s. Doumack: Symmetrische Verschlüsselung mit Blowfish-Algorithmus, (Stand: 12.10.2016).

¹⁴⁷s. Code-Bude: Sicheres Passwort Hashing mit Salts, (Stand: 12.10.2016).

¹⁴⁸vgl. Zander (2014), S.420 f.

```

1  /* Hashwerverstellung mit BCrypt & Salt */
2  password_hash($cookiedata, PASSWORD_BCRYPT);
3
4  /* Überprüfung des Hashwerts mit Datenbankeintrag */
5  password_verify($cookiehash, $fetch->dbdata);

```

Abbildung 2.3.13: Minimalbeispiel zur Erstellung und Überprüfung eines „salted“ Hashwerts

Obwohl die sogenannte „Salt & Pepper“ Methode an und für sich sehr einfach umzusetzen ist, machen es selbst hochrangige WebanwendungsherstellerInnen nach wie vor nicht richtig. So speichert beispielsweise die allseits bekannte Blogsoftware „WordPress“ die Passwörter seiner NutzerInnen im einfachen Hashformat ohne jeglichem Salz. Wenn ein Angreifer nun Zugang zur Passwortdatei eines WordPress-Blogs erhält, kann dieser schwache Passwörter mit vorkompromittierten Listen abgleichen, um sich somit Zugang zum jeweiligen Account zu verschaffen. Viel schlimmer ist es allerdings, dass WordPress einen Hash von diesem unsicheren Hashwert nutzt, um daraus jenes Cookie zu generieren, welches schlussendlich am Nutzerrechner abgespeichert wird. Somit können auch Cookies repliziert werden, um sich Zugang in einen WordPress-Blog Account zu verschaffen. Wie man sieht, ist das eines der Negativbeispiele in Sachen schlecht generierter Hashwerte. Aus diesem Grund wird die korrekte Vorgehensweise eines Salted-Hash-Verfahrens nochmals in der folgenden Abbildung (Abb. 2.3.14) schematisch dargestellt.¹⁴⁹

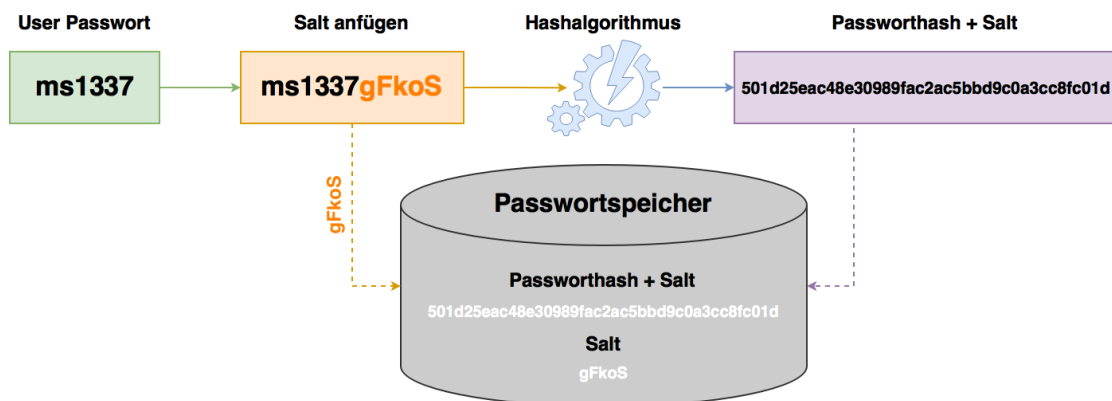


Abbildung 2.3.14: Salted-Hash-Verfahren bei gespeicherten Benutzerpasswörtern

Natürlich gibt es auch weitere häufig auftretende Fehler sowie falsche Mythen rund um Salt basierte Hashverfahren, welche nun in der folgenden Liste näher erläutert werden:

- Hashvorgang erfolgt clientseitig: Die Validierung sensibler Daten muss immer serverseitig geschehen und darf niemals benutzerseitig mittels JavaScript vorgenommen

¹⁴⁹ vgl. Anderson (2008), S.57.

werden. Dabei könnten Angreifer Hashwerte auslesen und somit Vollzugriff auf eine Anwendung erlangen. Außerdem teilt man bei der clientseitigen Verarbeitung den Wert des Salt bzw. des verwendeten Hashalgorithmus mit.

- Doppelte Verwendung von Hashalgorithmen wie vorhin bereits erläutert
- Mehrfachnutzung des Salt: Sehr häufig verwenden Entwickler in ihren Webprojekten immer den gleichen Salt. Vor allem dann, wenn er einfach hart in eine PHP-Datei hinein programmiert bzw. nur einmal generiert und anschließend immer wieder verwendet wird. Angreifer müssen hier lediglich einen einzigen Salt an jeden Hashwert anfügen, um auf alle Daten zeitgleich zugreifen zu können.
- Wahl eines zu kurzen Salt: Wird ein zu kleiner Salt gewählt, werden die zu verschlüsselnden sensiblen Daten nur sehr geringfügig verändert. Auf diese Art und Weise lassen sich ebenfalls sehr leicht Lookup- oder Rainbow-Tables erstellen. Ein Salt sollte immer zumindest halb so groß wie der eigentliche Hashwert sein bzw. im Idealfall sogar gleich groß.¹⁵⁰

2.4 MySQL / MySQLi

Der Vorläufer von MySQL wurde bereits im Jahre 1979 von Michael Widenius unter dem Namen „UNIREG“ für die schwedische Firma TcX entwickelt, welcher über die Jahre im Gegensatz zu Oracle- oder Sybase-Datenbanken nicht viel Aufsehen erregen konnte. Erst als sich datenbankgestützte Webseiten durchsetzen konnten, erkannte TcX das Potential ihrer Software, optimierte die Performance und benannte sich selbst in MySQL AB um. MySQL stieg dabei in Verbindung mit PHP zum am häufigsten eingesetzten DBMS überhaupt auf, wodurch das Unternehmen 2008 von Sun Microsystems und 2 Jahre später durch die Oracle Corporation übernommen wurde (*Abb. 2.4.1*). Die Beliebtheit der Open-Source-Software lag ganz klar auf der Hand, da es MySQL ermöglichte, Daten innerhalb von Tabellen sehr komfortabel und vor allem sehr schnell zu verwalten, wodurch Textdateien zunehmend uninteressanter wurden. So wurde beispielsweise auch die Verwendung sogenannter assoziativer Arrays möglich, welche sich aber mit der Zeit bei großen Datenmengen als sehr unpraktisch erwiesen. Ein weiterer Grund für den Erfolg von MySQL liegt aber sicherlich auch an der engen Verbindung zu PHP, wodurch die meisten Hosting Provider in der Regel PHP-Server mit integrierten MySQL-Datenbanken anbieten und selbst bekannte Open-Source Projekte wie „TYPO3“ oder „WordPress“ mittlerweile auf diese Tools setzen. Selbstverständlich existieren neben MySQL auch noch viele andere DBMS (z.B. DB++, FrontBase, filePro, IBM DB2, MaxDB, mSQL, Microsoft SQL Server, PostgreSQL, Sybase usw.), welche ebenfalls zum Großteil dazu in der Lage sind, mit PHP eng zusammenzuarbeiten. Eine Sonderrolle bezieht an dieser Stelle „SQLite“, welches neue Datenbankfunktionalitäten mit sich bringt und

¹⁵⁰vgl. Schäfers (2016), S.224 f.

Datenbanken in einer einzigen Datei abspeichert. „SQLite“ eignet sich sehr gut, um häufig Inhalte aus einer Datenbank auszulesen, aber nur sporadisch, um neue zu ergänzen. PHP bietet ab der Version 5.3 eine standardmäßige SQLite-Unterstützung, sowie mehrere Abstraktionsebenen für Datenbanksysteme, wodurch unabhängig vom DBMS programmiert werden kann.¹⁵¹



Abbildung 2.4.1: Offizielles MySQL Logo von Oracle

Eine SQL-Datenbank dient im Wesentlichen der Speicherung neuer, sowie der Modifizierung bereits bestehender Daten, welche in tabularer Form mittels Zeilen, Spalten, Spaltenüberschriften (d.h. Datenfelder) und Zellen (d.h. Datensätze) dargestellt werden. Man spricht in diesem Konsens auch oft von relationalen Datenbanken, da die Tabellen auch in Relation zueinander stehen können. Dabei beinhaltet jedes einzelne Datenfeld einen eigenen Datentyp, bei dem es sich beispielsweise um Texte (z.B. `varchar`), Zahlen (z.B. `int`) oder Datumsangaben (z.B. `date`) handeln kann. Um demnach eine relationale Datenbank zu erstellen, sollte man sich, bevor man die einzelnen Tabellen anlegt, verinnerlichen, welche Struktur die Datenbank schlussendlich haben soll. Natürlich kann man die Struktur im Nachhinein noch ändern, aber dies kann mitunter zu Datenverlusten führen. Um innerhalb von PHP eine Tabelle, eine Datenbank bzw. eine geeignete Struktur dafür zu erstellen, wird unter Entwicklern häufig die frei verfügbare Webanwendung „phpMyAdmin“ verwendet, welche Bestandteil einiger lokaler Webserver wie z.B. „XAMPP“ oder „MAMP“ ist und für den Transfer eigener Datenbanken ins Internet genutzt werden kann. Dabei läuft die komplette Bedienung von phpMyAdmin über das Browserfenster ab, wodurch es insbesondere für WebentwicklerInnen ein essentielles Werkzeug darstellt, da sie zur Bearbeitung der Datensätze lediglich ein geeignetes PHP-Interface entwickeln müssen, in welchem die SQL-Befehle ausgeführt werden.¹⁵²

MySQL – wie auch PostgreSQL oder Microsoft SQL-Server – basieren allesamt auf der Abfragesprache SQL, wobei sie sich in manchen Details aufgrund ihres Dialekts etwas voneinander unterscheiden. SQL stellt dabei mehrere Befehle für die Verwaltung einer Datenbank zur Verfügung, mit denen sich mühelos Tabellen erstellen, bearbeiten

¹⁵¹vgl. Maurice (2015), S.510 ff.

¹⁵²vgl. Theis (2014), S.187 ff.

sowie wieder löschen lassen. Außerdem kann man durch INSERT Befehle oder dergleichen Datensätze in Tabellen schreiben und mit UPDATE später aktualisieren. Der wohl mächtigste Befehl ist allerdings die Abfrage, welche besser unter dem SELECT Statement bekannt ist (Abb. 2.4.2). Mit dieser lassen sich nämlich bestimmte Datensätze – gefiltert nach diversen Kriterien mittels WHERE – aus einer oder mehreren Zeilen einer DB auslesen¹⁵³, wodurch jede Datenbank auch eine offenkundige Schwachstelle hat. Des Weiteren wird über Indizierung das Auslesen von Daten erheblich beschleunigt, wodurch jede Spalte innerhalb einer Tabelle einen eigenen Index annehmen kann. So wäre es in einer Datenbank über Telefonbücher beispielsweise sinnvoll, einen Index für die Spalte der Nachnamen festzulegen, wodurch eine Suchabfrage nach einem Nachnamen erheblich schneller von statten gehen würde. Ein besonderer Index in diesem Zusammenhang ist der Primärschlüssel. Dieser Schlüssel ist deshalb von Bedeutung, da er für sich die eindeutige Identifizierbarkeit eines beliebigen Datensatzes einverleiben kann. Daher kann ein Primärschlüssel aus einem oder mehreren Attributen bestehen, welche eine Tabellenzeile eindeutig identifizieren.¹⁵⁴ Häufig wird für diesen Schlüssel eine fortlaufende Nummer als Inhalt gewählt, was über automatisches Inkrementieren geschieht. Hierfür stellt MySQL – wie andere DBMS auch – den Modifikator `auto_increment` zur Verfügung, welcher zwar nicht zum Standardrepertoire von SQL gehört, jedoch die Primärschlüsselverwaltung erheblich vereinfacht.¹⁵⁵ Im vorhin erwähnten Beispiel wäre der Nachname kein geeigneter Kandidat für den Primärschlüssel, da er öfter als einmal vorkommen könnte. Selbst die Kombination aus Vor- und Nachnamen würde keine eindeutige Identifizierbarkeit garantieren, wodurch ein eigenes Feld mit einer einzigartigen ID für jeden Telefonbucheintrag die beste Lösung darstellt.¹⁵⁶

```

1  /* Starten der Datenbankverbindung und Wahl des Zeichensatzes */
2  /* Warnmeldung wird ausgegeben wenn Verbindungsaufbau scheitert */
3  $dbconnect = new mysqli($host, $username, $password, $dbname);
4  mysqli_set_charset($dbconnect, 'utf8');
5  if (mysqli_connect_errno()) {
6      echo 'Keine Verbindung möglich: ' . mysqli_connect_error();
7  }
8
9  /* SQL Abfrage nach allem aus der Benutzertabelle (wo ID = SID) */
10 $usercheck = 'SELECT * FROM thesis_users WHERE id=\''.$id.'\'';
11 $dbconnect->real_escape_string($usercheck);
12 $query = mysqli_query($dbconnect, $usercheck);

```

Abbildung 2.4.2: Minimalbeispiel einer SQL Abfrage inkl. DB Verbindungsaufbau

¹⁵³vgl. Williams & Lane (2005), S.160.

¹⁵⁴vgl. ebd., S.153.

¹⁵⁵vgl. ebd., S.157.

¹⁵⁶vgl. Maurice (2015), S.514.

Wie man in der vorangehenden Grafik erkennen kann, wurde hier bereits mit der neuen MySQLi API gearbeitet, welche neben der alten MySQL API, die seit der PHP Version 5.5 als veraltet markiert wurde, eine neue verbesserte („improved“) Erweiterung für PHP darstellt. MySQL improved wurde entwickelt, um prozedurale sowie objektorientierte Programmierung innerhalb von PHP voranzutreiben und ist außerdem mit einigen attraktiven Features wie „Prepared Statements“ ausgestattet. Die Schnittstelle von MySQLi besteht prinzipiell aus drei verschiedenen Klassen. So ist `mysqli` für den Verbindungsaufbau zuständig, während `mysqli_result` Elemente, die Ergebnisse von SELECT-Abfragen enthalten und `mysqli_stmt` sich um vorbereitete Anweisungen kümmert. Grundsätzlich sind mehrere Schritte notwendig, um mit PHP ein Ergebnis einer SQL-Abfrage zurückzugeben (Abb. 2.4.2):

1. Zu Beginn muss mittels `mysqli` die Verbindung mit dem Datenbankserver hergestellt werden
2. Des Weiteren muss die Datenbank sowie die Anmeldeinformationen für den Datenbankserver innerhalb des `mysqli` Statements festgelegt werden
3. Optional kann auf eine korrekte Kodierung durch die Wahl des Zeichensatzes mit `mysqli_set_charset` geachtet werden
4. Ist keine Verbindung möglich, wird über eine If-Bedingung der Error `mysqli_connect_error()` ausgegeben
5. Nun kann die eigentliche SELECT Abfrage an den Datenbankserver gestellt werden
6. Schlussendlich wird die Abfrage durch `mysqli_query` abgearbeitet, wodurch das Ergebnis mittels `fetch`-Anweisung übernommen werden kann
7. Optional kann via `close()` die Datenbankverbindung wieder geschlossen werden¹⁵⁷

2.4.1 Das SQL-Injection-Problem

Durch die große Vielfalt unterschiedlichster dynamischer Internetportale wie z.B. E-Commerce-Systeme, Dokumentenverwaltungsapplikationen oder ERP-Systeme ist die Anwendung sowie die Sensitivität der Daten, welche solche Systeme speichern, drastisch angestiegen. Dabei stellt die zunehmende Sensibilisierung solcher Daten ein oftmals riesiges Problem für Unternehmen dar, da diese schlussendlich selbst für die Absicherung vor potentiellen Angriffen Sorge tragen müssen. So kommt es häufig dazu, dass hauseigene Entwickler aus diesem Grund eigene Sicherheitsbarrieren entwickeln, welche von diversesten Foren oder dergleichen zusammen kopiert wurden und somit für Angreifer eine perfekte Angriffsfläche bieten, da diese ebenfalls Forenbeiträge öffentlich im Internet einsehen können. Außerdem garantiert kein einziger Forenbeitrag eine korrekte Programmierweise, wodurch sich alleine hier schon massenhaft unbekannte Fehler einschleusen können.¹⁵⁸ Angreifer können den schlecht implementierten Code zu

¹⁵⁷vgl. Maurice (2015), S.570.

¹⁵⁸vgl. Clarke (2012), S.6.

ihrem Vorteil nutzen, um eine sogenannte „Injection-Attack“ auszuüben, bei der es ihnen gelingt, schadhafte Statements an den Web- oder Datenbankserver zu schicken, welche schlussendlich dort ausgeführt werden und somit Informationen offenlegen können. Die drei am häufigsten vorkommenden Typen von Injection-Angriffen sind sogenannte XSS-Attacken (d.h. „Cross-Site Scripting“ – fügt ungeschönte benutzergenerierte Daten in eine Webseite ein), die „Eval-Injection“ (fügt ungeschönte benutzergenerierte Daten in ein Eval-Statement ein) sowie die „SQL-Injection“.¹⁵⁹

Heutzutage interagieren die meisten Webapplikationen mit einer dazugehörigen Datenbank, da alle beliebten Programmiersprachen wie ASP, C#, .NET, Java und auch PHP eine sehr einfache Möglichkeit der Datenbankintegration bieten. So liefert auch jede Programmiersprache ihren eigenen Operationssatz, um SQL-Anweisungen auszuführen, wodurch WebentwicklerInnen oft zu kombinierten Lösungsansätzen greifen, die wiederum ihre eigenen Schwachstellen beherbergen. So gehört auch die SQL-Injection zu einer dieser Schwachstellen, welche meistens dann auftritt, wenn WebentwicklerInnen nicht genau überprüfen, ob die Daten, welche über Formulare, Cookies oder Eingabeparameter in das System gelangen, validiert wurden, bevor sie von einer SQL Abfrage am Datenbankserver abgearbeitet werden. Kann ein Angreifer nun den Input, welcher an die SQL Abfrage geschickt wird, soweit manipulieren, dass dieser vom Datenbankserver als Code und nicht als Daten anerkannt wird, wird dieser ohne Probleme vom Datenbankserver ausgeführt und eröffnet somit viele neue Möglichkeiten der Manipulation (Abb. 2.4.3).¹⁶⁰

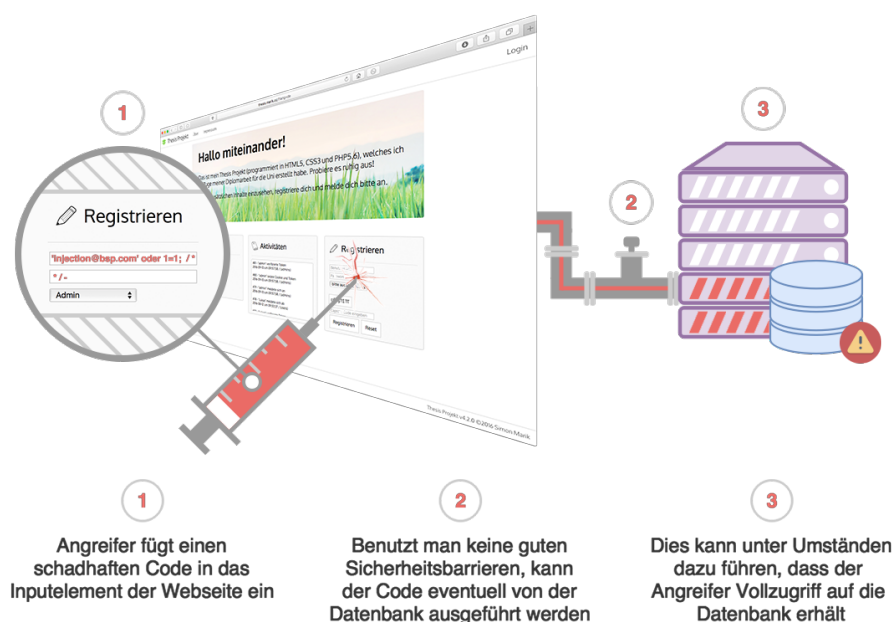


Abbildung 2.4.3: Schematische Vorgangsweise von Angreifern bei einer „SQL Injection“

¹⁵⁹ vgl. Brikmann (2016), S.157.

¹⁶⁰ vgl. Clarke (2012), S.13.

So stellt die SQL-Injection die klassische Angriffsform für ein Skript dar, welches eine SQL-Datenbank verwendet. Durch die simple Einschleusung eigener SQL-Statements innerhalb von Skripte können somit die ursprünglichen Anweisungen verändert, deaktiviert und sogar dazu gebracht werden, komplett eigene Anweisungen der Angreifer auszuführen. Dies kann allerdings nur gelingen, wenn z.B. die alte MySQL-API für den Datenbankverbindungs Aufbau verwendet wird und nicht die neue „improved“ Erweiterung. Meistens übernehmen die Anweisungen der Angreifer direkt die `$_GET`-Parameter einer PHP-Seite, wodurch durch die Angabe eines eigenen Wertes die SQL-Abfrage manipuliert werden kann. Jeder Wert, der in eine SQL-Abfrage integriert werden kann, muss entweder eine Zeichenkette oder eine Zahl sein, wodurch die Auswahl an möglichen Datentypen relativ eingeschränkt ist. Besonders bei Zahlen funktioniert diese Methode recht einfach, da man an Zahlen Code anhängen kann, sofern diese nicht vom System auf ihre Gültigkeit überprüft werden. Bei Zeichenketten stellen die doppelten Anführungsstriche bzw. die einfachen Hochkommata ein regelrechtes Problem dar, da zuerst der richtige Terminierer gefunden werden muss, um eigenen Code an diese anzuheften. Nachdem ein Angreifer jedoch einmal festgestellt hat, ob es sich bei dem zu verändernden Parameter um eine Zahl oder Zeichenkette handelt und gegebenenfalls die Zeichenkette korrekt terminieren kann, kann dieser ohne Probleme seine eigentliche Injection vornehmen.¹⁶¹

Indem man geschickt SQL-Anweisungen in ein vorhandenes System einschleust, kann man eine Anfrage regelrecht Zweckentfremden. Dadurch wird es entweder möglich an Daten zu gelangen, welche vorher gar nicht für eine Ausgabe vorgesehen waren, oder die Bedingungen einer SQL-Abfrage komplett außer Kraft zu setzen. Letzteres lässt sich beispielsweise über die Manipulation von URLs bewerkstelligen, indem man einfach gewisse Zahlenwerte der `$_GET`-Parameter analysiert und somit Rückschlüsse auf die Programmierweise der `SELECT`-Abfrage ziehen kann. Ein gebräuchliches Beispiel hierfür wäre die Session-ID, welche z.B. in Form einer 1 für Administratoren und einer 2 für BenutzerInnen vorkommen kann. Ändert man die entsprechende Zahl in der URL, kann somit – sofern das System nicht durch ein Tokensystem geschützt ist – auf einen anderen Gruppenbereich zugegriffen werden. Selbstverständlich muss ein Zugriff bei einer SQL-Injection nicht immer zwangsläufig über eine `SELECT`-Anweisung erfolgen, sondern kann beispielsweise auch über `DELETE`- und `UPDATE`-Statements geschehen. Auf diese Art und Weise lassen sich möglicherweise Daten – wie z.B. Bestellinformationen eines Onlinewarenkorbs – löschen bzw. aktualisieren, wodurch die Folgen für einen Käufer relativ kostenintensiv werden könnten.¹⁶²

¹⁶¹ vgl. Wassermann (2007), S.351 ff.

¹⁶² vgl. *ebd.*, S.355 f.

2.4.1.1 Verschlüsselte Passwörter in Datenbanken

Aufgrund solcher Probleme ist es besonders wichtig, Passwörter innerhalb von Datenbanken vor fremder Einsicht gut zu schützen. Da dies offensichtlich nur schwer zu erreichen ist, bietet es sich an, die Passwörter, die in einer DB gespeichert sind, verschlüsselt abzulegen, um somit den Angreifern das Handwerk zu legen. Angenommen die Passwörter wurden vorab z.B. mittels des Blowfish-Algorithmus verschlüsselt, ist der Angreifer mit den bloßen Hashwerten nicht in der Lage, Informationen daraus zu rekonstruieren. Passwörter waren aber schon jeher anfällig für Manipulation oder Spionage. So konnte man beispielsweise in den 1980er Jahren die Passworteingabe eines Betriebssystems der US Regierung durch wiederholtes Drücken der Entertaste schlichtweg umgehen, wodurch – nach Entdeckung des Fehlers – an alle Behörden umgehend ein Patch nachgeliefert wurde. Irrtümlicherweise wurden diese Patches allerdings nicht an die US Auslandsbehörden wie z.B. Deutschland verschickt, welche zu diesem Zeitpunkt ohnehin unlizenzierte Versionen der Software verwendeten. Somit gelang es Hackern in dieses System innerhalb Deutschlands einzudringen und alle Informationen zu stehlen. Bis heute wird darüber gemunkelt, dass die Daten damals an den russischen KGB verkauft wurden. Aber selbst im UNIX-Betriebssystem in seiner Ursprungsform konnte jede BenutzerIn ungehindert die Passwörter abfragen, da diese lediglich in einer Textdatei lokal gespeichert wurden, welche noch dazu von jedermann gelesen werden konnte.¹⁶³

Wie man sieht, ist es damals wie heute ungemein wichtig, gespeicherte Passwörter entsprechend in den jeweiligen Systemen abzusichern. Dies gilt vor allem für Datenbankserver und Webanwendungen, da die Daten hier meist irgendwo in der Welt aufliegen, bei dem man nicht genau sicher sein kann, wer gerade darauf Zugriff hat und inwiefern man dort vor potentiellen Cyberattacken geschützt ist. So sollte vor allem ein passender Verschlüsselungsalgorithmus für eine sichere Passwortspeicherung ausgewählt werden, wobei MD5, SHA1 oder gar Base64 sicherlich nicht dazugehören. Auch die vorhin bereits erwähnten Verschachtelungen diverser MD5- oder SHA1-Funktionen würden hier definitiv nicht zum gewünschten Ergebnis führen. Der entsprechende Hash würde sich einfach in einer Lookup-Table wiederfinden lassen, wodurch nur mächtige Verschlüsselungsalgorithmen wie z.B. der Blowfish-Algorithmus für Passwörter in Frage kommen.¹⁶⁴ Weiters sollte darauf geachtet werden, dass Serverfunktionen – gerade jene eines Datenbankservers – lediglich von nicht privilegierten Benutzerrechten ausgeführt werden (in einer `chroot` anstatt `root` Umgebung), um das Risiko vor Datendiebstahl und dergleichen bei erfolgreichen Angriffen zu minimieren. Außerdem stellt dies einen weiteren essentiellen Vorteil von MySQL-Datenbanken dar, da Oracle oder Microsoft Varianten mit sogenannten SYSTEM-Rechten betrieben werden müssen.¹⁶⁵

¹⁶³vgl. Anderson (2008), S.56 f.

¹⁶⁴vgl. Zander (2014), S.420.

¹⁶⁵vgl. Clarke (2012), S.21.

2.5 Webserver

Möchte man ein PHP-Skript von der Festplatte aus starten, wird man im Browserfenster lediglich den gleichen Programmiercode abgebildet bekommen wie zuvor auch in der Entwicklungsumgebung. Festplatten und Browser sind ebenfalls nicht dazu konzipiert, PHP-Code zu verarbeiten, wodurch eine eigene Umgebung benötigt wird, um den Code ausführbar zu machen. Das Environment stellt dabei einen Webserver mit integrierter PHP-Unterstützung dar, von wo aus alle Skripte und Seiten des jeweiligen PHP-Projekts ausgeführt werden. Deswegen müssen sich auch alle entsprechenden Quelldateien auf dem Webserver im Platzhalter `www` befinden, welcher direkt unter `root` liegt. Das Dokumentenverzeichnis wird bei jeder Serverdistribution ein bisschen anders aufgebaut, wobei in diesem Zusammenhang der beliebte Apache-Webserver besonders hervorgehoben werden soll, da auch lokale Serverprogramme wie „XAMPP“ oder „MAMP“ davon Gebrauch machen.¹⁶⁶ Außerdem ist es dabei möglich, einige PHP-Optionen nicht nur innerhalb der `php.ini` zu konfigurieren, sondern direkt durch die Webserver-Konfiguration – z.B. innerhalb einer sogenannten „`htaccess`“-Datei – zu verändern. Welche Methode dabei welche Option ändern kann (Tab. 2.5.1), hängt von der sogenannten „Changeable“ Spezifikation der PHP-Dokumentation ab. Die Konfiguration durch den Webserver ist allerdings erneut sehr vage formuliert, da `htaccess`-Dateien momentan ebenfalls nur vom Apache-Webserver unterstützt werden.¹⁶⁷

Option:	kann geändert werden in / mit:
PHP_INI_ALL	php.ini, httpd.conf, htaccess, ini.set()
PHP_INI_USER	ini.set()
PHP_INI_PERDIR	php.ini, httpd.conf, htaccess
PHP_INI_SYSTEM	php.ini, httpd.conf
php.ini bezogen	php.ini

Tabelle 2.5.1: „Changeable“ Spezifikation der PHP-Dokumentation¹⁶⁸

Solange demnach eine Datei eine Dateieindung enthält, welche vom PHP-Interpreter geparst werden kann (Dateisuffix wurde an das PHP-Modul weitergeleitet), kann diese vom Webserver verarbeitet werden und mittels `echo()` oder durch andere Ausgabefunktionen übergebene Daten an die BenutzerIn ausliefern. Die Variablenzuweisungen, welche beispielsweise für Include-Dateien üblich sind, bekommen die NutzerInnen dabei gar nicht mit. Passiert es, dass eine nicht zu erfassende Dateieindung wie z.B. „`inc`“ vom Webserver ungeparst an den Client ausgeliefert wird, kann dieser eventuell die Variablenzuweisungen einsehen und somit an Passwörter bzw. andere sensible Daten gelangen.¹⁶⁹ Ist die

¹⁶⁶vgl. Beighley & Morrison (2009), S.18 f.

¹⁶⁷vgl. Wassermann (2007), S.403 f.

¹⁶⁸ebd., S.403.

¹⁶⁹vgl. ebd., S.404.

Dateiendung erfolgreich validiert worden, muss der Webserver mit seiner IP-Adresse noch für eine bestimmte Domäne registriert sein, damit die Seite auch schlussendlich im Webbrowser aufgerufen werden kann. Der Webbrowser kennt dabei den Domänennamen nicht direkt, sondern findet diesen beim jeweiligen DNS-Server heraus, je nachdem welche TLD (z.B. .at, .de, .com usw.) die Webseitendomäne besitzt. Der DNS-Server stellt dabei ein System an Servern dar, deren Aufgabe daraus besteht, Domänennamen in IP-Adressen zu übersetzen, wodurch der jeweilige Webserver erst gefunden werden kann. Dabei ist DNS hierarchisch – also in mehreren Ebenen aufgebaut, wodurch je nach Ebene unterschiedliche Informationen gespeichert werden (Abb. 2.5.1).¹⁷⁰

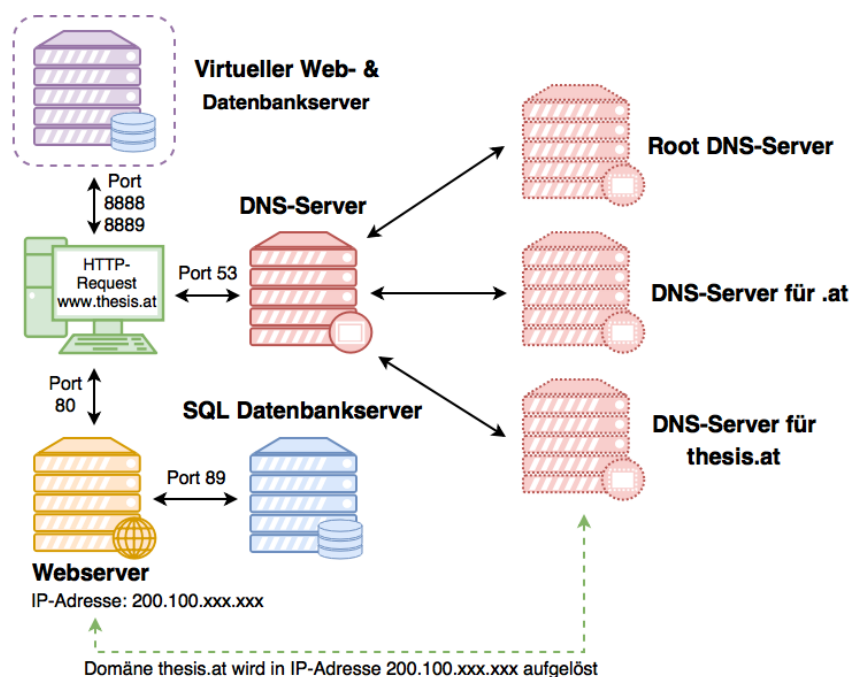


Abbildung 2.5.1: Servernetzwerkstruktur inkl. DNS zur Darstellung einer Internetseite

Über den Port lässt sich der Aufgabenbereich eines Servers ermitteln. So verwendet ein Webserver der mittels HTTP-Protokoll arbeitet meistens den Port 80, wobei auch gern der Port 8888 für Apache-Webserver im Entwicklermodus und 89 sowie 8889 für MySQL-Datenbanken verwendet wird. Der Port kann dabei einen Zahlenraum von 0 bis 65535 abdecken, durch die ein Server weiß, welches Service benutzt werden muss. Das alles wird durch die Zustandslosigkeit des HTTP-Protokolls ermöglicht, welches nach jedem einzelnen HTTP-Response die Verbindung zum Webserver wieder schließt (ohne JavaScript) und den Statuscode 200 an den Server retourniert. Sollte eine URL einmal falsch eingegeben werden, werden andere Statuscodes gesendet, welche in der nachfolgenden Tabelle (Tab. 2.5.2) näher erläutert werden.¹⁷¹

¹⁷⁰vgl. Günster (2013), S.63 ff.

¹⁷¹vgl. ebd., S.68 f.

Statuscode:	Statusmeldung:	Bedeutung:
200	OK	Alles funktioniert – der Inhalt wird bereitgestellt
302	Found	Der Inhalt existiert zwar, aber nicht im angegebenen Verzeichnis. Der Header Location enthält die Information, wo die zu suchende Datei gefunden werden kann. Ein Redirect (d.h. Weiterleitung) wird eingeleitet.
304	Not Modified	Der Inhalt wurde zwar gefunden, existiert aber bereits in seiner neusten Version im Cache. Die Version des Inhalts wird ebenfalls im Header mitgesendet.
400	Bad Request	Der HTTP-Request war fehlerhaft. Dabei kann es sich entweder um einen falsch gesetzten Header bzw. Parameter oder um sonstige serverspezifische Probleme handeln.
401	Unauthorized	Nur angemeldete BenutzerInnen dürfen diesen Inhalt sehen
403	Forbidden	Für die Einsicht des Inhalts werden besondere Rechte erfordert. Dies kann auch passieren, wenn man bereits eingeloggt ist.
404	Not Found	Dieser Inhalt existiert nicht
500	Internal Server Error	Der Server hat ein Problem bei der Verarbeitung der Anfrage festgestellt (Am öftesten auftretender Statuscode bei der Entwicklung von PHP-Projekten)

Tabelle 2.5.2: HTTP-Statuscodes inkl. deren Bedeutung für BenutzerInnen ¹⁷²¹⁷² Günster (2013), S.71.

Kapitel 3

Abgrenzung zu anderen Technologien

Aufgrund der zustandslosen Natur des HTTP-Protokolls gelangt man bei der Erstellung dynamischer Webinhalte oder gar bei der Implementierung sicherheitstechnischer Aspekte sehr schnell an eine Art imaginäre Grenze, welche als nur schwer überwindbar gilt. Vor allem bei der Realisierung abhörsicherer Kommunikationskanäle bzw. Datentransfers innerhalb von Webprojekten ist die Wahl einer entsprechend abgesicherten Authentifizierungsmethode, wie beispielsweise durch ein Sicherheitstoken-System, unbedingt erforderlich, da Cookies und dergleichen im HTTP-Zustand über keinen Secure-Modus verfügen und somit von Hand für eine korrekte Verschlüsselung bzw. Privatsphäre gesorgt werden muss.

Beinahe in jeder Literatur wird interessanterweise grundsätzlich von eigenständig programmierten Authentifizierungssystemen abgeraten, wodurch meistens ein Kapitel rund um SSL und HTTPS von den jeweiligen AutorInnen verfasst wurde. Heutzutage würden WebentwicklerInnen ein solches System – gerade bei E-Commerce-Applikationen oder dergleichen – nicht mehr einsetzen und lieber auf den HTTPS-Port des SSL-Protokolls zurückgreifen. Gerade im Informatikunterricht stellt diese Form der Programmierung einen komplett neuen sicherheitstechnischen Zugang für SchülerInnen dar, welcher auch in etwaigen späteren Projekten ganz einfach umgesetzt werden kann und ebenfalls einen guten Schutz vor Angriffen oder Spionage bietet. Alleine aus diesem Grund sind Cookie-Verschlüsselungs- sowie Authentifizierungsverfahren für den Bildungseinsatz nicht auszulassen, sondern stellen vielmehr eine perfekte Abgrenzung zu SSL-verschlüsselten Kommunikationskanälen dar. Deswegen muss dieser Themenkreis auch unbedingt allen SchülerInnen näher gebracht werden.

Allerdings muss die Unterstützung von SSL und HTTPS vom Hosting Provider serverseitig gegeben sein, bevor man eine Webseite über den HTTPS-Port (443) aufrufen kann. Für Private, aber vor allem für Bildungsinstitutionen, stellt dies oftmals ein regelrechtes Problem dar, da beispielsweise nicht genug Geld aufgewendet werden kann, um

ein besseres Hosting Abonnement zu erwerben, welches mit SSL ausgestattet ist. Nichts desto trotz werden in den folgenden Unterkapiteln die Technologien des SSL-Protokolls und die des HTTPS-Ports erläutert und näher beschrieben, um eine fundamentale theoretische Grundlage über andere Sicherheitsmechanismen innerhalb des WWW und deren Spezifikationen zu schaffen.

3.1 SSL / HTTPS

Das SSL-Protokoll (d.h. „Secure Sockets Layer“) wurde im Jahr 1994 von Netscape entwickelt, welches sich damals durch den großen Marktanteil des Netscape Navigator Browsers weit verbreiten und somit fest etablieren konnte. Dabei arbeitet SSL keineswegs nur mit HTTP zusammen, sondern ist auch dazu in der Lage, mit jedem anderen Transportprotokoll zur sicheren Datenübertragung verwendet zu werden. Somit erweitert SSL die Socket-Schnittstelle um neue sicherheitstechnische Aspekte, welche andere Anwendungsprotokolle wie z.B. FTP oder POP3 problemlos nutzen können.¹⁷³ SSL garantiert dabei die Authentizität, Integrität sowie die Vertraulichkeit von Daten, wodurch das Protokoll genau zwischen dem Transport- und dem Applikationslayer angesiedelt ist. Aus diesem Grund ist SSL vom Anwendungsprotokoll weitgehend unabhängig einsetzbar, wodurch die IANA (d.h. „Internet Assigned Numbers Authority“), welche für die Zuordnung von Nummern und Namen im Internet zuständig ist, einige Protokolle bezüglich SSL im hauseigenen RFC (RFC 6335)¹⁷⁴ dokumentiert hat (Tab. 3.1.1).¹⁷⁵

Protokoll:	Port:	Beschreibung:
nsiops	261	IIOIP Name Service via SSL
https	443	HTTP via SSL
smtps	465	SMTP via SSL
nnntp	563	NNTP via SSL
ldaps	636	LDAP via SSL
ftps-data	989	FTP Datenverbindung via SSL
ftps	990	FTP Kontrollverbindung via SSL
telnets	992	Telnet via SSL
imaps	993	IMAP via SSL
ircs	994	IRC via SSL
pop3s	995	POP3 via SSL

Tabelle 3.1.1: Ports der unterschiedlichen SSL-Protokolle ¹⁷⁶

Prinzipiell basiert SSL auf dem sogenannten „Handshake-Protokoll“, welches sich direkt unter dem Applikationslayer befindet und dabei für die Authentifizierung sowie für die

¹⁷³vgl. Zacharias (2002), S.84.

¹⁷⁴s. Internet Assigned Numbers Authority: Service Name and TCP Port Number Registry, (Stand: 18.10.2016).

¹⁷⁵vgl. Fuchs (2001), S.99.

¹⁷⁶ebd., S.100.

Wahl des zu verwendeten Verschlüsselungs- und Komprimierungsalgorithmus zuständig ist (Abb. 3.1.1).

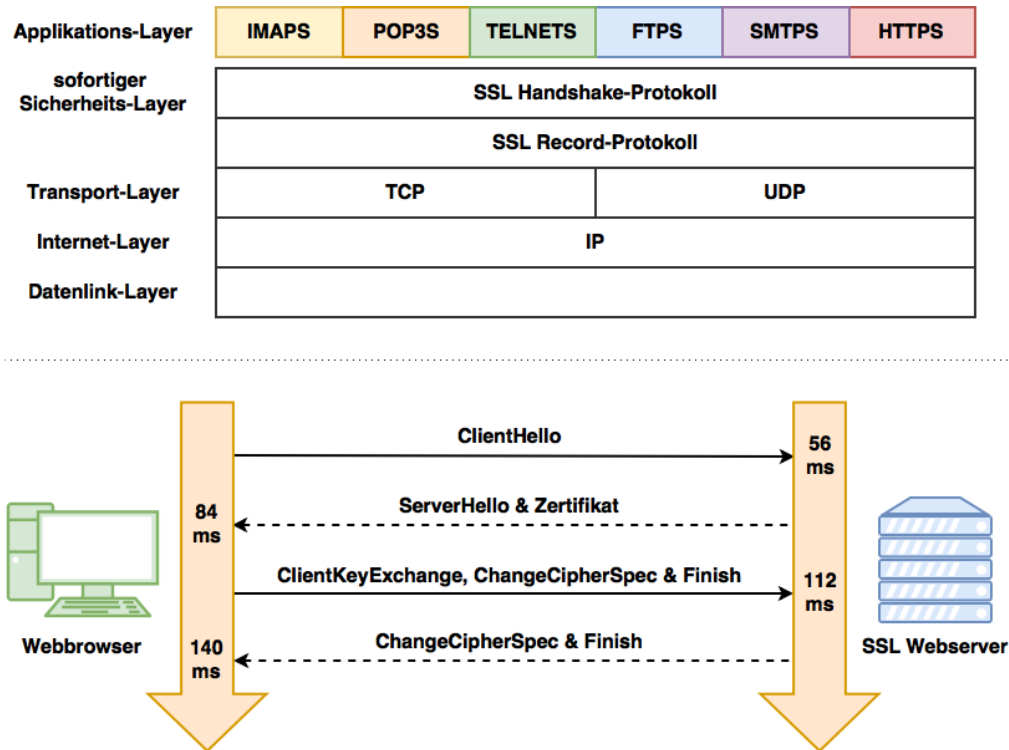


Abbildung 3.1.1: SSL Architektur inkl. Protokollablauf

Um nun innerhalb eines abhörsicheren Kanals zu kommunizieren, übermittelt der Client dem Server ein sogenanntes „ClientHello“, bei dem der Client folgende Informationen mitteilt:

1. Die SSL Version, die verwendet werden soll
2. Einen 32-Bit Zeitstempel
3. Eine 28 Byte große Zufallszahl
4. Eine Session-ID, welche beim ersten Protokollaufbau leer ist
5. Eine Liste der unterstützten Verschlüsselungsalgorithmen
6. Eine Liste der unterstützten Komprimialgorithmen

Daraufhin antwortet der Server mit einem „ServerHello“, welches grundsätzlich dieselben Informationen eines „ClientHello“ enthält, außer dass die Session-ID bereits definiert und der Verschlüsselungs- sowie Komprimialgorithmus mittlerweile ausgewählt wurde. Bei der Auswahl des zu verwendenden Algorithmus orientiert sich der Server an der Liste des Clients, welche nach Priorität geordnet ist. Wird kein einziger der vorgeschlagenen Algorithmen serverseitig unterstützt, so wird ein Error an den Client retourniert. Anschließend

wird vom Server ein Zertifikat mitgesendet, welches üblicherweise im „X.509v3“ Format vorliegt.¹⁷⁷ Dabei gliedert sich das Zertifikat in zwei Hauptteile:

- Zertifizierungsinformationen (z.B. Name des Clients, Adresse, Public Key, Gültigkeitszeitraum des Zertifikats, Zertifikat-ID und Name des Servers)
- Digitale Unterschrift der „Certification Authority“ (wird aus den Zertifizierungsinformationen mit Hilfe des Private Keys der „Certification Authority“ generiert)

Das signierte digitale Zertifikat wird anschließend wieder an den Client zurückgesendet. Angenommen der Client und der Server sind nun im Besitz eines gültigen Zertifikats, eines Public Keys und eines Private Keys, kann der Client nun eine sichere Botschaft an den Server schicken. Dabei wird im vereinfachten Sinne die Nachricht des Clients mit dem eigenen Private Key und dem Public Key des Servers verschlüsselt und mit dem servereigenen Private Key und dem Public Key des Clients wieder entschlüsselt. Somit kombiniert SSL die Vorteile der symmetrischen (d.h. ein Schlüssel) und der asymmetrischen (d.h. zwei Schlüssel) Verschlüsselung, da 128-Bit Schlüssel, welche typisch für symmetrische Verschlüsselungsmethoden sind, bis heute als absolut abhörsicher gelten.¹⁷⁸

Wurde nun ein Zertifikat erfolgreich übermittelt, kann der Client überprüfen, ob es sich bei der im Zertifikat angegebenen URL auch um die des Servers handelt und ob es schlussendlich von einer dem Client bekannten Autorität signiert wurde. Trifft dies alles zu, generiert der Client ein 48 Byte großes sogenanntes „Pre-Master-Secret“ und verschlüsselt dieses mit dem Public Key des Servers. Dieses „Pre-Master-Secret“ wird dann mittels der „ClientKeyExchange“-Nachricht an den Server geschickt bevor die „ChangeCipherSpec“- und die „Finished“-Meldung an den Server gehen. Abschließend schickt der Server dann ebenfalls noch eine „ChangeCipherSpec“- und eine „Finished“-Nachricht an den Client (*Abb. 3.1.1*). Der Sessionschlüssel, mit dem die Kommunikation schlussendlich verschlüsselt wird, bildet sich aus dem Pre-Master-Secret, der Zufallszahl des „ClientHello“ und der Zufallszahl des „ServerHello“. Da sich diese Form des Protokolls in der Praxis als eher aufwendig erwies, sieht der SSL-Standard auch eine abgekürzte Version des „Handshake-Protokolls“ vor. Dabei kann sofort mit dem „ChangeCipherSpec“-Kommando fortgefahren werden, sofern die beim „ClientHello“ mitgesendete Session-ID dem Server bereits bekannt ist. Des Weiteren sieht SSL vor, dass sich der Client gegenüber einem Server authentifizieren kann. Dafür schickt der Server ein sogenanntes „CertificateRequest“, nachdem ein Zertifikat erfolgreich übertragen wurde. In der Folge werden die zu übertragenden Daten in einzelne Fragmente zerlegt, die maximal je 16383 Byte groß sein dürfen. Dabei enthält ein „SSL Record“ folgende Daten:

¹⁷⁷vgl. *Fuchs* (2001), S.100 f.

¹⁷⁸vgl. *Zacharias* (2002), S.85 ff.

1. Content-Type (welches Protokoll wird verschlüsselt)
2. Protokollversionsnummer von SSL
3. Länge des Records
4. Verschlüsselte und komprimierte Daten
5. Einen MAC (d.h. „Message Authentication Code“), der eine Hashsumme der verschlüsselten Daten einschließlich einer Sequenznummer enthält, welche Replay-Attacken verhindern soll.¹⁷⁹

3.2 Vor- & Nachteile

Obwohl SSL als generell sehr sicher gilt, gab es in der Vergangenheit schon mehrmals Berichterstattungen, inwiefern SSL dennoch für Manipulation oder gar Spionage anfällig sein kann. So wurde beispielsweise bestätigt, dass um 2010 SSL gesicherte Kommunikationskanäle standardmäßig von staatlichen Einrichtungen abgehört wurden, während im Jahr 2011 sogar ein Herausgeber von Zertifikaten kompromittiert wurde, wodurch Unbefugte Zugriff auf noch gültige SSL-Zertifikate erhielten. Aber selbst als im Jahr 2014 der sogenannte „Heartbleed“-Bug entdeckt wurde, über welchen mehrere Daten zugleich ausgelesen werden konnten, erlebte die SSL-Welt eine erneute Erschütterung ihrer Standfeste. Wie man sieht, steht vor allem die Herausgabe und die Fälschung von Zertifikaten bei einer SSL-Verbindung in Kritik. Aufgrund der vielen negativen Kriterien trägt SSL mittlerweile vielmehr zur Bequemlichkeit als zur Sicherheit innerhalb des WWW bei, wodurch Aktionen wie Online-Shopping eine weitgehend erhöhte Zeitersparnis bringen. Dabei dient die SSL verschlüsselte Kommunikation den Nutzern zusätzlich als Hinweis für einen vertrauenswürdigen Anbieter. Je mehr Vertrauen man den potenziellen BenutzerInnen entgegenbringt, desto eher kann man diese schlussendlich via Registrierung dauerhaft binden. Von der technischen Seite aus gesehen ist es außerdem ein immens großer Vorteil, dass SSL unabhängig von der Version des Betriebssystems arbeitet. Der verwendete Browser ist demnach nahezu irrelevant und es wird keine zusätzliche Software benötigt. Somit kann jeder ohne Einschränkung eine Verbindung mit SSL-Verschlüsselung aufbauen.¹⁸⁰ Interessanterweise werden Webseiten mit SSL-Verschlüsselung allmählich auch besser bei Google gerankt als herkömmliche Internetseiten. Somit verlieren unsichere Seiten immer mehr an Traffic, während SSL gestützte Seiten bessere Aussichten auf Erfolg haben, was einen weiteren wichtigen Vorteil dieser Technologie darstellt.¹⁸¹

Problematisch wird es bei SSL allerdings, wenn man neben einer TCP-Unterstützung auch eine für UDP sucht. Hier wird man leider nicht fündig werden. Außerdem werden auch keine digitalen Signaturen an den eigentlichen Inhalt angehängt, was ebenfalls ein relativ

¹⁷⁹ vgl. *Fuchs* (2001), S.101 f.

¹⁸⁰ s. *OnPageWiki: SSL*, (Stand: 18.10.2016).

¹⁸¹ s. *Blogstories: Vor- und Nachteile einer SSL-Verschlüsselung*, (Stand: 18.10.2016).

großer Nachteil dieser Technologie ist. Der Client kann zwar dadurch beweisen, dass die Übertragung nicht manipuliert wurde, hat aber keinerlei Informationen darüber, ob die Daten auch wirklich vom richtigen Server stammen. Gerade bei E-Commerce-Systemen wäre eine solche Beweisbarkeit sehr von Vorteil, da sonst ein Vertragspartner ein Angebot leicht abstreiten könnte. Des Weiteren bietet SSL nur einen sehr marginalen Schutz gegen sogenannte „Man-in-the-Middle“ Attacken. Dadurch, dass man auch selbst „Certification Authorities“ erstellen kann, ist es Angreifern möglich, für den Serverschlüssel selbst ein Zertifikat zu erstellen. Wenn die BenutzerIn den Warnhinweis einfach ignoriert und auf „OK“ klickt, wird die „Certification Authority“ des Angreifers ohne Weiteres akzeptiert. Man könnte außerdem die „Certification Authority“ im Browser der Nutzer eintragen, wodurch der Browserschlüssel, welcher dieses Zertifikat enthält, anstandslos akzeptiert würde. Viele Quellen berichten neben dem altbekannten finanziellen Problem außerdem von einer schwachen User-Experience, welche unter SSL zu erfahren wäre, da beispielsweise die Ausgabe von Webseiteninhalten darunter leidet.¹⁸² Das größte Problem stellt allerdings die ständige Widerrufung von SSL-Zertifikaten dar, wenn z.B. der Serverschlüssel oder ähnliches kompromittiert wurde. Ein derartiger Widerruf erfolgt jedoch nur bei jener „Certification Authority“, welche den Schlüssel in Form einer CRL (d.h. „Certificate Revocation List“) zertifiziert hat. Clients müssten diese Listen normalerweise regelmäßig herunterladen, um am aktuellen Stand zu bleiben. Dies wird jedoch nur von den wenigsten wegen der damit verbundenen Strapazen so gehandhabt.¹⁸³

¹⁸²s. *Blogstories*: Vor- und Nachteile einer SSL-Verschlüsselung, (Stand: 18.10.2016).

¹⁸³vgl. *Fuchs* (2001), S.102.

Kapitel 4

Relevanz für den Unterricht

Security im Webdatenbereich stellt eines der zentralen Themen unserer Zeit dar. Dabei ist das Interesse sicherlich nicht nur auf den Informatikunterricht beschränkt, sondern erzielt auch ein großes mediales sowie gesellschaftliches Interesse. Insbesondere durch die enorme Nutzung des Internets an sich aber auch durch das zahlreiche Angebot an unzähligen Webdiensten, entstehen neue Wege für Menschen alltägliche Dinge zu erledigen, mit denen allerdings auch neue Wege der Onlinepiraterie einhergehen. Allerdings werden nicht nur auf illegalem Wege Daten gesammelt. Auch Großkonzerne wie Facebook und Co sind ebenfalls kräftig am Datensammeln, welche durch den anschließenden Informationsverkauf lukratives Geld wert sind. Somit betrifft Datensicherheit jeden Menschen, der in irgendeiner Form mit dem Medium Internet verbunden ist.

Trotz der Relevanz von Datenschutz und Privatsphäre im „Onlinealltag“ werden moderne Formen der Webentwicklung, sowie deren sicherheitstechnische Aspekte im Informatikunterricht bisher jedoch allenfalls am Rand behandelt. So passiert es oftmals, dass Lehrende lediglich über die Gefahren von Cookies sprechen und dabei nicht genau aufzeigen, um welche Daten es sich dabei handeln kann, die gespeichert werden sollen. Dabei wird auch vollkommen außer Acht gelassen, dass dynamisch entwickelte Webseiten ohne Cookies gar nicht mehr auskommen würden und nicht alle gesetzten Cookies von Grund auf bössartig sind. Aus diesem Grund wird das Thema Datensicherheit oft zu besseren Facebook, Twitter oder Amazon „Guideline-Stunden“ degradiert, bei denen SchülerInnen weder die genauen Anwendungsmöglichkeiten noch die positiven Seiten dieser Internetressource erfahren können.

Selbst in der universitären Lehre beschäftigt sich nur ein geringer Prozentsatz der Lehrveranstaltungen mit dem Thema Datensicherheit bzw. Datenintegrität. Auch hier wird die Thematik rund um Cookies und deren artgerechter Verschlüsselung mehr oder weniger außer Acht gelassen, wodurch eines der wichtigsten Internettools überhaupt an informatischen Fakultäten unbesprochen und nicht diskutiert bleibt. Allerdings sollte dieser

Themenschwerpunkt schon bei weitem früher angesetzt werden, um SchülerInnen sowie spätere StudentInnen für Cookies rechtzeitig zu sensibilisieren und etwaige Mythen darum auszumerzen. Daher werden in den folgenden Unterkapiteln zu Beginn die rechtlichen Rahmenbedingungen anhand des österreichischen Schulorganisationsgesetzes, Schulunterrichtsgesetzes sowie der Leistungsbeurteilungsverordnung erarbeitet, um diese für einen eventuellen Unterrichtseinsatz innerhalb der Sekundarstufe II aufzuschlüsseln. Abschließend werden dann – unter Berücksichtigung der Rechtsaspekte – die Ergebnisse in den Lehrplan des Unterrichtsfachs Informatik übergeleitet, um somit die thematische Relevanz für das Pflichtfach (5. Klasse) sowie das Wahlpflichtfach (6. - 8. Klasse) Informatik aufzuzeigen.

4.1 Rechtliche Rahmenbedingungen

Die juristische Absicherung innerhalb von Bildungsinstitutionen erfolgt im Wesentlichen über drei Säulen der Gesetzgebung. Diese untergliedern sich einerseits in Gesetze und Verordnungen (z.B. SchOG, SchUG, LBVO, Schulzeitgesetz, Schulpflichtgesetz usw.), Gesetzesnovellen, neue Lehrpläne sowie sonstige Verordnungen, welche allesamt vom BMB (d.h. Bundesministerium für Bildung) erlassen wurden.¹⁸⁴ Seit Mai 2016 befindet sich das BMB unter der Leitung von Frau Dr. Sonja Hammerschmid, welche erst kürzlich ein eher umstrittenes Bildungsreformpaket für mehr Autonomie innerhalb von Bildungseinrichtungen und deren Direktionen vorgelegt hat.¹⁸⁵

Im Folgenden wird nun anhand der aktuellen Gesetzeslage des Schulorganisationsgesetzes, des Schulunterrichtsgesetzes sowie der Leistungsbeurteilungsverordnung analysiert, inwiefern die Thematik „Security im Webdatenbereich“ durch die bildungstechnischen Rechtsaspekte der Republik Österreich gedeckt sind und ob Cookies darin überhaupt eine wesentliche Rolle spielen können.

4.1.1 SchOG - Schulorganisationsgesetz

Mit dem Schulgesetzwerk von 1962, dessen Hauptstück das Schulorganisationsgesetz darstellt, wurde das Schulwesen der Zweiten Republik auf eine neue Basis gestellt, die mit Modifizierungen durch eine Vielzahl von Novellen bis heute Gültigkeit hat.¹⁸⁶

Wie der Name bereits zum Ausdruck bringt, behandelt dieses Gesetz die allgemeine Organisation von Bildungseinrichtungen, welche in 133 Paragraphen geregelt ist. Dieses Bundesgesetz gilt für die allgemeinbildenden und berufsbildenden Pflichtschulen

¹⁸⁴s. *Bundesministerium für Bildung*: Schulrecht Überblick, (Stand: 20.10.2016).

¹⁸⁵s. *Der Standard*: Bildungsreform - Mehr Macht für Direktoren, (Stand: 20.10.2016).

¹⁸⁶s. *Loew*: Das Schulorganisationsgesetz 1962, (Stand: 20.10.2016).

sowie mittleren und höheren Schulen. Ausgenommen vom Geltungsbereich dieses Bundesgesetzes sind die land- und forstwirtschaftlichen Schulen (vgl. §1 SchOG). Dabei umfasst das SchOG nur einen sehr kleinen Aufgabenbereich, welcher sich z.B. in der Organisation der Lehrpläne (vgl. §6 SchOG), Modell- oder Schulversuche (vgl. §7, 7a SchOG), Ganztagschulbetrieb (vgl. §8d SchOG), Klassenschülerzahlen (vgl. §14 SchOG) usw. bemerkbar macht.¹⁸⁷

Innerhalb dieses Bundesgesetzes wird allerdings kein Platz geboten, um fachdidaktische Unterrichtskonzepte bzw. deren kompetenzorientierten Aufbereitung für eine oder mehrere Unterrichtseinheiten zu regulieren, wodurch diese Säule der Legislative keinen Aufschluss über die Relevanz von Webdatensicherheit im Informatikunterricht gibt.

4.1.2 SchUG - Schulunterrichtsgesetz

Das Schulunterrichtsgesetz stellt ein weiteres Bundesgesetz der Republik über die Ordnung von Unterricht und Erziehung, in den vom Schulorganisationsgesetz geregelten Schulformen, mit Ausnahme der in Semester gegliederten Sonderformen (vgl. §1 SchUG). Interessanterweise werden hier – ganz im Gegensatz zum SchOG – die land- und forstwirtschaftlichen Schulen zu den höheren bzw. mittleren berufsbildenden Schulen gezählt. Dieses Bundesgesetz regelt unter anderem die innere Ordnung des Schulwesens als Grundlage des Zusammenwirkens von Lehrenden, Lernenden und Erziehungsberechtigten als eine gemeinsame Schulgemeinschaft (vgl. §2 SchUG).¹⁸⁸ Darüber hinaus werden noch weitere allgemeine Informationen über die Schule per se (2. Abschnitt SchUG), die Rechte und Pflichten der SchülerInnen (vgl. §57a, 43 SchUG) oder Erziehungsberechtigten (vgl. §61 SchUG), sowie weiterführende Regelungen zum Schulgemeinschaftsausschuss bereitgestellt (vgl. §64 SchUG).¹⁸⁹

Für diese Diplomarbeit und im Zusammenhang mit der vorhin gestellten Frage, stellt aber vor allem §17 SchUG „Unterrichtsarbeit“ den bedeutsamsten aller Paragraphen dar. Allerdings sind auch §57b (Abs. 3) SchUG „Schülerinnen- und Schülerkarte“, sowie §77 SchUG „Klassenbücher“ und §77a SchUG „Formblätter und Datenmuster; Aufbewahrung von Protokollen und Aufzeichnungen“ von beispielhafter Relevanz, da diese grundsätzlichen Aufschluss darüber geben, inwiefern mit den direkt personenbezogenen Daten der SchülerInnen innerhalb von Bildungseinrichtungen umgegangen werden muss.

Innerhalb der Unterrichtsarbeit sind Lehrende dazu angehalten, die Aufgabe der österreichischen Schule, welche innerhalb von §2 SchOG geregelt ist, zu erfüllen. Dabei

¹⁸⁷ s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Schulorganisationsgesetz 1962, (Stand: 20.10.2016).

¹⁸⁸ s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Schulunterrichtsgesetz 1986, (Stand: 20.10.2016).

¹⁸⁹ s. *Schülerunion*: Schulunterrichtsgesetz 1986, (Stand: 20.10.2016).

wird ausdrücklich auf die Verwendung des entsprechenden Lehrplanes je nach Schulart, sowie auf die Einbindung äußerer Gegebenheiten hingewiesen, um nach Stand der Wissenschaft entsprechend vermitteln und somit den Unterricht gegenwartsbezogen gestalten zu können. In diesem Sinne soll durch geeignete Methoden und zweckmäßigen Einsatz von Unterrichtsmitteln eine fundierte Grundlage geschaffen werden, welche es mittels Übungen zu festigen gilt.¹⁹⁰ Innerhalb der Klasse spielt an dieser Stelle die Lehrperson eine eigenständige und verantwortungsvolle Rolle. Nichtsdestotrotz kann dabei eine Vielzahl an vermeintlichen Parallelen über die Wichtigkeit von Webdatensicherheit im Unterricht gezogen werden. Die äußeren Gegebenheiten werden beispielsweise durch die flächendeckende Nutzung von Cookies innerhalb moderner Webapplikationen abgesichert, wodurch sich in weiterem Sinne eine Verbindung zu „Social Networks“ und dergleichen ergeben würde, da diese ohne den Einsatz von Cookies ebenfalls nicht funktioniert. Durch das gemeinsame Erarbeiten eines selbst programmierten PHP Session basierten Login-Systems mit Salted-Hash-Cookies wäre abschließend noch die Methode sowie die Übung zur Festigung des Erlernten abgedeckt.

Aus datenschutzrechtlicher Sicht sind die Regelungen rund um Schülerinnen- und Schülerkarten, Klassenbücher sowie die Aufbewahrung von Protokollen und Aufzeichnungen von besonderer Bedeutung. Diese enthalten nämlich allesamt direkt bzw. indirekt personenbezogene Daten von SchülerInnen, welche somit unter einer besonderen Sorgsamkeitspflicht stehen. Solche Datensammlungen sind für Bildungsinstitutionen allerdings unausweichlich, da SchülerInnenkarten z.B. zum Nachweis der Zugehörigkeit an der betreffenden Schule unbedingt benötigt werden. Diese beinhalten im Regelfall die Bezeichnung der Schule, den Vor- sowie Nachnamen und ein Lichtbild der SchülerIn, deren bzw. dessen Geburtsdatum sowie das Ausstellungsdatum. Dabei gilt das Verlangen einer solchen SchülerInnenkarte sowie die Zustimmung als Zustimmung im Sinne des §4 (Abs. 14) DSGVO, wodurch die Schule ermächtigt wird, solche Daten zu erheben.¹⁹¹

Selbst bei Klassenbüchern – vor allem jene, die elektronisch geführt werden – müssen Abfrageberechtigungen eingeräumt und Einsichts- oder Zugriffsmöglichkeiten für andere Personen als dem an der Schule tätigen Lehr- und Verwaltungspersonal gesperrt werden. Dabei sind Datensicherheitsmaßnahmen gemäß §14 DSGVO zu treffen und die Bestimmungen des §15 DSGVO über das Datengeheimnis anzuwenden. Diese Bücher müssen, unter der Beachtung der Zugriffsbeschränkungen und Datensicherheitsmaßnahmen, drei Jahre ab dem Ende des letzten Schuljahres des betreffenden Jahrgangs an der Schule und somit gegebenenfalls am schulinternen Server aufbewahrt werden. Nach Ablauf dieser Aufbewahrungsfrist sind physische Klassenbücher zu vernichten und elektronisch

¹⁹⁰s. *Jusline*: §17 (Abs. 1) SchUG 1986 Unterrichtsarbeit, (Stand: 20.10.2016).

¹⁹¹s. *Jusline*: §57b SchUG 1986 Schülerinnen- bzw. Schülerkarte, (Stand: 20.10.2016).

gespeicherte Klassenbuchaufzeichnungen dauerhaft zu löschen.¹⁹²

Abschließend werden auch die Prüfungsprotokolle samt Beilagen sowie die Protokolle schulinterner Vorgänge zum Zweck der Dokumentation für behördliche Verfahren aufbewahrt. Die Form bzw. das Datenmuster für diese Protokolle stammen dabei allesamt von den Landesschulräten, welche bei Prüfungsprotokollen unter anderem die Namen der Prüfungskommission, die Daten der Prüfungskandidatin oder des Prüfungskandidaten, die Aufgabenstellungen, die Beschreibung der Leistungen und ihre Beurteilung, die Prüfungsergebnisse und die bei der Prüfung oder auf Grund der Prüfungsergebnisse getroffenen Entscheidungen und Verfügungen enthalten. Dadurch, dass es sich dabei ebenfalls um direkt bzw. indirekt personenbezogene Daten handelt, sind Prüfungs- und sonstige Schulprotokolle, unter Beachtung der Zugriffsbeschränkungen und Datensicherheitsmaßnahmen, drei und in manchen Fällen sogar bis zu sechzig Jahre aufzubewahren.¹⁹³

4.1.3 LBVO - Leistungsbeurteilungsverordnung

Die Leistungsbeurteilungsverordnung stellt eine Verordnung des Bundesministers für Unterricht und Kunst vom 24. Juni 1974 über die Leistungsbeurteilung in Pflichtschulen sowie mittleren und höheren Schulen dar. Grundlage der Leistungsbeurteilung sind die Leistungsfeststellungen nach Maßgabe aller Bestimmungen dieser Verordnung (vgl. §1 LBVO). Dabei kann der Leistungsfeststellung nur die im Lehrplan festgelegten Bildungs- und Lehraufgaben und jene Lehrstoffe zugrunde gelegt werden, die bis zum Zeitpunkt der Leistungsfeststellung in der betreffenden Klasse behandelt worden sind (vgl. §2 LBVO).¹⁹⁴ Darüber hinaus werden noch weitere allgemeine Informationen über Schularbeiten (vgl. §7 LBVO), mündliche Prüfungen (vgl. §5 LBVO) oder Hausübungen (vgl. §4 LBVO), sowie weiterführende Regelungen zu Betragensnoten bereitgestellt (vgl. §18 LBVO).¹⁹⁵

In Bezug auf diese Diplomarbeit stellt aber lediglich §3 (Abs. 2) LBVO einen mehr oder weniger relevanten Paragraphen dar, in welchem die Einbeziehung praktischer und graphischer Arbeitsformen – wie z.B. die Arbeit am Computer – lediglich als „zulässig“ betrachtet wird.¹⁹⁶

¹⁹²s. *Jusline*: §77 SchUG 1986 Schulverzeichnisse, Protokolle, Formblätter, (Stand: 20.10.2016).

¹⁹³s. *Jusline*: §77a SchUG 1986 Formblätter und Datenmuster, (Stand: 20.10.2016).

¹⁹⁴s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für LBVO 1974, (Stand: 20.10.2016).

¹⁹⁵s. *Schülerunion*: Leistungsbeurteilung, (Stand: 20.10.2016).

¹⁹⁶s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für LBVO 1974, (Stand: 20.10.2016).

4.2 Lehrpläne Unterrichtsfach Informatik

Wie vom Schulunterrichtsgesetz bereits festgelegt stützt sich die Unterrichtsarbeit aller Lehrenden auf die fachspezifischen Lehrpläne. Die Oberstufenlehrpläne, welche allesamt vom BMB ausgestellt werden, wurden im Juli 2004 in einer neuen Auflage kund gemacht und traten aufsteigend ab dem Schuljahr 2004/05 in Kraft, sodass mit dem Schuljahr 2007/08 die veralteten Lehrpläne zur Gänze durch die neuen ersetzt wurden. Die neuen Lehrpläne wurden im Zuge einer AHS-Oberstufenreform erarbeitet, wodurch die ersten drei allgemeinen Teile der Pläne grundlegend überarbeitet wurden. Diese Teile untergliedern sich in das allgemeine Bildungsziel (1. Teil), die allgemeinen didaktischen Grundsätze (2. Teil), sowie die Schul- und Unterrichtsplanung (3. Teil), welche durch spezifische Aussagen für die Oberstufe ergänzt wurden und somit für die Sekundarstufe I sowie die Sekundarstufe II gültig sind. Die Stundentafeln, welche den vierten Teil eines Lehrplans ausmachen, wurden bereits im Juni 2003 geändert, wodurch Autonomiestundentafeln sowie subsidiäre „nichtautonome“ Stundentafeln hinzukamen.¹⁹⁷



Abbildung 4.2.1: „Word Cloud“ über den Lehrstoff des Unterrichtsfachs Informatik

Das allgemeine Bildungsziel definiert gemeinsam mit den allgemeinen didaktischen Grundsätzen und dem Teil „Schul- und Unterrichtsplanung“ Verbindlichkeiten, Verantwortlichkeiten und Freiräume bei der Umsetzung des Lehrplans, während die Stundentafeln Unterrichtsgegenstände benennen und das Stundenausmaß der Unterrichtsgegenstände angeben. In den Lehrplänen der einzelnen Unterrichtsgegenstände wird weiters die sogenannte „Bildungs- und Lehraufgabe“ festgelegt, welche sich sowohl auf den Kern- als auch auf den Erweiterungsbereich bezieht. Bezüge werden außerdem zum allgemeinen

¹⁹⁷s. Bundesministerium für Bildung: Lehrpläne der AHS-Oberstufe Überblick, (Stand: 21.10.2016).

Bildungsziel und insbesondere als Beiträge zu den Bildungsbereichen angeführt. Im Abschnitt „didaktische Grundsätze“ werden Anleitungen zur Gestaltung des Unterrichts gegeben und im Abschnitt „Lehrstoff“ werden die zu erreichenden Ziele bzw. Inhalte festgelegt. Geht es nach der aktuellen Gesetzeslage, so definiert sich ein Lehrplan einerseits aus der Einheitlichkeit und Durchlässigkeit der notwendigen Vorgaben des Schulwesens und eröffnet andererseits Freiräume, die der Konkretisierung am Standort vorbehalten sind. In diesem rechtlichen Kontext dient der Lehrplan demnach als Grundlage für:

- Konkretisierung des Erziehungsauftrags der Schule
- Inhaltliche und methodische Planung und Steuerung des Unterrichts
- Gestaltung des Erweiterungsbereichs und schulautonomer Lehrpläne
- Planungen der schulpartnerschaftlichen Gremien
- Standortbezogenes Bildungsangebot
- Berücksichtigung der individuellen Interessen und persönlichen Lebensrealität der SchülerInnen¹⁹⁸

Um allerdings den Unterrichtseinsatz von Security im Webdatenbereich besser analysieren sowie beurteilen zu können, werden nun die diversen informatikrelevanten Lehrpläne durchbesprochen, damit die Wichtigkeit dieser Thematik je nach Schulstufe aufgezeigt werden kann.

4.2.1 Sekundarstufe II

In der Sekundarstufe II, welche ab der 9. Schulstufe beginnt und somit den Beginn der Oberstufe symbolisiert (*Abb. 4.2.2*), wird das Unterrichtsfach Informatik zum ersten Mal als Pflichtfach angeführt. So gibt es auch für die Sekundarstufe II den ersten richtigen Lehrplan, welcher erst einen kompetenzorientierten Informatikunterricht ermöglicht. Dabei ist der Lehrplan für die AHS-Oberstufe nach dem folgenden Muster aufgebaut:

- Gegenstandsbezeichnung: Pflichtgegenstand oder Wahlpflichtgegenstand
- Bildungs- und Lehraufgabe: Selbstverständnis des Gegenstandes (Hauptziele, welche die SchülerInnen im jeweiligen Fach erreichen sollen)
- Didaktische Grundsätze: fachspezifische Hinweise und Grundsätze für die eingesetzte Methodik und Didaktik usw.
- Lehrstoff: Hier wird in verbindlichen Vorgaben formuliert, welche Lernziele die SchülerInnen erreichen sollen (geteilt nach Schulstufen bzw. nach fachspezifischen Gliederungselementen). Die zeitliche Gewichtung und die konkrete Umsetzung der Vorgaben obliegen jedoch den Lehrkräften.¹⁹⁹

¹⁹⁸ s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Lehrpläne - AHS 1984, (Stand: 21.10.2016).

¹⁹⁹ s. *Bundesministerium für Bildung*: Lehrpläne der AHS-Oberstufe Überblick, (Stand: 21.10.2016).

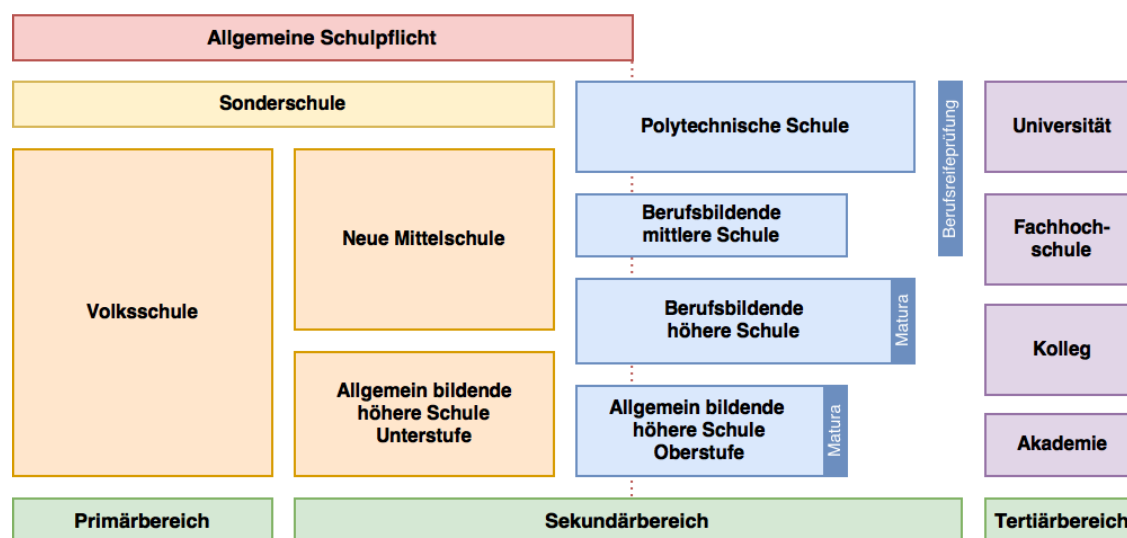


Abbildung 4.2.2: Das österreichische Schulsystem

Geht es nach dem BMB, so verkörpert Bildung mehr als die Summe des Wissens, welches in den einzelnen Unterrichtsgegenständen erworben werden kann. Aus diesem Grund wurden insgesamt fünf Bildungsbereiche geschaffen wie z.B. „Sprache und Kommunikation“, „Mensch und Gesellschaft“ oder „Kreativität und Gestaltung“, welche als Benennung wichtiger Segmente im Bildungsprozess zu verstehen sind. In den Bildungsbereichen sind ausformulierte Zielsetzungen enthalten, die von vielen Unterrichtsprinzipien wie beispielsweise „Wirtschaftserziehung“, „Erziehung zur Anwendung neuer Technologien“ oder „Vorbereitung auf die Arbeits- und Berufswelt“ vertreten werden. Für diese Arbeit stellt allerdings der dritte Bildungsbereich „Natur und Technik“ den wohl Bedeutsamsten dar. Hier dient die Natur als Grundlage des menschlichen Lebens, welche in vielfältiger, auch technisch veränderter Gestalt in Erscheinung treten kann. So wird etwa die Vermittlung der Kenntnisse über die Wirkungszusammenhänge der Natur für einen bewussten Umgang mit Hilfe der modernen Technik vorausgesetzt, wodurch die Bereiche Mathematik, Naturwissenschaften und Technik eine Orientierungsgrundlage in einer modernen, von Technologien geprägten Gesellschaft bilden. Daher soll der Unterricht grundlegendes Wissen, Entscheidungsfähigkeiten und Handlungskompetenzen vermitteln, welche die SchülerInnen dazu befähigt, sich mit ethischen Fragen im Zusammenhang mit Natur und Technik sowie Mensch und Gesellschaft auseinanderzusetzen.²⁰⁰

Zusätzlich zu den definierten Bildungsbereichen wurden die allgemeinen didaktischen Grundsätze in insgesamt neun Punkten wie z.B. „Interkulturelles Lernen“, „Stärken von Selbsttätigkeit und Eigenverantwortung“ oder „Bewusste Koedukation und Geschlechtssensible Pädagogik“ neu formuliert. In Bezug auf das Thema „Security im Webdatenbereich“ ist der didaktische Grundsatz „Herstellen von Bezügen zur Lebenswelt“

²⁰⁰s. Bundesministerium für Bildung: Verordnung über die Änderung der AHS Lehrpläne 2004, (Stand: 21.10.2016).

von besonderem Interesse, da im Sinne des exemplarischen Lernens möglichst zeit- und lebensnahe Themen gewählt werden sollten. Die Materialien und Medien, welche im Unterricht eingesetzt werden, sollten auch möglichst aktuell und anschaulich sein, um die SchülerInnen zu aktiver Mitarbeit anzuregen. Diese Voraussetzungen werden bei Themen wie Cookies, Datensicherheit oder moderne Webtechniken absolut erfüllt und ist somit vollkommen didaktisch begründet. Das BMB weist sogar nochmals explizit auf neue Technologien (u.a. elektronische) hin, welchen eine verstärkte Bedeutung zukommen sollte. Dies gilt in besonderem Maße für die Oberstufe der AHS, da hier in allen Gegenständen Informationsmanagement sowie Lern- und Unterrichtsorganisation mit Mitteln der Informationstechnologie zu praktizieren sind. Dabei sind in kooperativen Arbeitsformen Informationsquellen zu erschließen und unterschiedliche Informationsformen zu bearbeiten, Inhalte zu systematisieren und zu strukturieren. Die Inhalte und deren Interpretation sind stets kritisch zu hinterfragen, um Auswirkungen auf den Einzelnen und die Gesellschaft zu reflektieren. Schlussendlich soll auch die Gestaltung, Modellierung und Erstellung eigenständiger Arbeiten mit Mitteln der Informationstechnologie ermöglicht werden, wodurch sich ein selbst programmiertes Salted-Hash-Cookie gestütztes PHP-Session-Login-System ideal für den Unterrichtseinsatz anbietet.²⁰¹

Neben den didaktischen Grundsätzen entwickelten sich um 2013 auch die digitalen Kompetenzen für die sogenannte „Informatische Bildung“, wodurch Grundkompetenzen für die Themenfelder „Informationstechnologie, Mensch und Gesellschaft“, „Informatiksysteme“, „Angewandte Informatik“ und „Praktische Informatik“ definiert wurden. Durch die zu behandelnden Inhalte wie z.B. Cookies oder Webdatensicherheit finden sich in den Kompetenzmodellen für das Pflichtfach Informatik (5. Klasse)²⁰² sowie das Wahlpflichtfach Informatik (6. - 8. Klasse)²⁰³ sehr viele kompetenzorientierte Überschneidungen, welche es für die SchülerInnen im Laufe des Unterrichtskonzepts zu erwerben gilt. Um allerdings nicht vorzugreifen, werden die vorhin angesprochenen Überschneidungen der Reihe nach in den jeweiligen Szenarien im empirischen Teil dieser Arbeit behandelt.

4.2.1.1 Pflichtfach Informatik (5. Klasse)

Informatik tritt in der AHS-Oberstufe zum ersten Mal in Form eines Pflichtgegenstandes auf. Obwohl das Thema „Security im Webdatenbereich“ nicht explizit im Lehrplan angeführt und das Curriculum an sich sehr offen formuliert und gestaltet ist, lassen sich dennoch einige Parallelen dazu ziehen. Der zu behandelnde Lehrstoff umfasst nämlich unter anderem die folgenden drei Punkte, welche die SchülerInnen innerhalb der 9 Schulstufe erlernen sollen und dabei die komplette Thematik abdecken:

²⁰¹ s. *Bundesministerium für Bildung*: Verordnung über die Änderung der AHS Lehrpläne 2004, (Stand: 21.10.2016).

²⁰² s. *Digi.Komp*: Das Kompetenzmodell Informatik 5. Klasse 2013, (Stand: 21.10.2016).

²⁰³ s. *Digi.Komp*: Das Kompetenzmodell Wahlpflichtfach Informatik 2013, (Stand: 21.10.2016).

- Kalkulationsmodelle erstellen und die Ergebnisse bewerten und interpretieren sowie eine einfache Datenbank benutzen
- Einblicke in wesentliche Begriffe und Methoden der Informatik, ihre typischen Denk- und Arbeitsweisen, ihre historische Entwicklung sowie ihre technischen und theoretischen Grundlagen gewinnen und Grundprinzipien von Automaten, Algorithmen und Programmen kennenlernen
- wesentliche Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennenlernen sowie die Auswirkungen des Technikeinsatzes auf einen Einzelnen oder die Gesellschaft nachvollziehen²⁰⁴

Innerhalb des Lehrplans stellt Informatik das Ergebnis von Lernprozessen dar, in denen Grundlagen, Methoden, Anwendungen und Arbeitsweisen erschlossen und die gesellschaftliche Dimension von Informationstechnologien verdeutlicht werden. So sollen SchülerInnen informationstechnische Grundkenntnisse vermittelt bekommen, so dass sie diese zur Lösung einer Problemstellung sicher und kritisch einsetzen können. Dabei demonstriert die Analyse realer Prozesse wie z.B. von Salted-Hash-Authentifizierungsverfahren die Struktur komplexer Systeme und deren Wechselwirkungen, welche für die Vermittlung an SchülerInnen äußerst relevant sind. Außerdem ist es ein weiteres Bildungsziel, eine tiefere Einsicht in gesellschaftliche Zusammenhänge und Auswirkungen der Informationstechnologie z.B. in Bezug auf Sicherheit und Recht zu geben, wodurch sich das Thema Cookies erneut ideal für einen Unterrichtseinsatz eignet. Auch einige der vorgegebenen didaktischen Grundsätze werden beim Thema „Security im Webdatenbereich“ erfüllt. So werden beispielsweise vielseitige Bezüge aus der Lebens- und Begriffswelt der Jugendlichen aufgegriffen (Cookie Einsatz auch bei Facebook und Co), welche durch variierende Arbeitsformen wie Einzelarbeit, Gruppenarbeit oder kurzen Vorträgen sowie Demonstrationen, die Bedeutung gemeinschaftlichen Problemlösens hervorhebt. Durch die explorative, systematische und exemplarische Vorgehensweise wird auch die Sichtweise sowie die Orientierung erweitert, was zu einer Vertiefung von Wissen und Können führt. Somit wird ganz im Sinne des Bildungsbereichs „Natur und Technik“ die Erkenntnis für die Modellierung und Visualisierung eines Informatiksystems gewonnen, welches auch gegebenenfalls durch informatische Verfahren abstrahiert werden muss.²⁰⁵

4.2.1.2 Wahlpflichtfach Informatik (6. - 8. Klasse)

Ab der 10. Schulstufe wird Informatik zum Wahlpflichtgegenstand regelrecht degradiert. Prinzipiell dient das Wahlpflichtfach als Erweiterung bzw. Vertiefung vom Bildungsinhalt des Pflichtgegenstandes, welcher auch in einer höheren Schulstufe als jener, in der er erstmals angeboten wurde, erfolgen kann.²⁰⁶ Der geringe Wochenstundenanteil von zwei

²⁰⁴s. Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 5. Klasse 2004, (Stand: 24.10.2016).

²⁰⁵s. ebd.

²⁰⁶s. Jusline: § 11 (Abs. 3a) SchUG 1986 Pflichtgegenstände und verbindliche Übungen, (Stand: 24.10.2016).

Stunden wird dabei allerdings häufig kritisiert, da Informatik ein gesellschaftlich wie auch beruflich wichtiges Thema geworden ist. Allerdings bietet der zu behandelnde Lehrstoff erneut zahlreiche Parallelen, welche in Bezug auf diese Diplomarbeit von Relevanz sind:

- Datenbanken
- Konzepte von Programmiersprachen
- grundlegende Algorithmen und Datenstrukturen
- Informatik, Gesellschaft und Arbeitswelt
- Rechtsfragen²⁰⁷

Obwohl sich der Lehrplan für die 6. – 8. Klassen auf eine einzige DIN-A4 Seite beschränkt und stellenweise lediglich auf das Curriculum des Pflichtgegenstandes verweist, finden sich ein paar vertiefende Bildungs- sowie Lehrziele, welche in Bezug auf Cookies und Datensicherheit für eine bessere Verständlichkeit unter SchülerInnen sorgen. So wird erneut auf die Einsicht in die Systematik der Fachwissenschaft unter Berücksichtigung des soziokulturellen Umfelds eingegangen, wodurch zu konstruktiven Problemlösungen anhand von einfach umzusetzenden informatischen Mitteln geführt wird. Durch die Problemstellung des Salted-Hash-Verfahrens und der damit verbundenen Modularisierung stellt das Thema ein sehr komplexes Projekt dar, welches nebenbei zum abstrakten Denken anregt. Dadurch, dass die didaktischen Grundsätze spärlich ausgefallen sind, wird der Lehrplan des Pflichtgegenstandes weiterhin als Leitlinie des Wahlpflichtunterrichts propagiert. Es gibt aber noch einige weitere vertiefende didaktische Grundsätze, von welchen die in Frage kommenden nachfolgend aufgelistet werden:

- Durch Vergleich von Softwareprodukten sind grundlegende Funktionalitäten von Software zu vermitteln (SSL vs. Salted-Hash).
- Die Hinführung zu modul- und projektartigen Arbeiten bereitet sowohl eine Grundlage für das Fach selbst als auch für die fächerübergreifende Kooperation.
- Die Auswahl der Themen hat nach Möglichkeit Aktualitätsbezug zu haben und die Aufgabenstellungen haben nach Maßgabe des Unterrichts die Interessen der SchülerInnen zu berücksichtigen (Datensicherheit, „Social Networks“).
- Die im Lehrstoff vorgesehenen Inhalte sind so zu organisieren, dass sich jedes einzelne Thema in vernetzter Form im Unterricht wiederfindet und eine Vertiefung und Erweiterung ermöglicht (Bezug zum Pflichtgegenstand).²⁰⁸

²⁰⁷ s. Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 6.-8. Klasse 2004, (Stand: 24.10.2016).

²⁰⁸ s. *ebd.*

Teil II

Empirische Arbeit

Kapitel 5

Unterrichtsszenarien

In den folgenden Unterkapiteln werden nun die fünf ausgearbeiteten Unterrichtsszenarien, „Cookies im Webdatenbereich“, „Webinterfaces anhand aktueller Standards“, „Der Umstieg von HTML auf PHP“, „PHP Anmeldeskripts mit Base64 kodierten Cookies“ und „Authentifizierung anhand des Salted-Hash-Verfahrens“ einzeln beschrieben und näher erläutert, welche im Zusammenschluss das Unterrichtskonzept zu Security im Webdatenbereich mit dem Titel „Kekse ohne Salz schmecken nicht“ darstellen.

Von besonderer Bedeutung ist es, dass die Szenarien, obwohl sie in unterschiedlichen Schulstufen der Sekundarstufe II angewendet werden können und sollen, modular aufeinander aufbauend gestaltet sind. So zieht sich über das Pflichtfach Informatik (5. Klasse) ein einziger roter Faden, auf welchem weitestgehend im Wahlpflichtfach (6. - 8. Klasse) zurückgegriffen bzw. aufgebaut werden kann. Es ist auch möglich, lediglich einzelne Aspekte aus dem Unterrichtskonzept zu entnehmen und andere einfach auszulassen bzw. komplett neue Schwerpunkte zu definieren. Bezieht man sich dabei auf die Terminologie des „kompetenzorientierten Unterrichts“, werden Anforderungsbereiche wie z.B. Reproduktions-, Transfer-, Reflexions-, oder Problemlösungsleistungen innerhalb des Unterrichts vorausgesetzt, welche allesamt in den Planungen der Unterrichtsszenarien miteinbezogen wurden.²⁰⁹

Die einzelnen Unterrichtsszenarien decken neben einer Beschreibung sowie Ausarbeitung des Themas (Theorie, Lehrplanbezug und Kompetenzerwerb), eine detaillierte Planungsmatrix (Lehr- und Lernorganisation, -ziele und Inhalt) sowie etwaige Anwendungsbeispiele mit erklärenden Screenshots ab. Für die erleichterte Realisierung dieses Unterrichtskonzeptes sind außerdem alle Quelldateien, welche im Zuge der Szenarien benötigt werden, der Übersicht halber aufgelistet und anschließend im Anhang vollständig abgebildet.

²⁰⁹s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Prüfungsordnung AHS 2012, (Stand: 18.11.2016).

5.1 Pflichtfach Informatik (5. Klasse)

Für das Pflichtfach Informatik wurden zwei der insgesamt fünf Unterrichtsszenarien entwickelt, da sie zu Beginn einen ersten Einblick in die Materie bieten sollen, nur sehr geringe Programmierfertigkeiten vorausgesetzt werden und sich somit ideal für alle 5. Klassen eignen. Der Lehrplanbezug für die Szenarien „Cookies im Webdatenbereich“ und „Webinterfaces anhand aktueller Standards“ wird mit folgenden Verweisen daraus abgedeckt:

- Inhalte systematisieren und strukturieren sowie Arbeitsergebnisse zusammenstellen und multimedial präsentieren können
- Ein vernetztes Informationssystem für die individuelle Arbeit aufbauen und nutzen können
- Ein sicherer Umgang mit Standardsoftware zur schriftlichen Korrespondenz, zur Dokumentation, zur Publikation von Arbeiten, zur multimedialen Präsentation sowie zur Kommunikation erreichen
- Einblicke in wesentliche Begriffe und Methoden der Informatik, ihre typischen Denk- und Arbeitsweisen, ihre historische Entwicklung sowie ihre technischen und theoretischen Grundlagen gewinnen und Grundprinzipien von Automaten, Algorithmen und Programmen kennenlernen
- wesentliche Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennenlernen sowie die Auswirkungen des Technikeinsatzes auf einen Einzelnen oder die Gesellschaft nachvollziehen²¹⁰

5.1.1 Szenario 1: Cookies im Webdatenbereich

5.1.1.1 Themenbeschreibung & Ausarbeitung

Dem ersten Szenario des Unterrichtskonzeptes „Cookies im Webdatenbereich“ liegt eine Kombination aus theoretischen Inputs, praktischen Übungsaufgaben (Einzel- sowie Partnerarbeiten) und lebensnahen Verknüpfungen zugrunde, mit denen ein neuer Zugang zur Sicherheit und Anwendung von Cookies im Internet für SchülerInnen gelegt wird. So sollen die SchülerInnen zu Beginn in Form einer Partnerarbeit herausfinden, inwiefern Internetseiten Daten am Computer ablegen und wie man darüber informiert wird, bevor sie ihre Erkenntnisse gemeinsam mit ihren MitschülerInnen an der Tafel zusammentragen. Zur Aufklärung werden danach die wichtigsten Paragraphen des DSG, UrhG und TKG sowie weitere Begriffsdefinitionen geklärt, ehe man auf die sicherheitstechnischen Aspekte eingehen kann. Durch die Anregung zum Nachdenken über potentielle Cookie-Attacken können abschließend Cookies mit Hilfe eines Editors analysiert und editiert werden um aufzuzeigen, dass Manipulationen recht einfach vonstatten gehen.

²¹⁰s. Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 5. Klasse 2004, (Stand: 18.11.2016).

Als theoretische Grundlage bzw. Voraussetzung kann für das Unterrichtsszenario 1 „Cookies im Webdatenbereich“ das Kapitel 2.3.4 „Cookies“, aus dem Teil der theoretischen Grundlagen dieser Diplomarbeit herangezogen werden. Im Weiteren sind die Unterkapitel 2.3.4.1 „Rechtliche Aspekte“, 2.3.4.1.1 „Security“ sowie 2.3.4.2 „Probleme mit Cookies“ von besonderer Bedeutung, da hier besonders auf die aktuelle Gesetzeslage sowie auf die gängigen Angriffsmethoden von Cookies eingegangen wird.

Um über die Kompetenzorientierung dieses Unterrichtsszenarios Aufschluss geben zu können, wurden die zu behandelnden Themen und Inhalte mit den digitalen Grundkompetenzen für das Pflichtfach Informatik vom Digi.Komp des BMB verglichen und analysiert, wodurch sich die folgenden Schnittmengen ergaben:

- Bedeutung von Informatik in der Gesellschaft:
 - Die SchülerInnen können Bereiche für den Einsatz von Informatiksystemen und ihre gesellschaftlichen und wirtschaftlichen Auswirkungen beschreiben.
 - Die SchülerInnen können den Einfluss von Informatiksystemen auf ihren Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen.
- Verantwortung, Datenschutz und Datensicherheit:
 - Die SchülerInnen können ihre Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären.
 - Die SchülerInnen können über ihre Verantwortung beim Einsatz von Informatiksystemen reflektieren.
 - Die SchülerInnen können verschiedene Schutzmaßnahmen für Daten und IT-Systeme reflektieren und bewerten.
- Technische Grundlagen und Funktionsweisen:
 - Die SchülerInnen können Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären.
 - Die SchülerInnen verstehen grundlegende technische Konzepte von Informatiksystemen.
- Netzwerke:
 - Die SchülerInnen können verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten beschreiben und erklären.
 - Die SchülerInnen können verschiedene Internetdienste nutzen und konfigurieren.
 - Die SchülerInnen können technische Aspekte von Netzwerken hinsichtlich der Qualität und Sicherheit einschätzen.

- Die SchülerInnen können die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.
- Suche, Auswahl und Organisation von Information:
 - Die SchülerInnen können wichtige Informationsquellen im Internet anführen, die für ihre schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind.
 - Die SchülerInnen können über die Relevanz und Qualität von Informationen reflektieren.
- Konzepte der Informationsverarbeitung:
 - Die SchülerInnen können informatische Konzepte der Informatik benennen und an Hand von Beispielen erklären.
 - Die SchülerInnen können Konzepte der Informatik bei der Lösung konkreter Aufgaben anwenden.²¹¹

5.1.1.2 Planungsmatrix

In der folgenden Planungsmatrix wird der geplante Doppelstundenverlauf (2x 50min) des Szenarios 1 anhand der Themen und Inhalte, der Lehr- und Lernziele sowie der Lehr- und Lernorganisation aufgezeigt, welche sich schlussendlich in die zu verwendeten Methoden, Medien und Sozialformen aufgliedert.

Zeit:	Lehr- / Lernziele:	Thema / Inhalt:	Lehr- / Lernorganisation:		
			Methoden:	Medien:	Sozialform:
1. Std. 05 min	Herstellung von Ruhe und Aufmerksamkeit	Begrüßung, Organisatorisches	Vortrag		LSG
1. Std. 10 min	Hinführung zum Thema / Sensibilisierung	Datenspeicherung und -sicherheit im Internet	Vortrag	Computer, Internet, Projektor	LV
1. Std. 20 min	Aktivierung des Vorwissens / Anregung zum Nachdenken	Welche Seiten speichern was? Wie wird man darüber informiert?	Partnerarbeit	Notizen	SSG
1. Std. 15 min	Überprüfung der Partnerarbeit / Vertiefung, Verständnisaufbau und Feedback	Welche Seiten speichern was? Wie wird man darüber informiert?	Reflexion	Notizen, Tafel, Computer, Internet, Projektor	LSG
2. Std. 20 min	Wissensvermittlung / Anregung zum Nachdenken	DSG, UrhG, TKG, personenbezogene und indirekt personenbezogene Daten, Opt-In / Opt-Out	Gesetzes-texte lesen	Computer, Internet, Projektor, RIS	LSG
2. Std. 10 min	Überprüfung des Gelesenen / Vertiefung, Verständnisaufbau und Feedback	Probleme mit Cookies (Sicherheit)	Diskussion		LSG

²¹¹s. *Digi.Komp*: Das Kompetenzmodell Informatik 5. Klasse 2013, (Stand: 21.11.2016).

2. Std. 10 min	Wissensvermittlung / Anregung zum Nachdenken	Cookie-Steal- bzw. Cookie- Replay-Attack	Vortrag	Computer, Projektor, Grafik der Angriffe	LV
2. Std. 10 min	Festigung des Wissens / Aufgabenbewältigung durch erworbenes Wissen	Editierung eines Cookies	Vortrag, Einzelarbeit	Computer, Internet, Projektor, Cookieeditor	LV, LSG

Tabelle 5.1.1: Planungsmatrix Szenario 1: Cookies im Webdatenbereich

In Bezug auf die Rückkopplung und (Selbst-)Reflexion können für das Unterrichtsszenario 1 „Cookies im Webdatenbereich“ die folgenden Faktoren zur Überprüfung herangezogen werden:

- Reaktionen der SchülerInnen
- Mitarbeit der SchülerInnen sowie zahlreich aufkommende Fragen
- Ergebnisse der Notizen und des Tafelbildes
- Partizipation am Meinungsaustausch
- Selbstreflexion der SchülerInnen

5.1.1.3 Anwendungsbeispiele

Für das erste Unterrichtsszenario benötigt man, neben der vorhin bereits erwähnten Theorie sowie den notwendigen Internetquellen einen Cookieeditor, um mit den SchülerInnen eine Speicheranalyse inkl. anschließender Bearbeitung durchführen zu können. Dadurch, dass die Wahl des entsprechenden Cookieeditors vom Browser abhängig ist, kann für Google Chrome „EditThisCookie“ (Abb. 5.1.1) bzw. für Apple Safari „Cookie“ als kostenlose Anwendungssoftware empfohlen werden, welche für die Installation jeweils Administratorrechte voraussetzen.

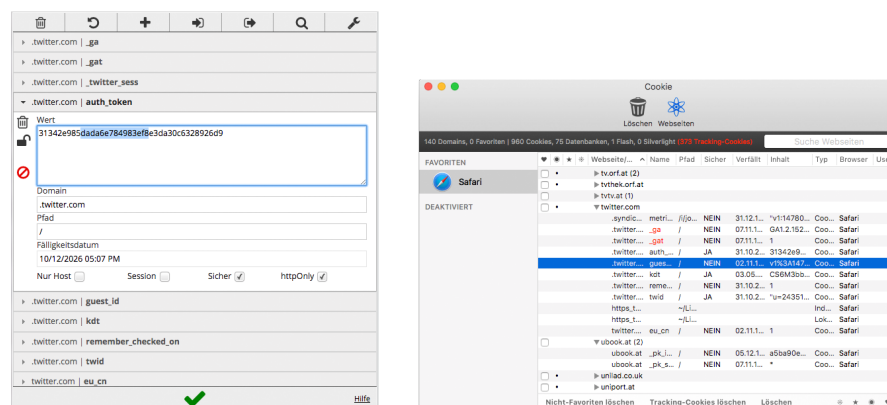


Abbildung 5.1.1: Diverse Cookieeditoren (Chrome & Safari) für Speicheranalyse und Bearbeitung

Weitere Unterrichtsmaterialien stellen beispielsweise Abbildung 2.3.9 „Vorgehensweise von Angreifern bei einer Cookie-Steal- bzw. Cookie-Replay-Attack“ dar.

5.1.2 Szenario 2: Webinterfaces anhand aktueller Standards

5.1.2.1 Themenbeschreibung & Ausarbeitung

Dem zweiten Szenario des Unterrichtskonzeptes „Webinterfaces anhand aktueller Webstandards“ liegt eine Kombination aus theoretischen Inputs, praktischen Übungsaufgaben (nur Einzelarbeiten) und berufstechnischen Bezügen zugrunde, mit denen die standardisierte Entwicklung moderner Webinterfaces für SchülerInnen zugänglich gemacht wird. So sollen die SchülerInnen zu Beginn durch das Vorzeigen eines Musterbeispiels dazu angeregt werden, mittels aktueller Webentwicklungstechniken eine eigene kleine Webseite zu bauen, welche dabei vollkommen responsiv und optisch ansprechend gestaltet ist. In der Folge werden die wichtigsten Meta-Elemente und Tags geklärt, ehe man auf die Einbettung dieser, innerhalb eines Grundgerüsts, eingehen kann. Durch die Anpassung und Individualisierung der Webseite sowie durch das Einfügen von personalisierten Inhalten, können die SchülerInnen zum Abschluss dieses Szenarios noch einmal uneingeschränkt ihrer Kreativität nachgehen.

Als theoretische Grundlage bzw. Voraussetzung kann für das Unterrichtsszenario 2 „Webinterfaces anhand aktueller Webstandards“ das Kapitel 2 „Standards in der Webentwicklung“, aus dem Teil der theoretischen Grundlagen dieser Diplomarbeit herangezogen werden. Im Weiteren sind die Unterkapitel 2.1 „HTML5“ sowie 2.2 „CSS3“ von besonderer Bedeutung, da hier besonders auf die neuen Elemente von HTML5 sowie auf die grundlegenden Anwendungsmethoden des responsiven Webdesigns eingegangen wird.

Um über die Kompetenzorientierung dieses Unterrichtsszenarios Aufschluss geben zu können wurden die zu behandelnden Themen und Inhalte mit den digitalen Grundkompetenzen für das Pflichtfach Informatik vom Digi.Komp des BMB verglichen und analysiert, wodurch sich die folgenden Schnittmengen ergaben:

- Bedeutung von Informatik in der Gesellschaft:
 - Die SchülerInnen können Bereiche für den Einsatz von Informatiksystemen und ihre gesellschaftlichen und wirtschaftlichen Auswirkungen beschreiben.
 - Die SchülerInnen können Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen.
- Verantwortung, Datenschutz und Datensicherheit:
 - Die SchülerInnen können beim Einsatz von Informatiksystemen ihr Wissen um Pflichten und Rechte in Bezug auf ihre Person und ihre Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden.

- Geschichte der Informatik:
 - Die SchülerInnen können Meilensteine in der Entwicklung der Informatik beschreiben und maßgebliche dahinterstehende Persönlichkeiten nennen.
 - Die SchülerInnen können ihr Wissen über die Geschichte der Informatik in Beziehung zu aktuellen Entwicklungen setzen.
- Berufliche Perspektiven:
 - Die SchülerInnen können Berufsfelder benennen, in denen die Anwendung der IT eine bedeutende Rolle spielt.
- Technische Grundlagen und Funktionsweisen:
 - Die SchülerInnen können Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären.
 - Die SchülerInnen verstehen grundlegende technische Konzepte von Informatiksystemen.
- Netzwerke:
 - Die SchülerInnen können verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten beschreiben und erklären.
 - Die SchülerInnen können verschiedene Internetdienste nutzen und konfigurieren.
 - Die SchülerInnen können die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.
- Mensch-Maschine-Schnittstelle:
 - Die SchülerInnen können verschiedene Arten der MMS beschreiben und die Fragen der Barrierefreiheit für Menschen mit besonderen Bedürfnissen erklären.
- Produktion digitaler Medien:
 - Die SchülerInnen können gängige Medienformate und ihre Eigenschaften beschreiben.
 - Die SchülerInnen können digitale Produkte in Bezug auf inhaltliche Relevanz und Design reflektieren.
- Konzepte der Informationsverarbeitung:
 - Die SchülerInnen können informatische Konzepte der Informatik benennen und an Hand von Beispielen erklären.
 - Die SchülerInnen können Konzepte der Informatik bei der Lösung konkreter Aufgaben anwenden.

- Algorithmen, Datenstrukturen und Programmierung:
 - Die SchülerInnen können einfache Aufgaben mit Mitteln der Informatik modellieren.
 - Die SchülerInnen können an Hand von einfachen Beispielen die Korrektheit von Programmen bewerten.²¹²

5.1.2.2 Planungsmatrix

In der folgenden Planungsmatrix wird der geplante Doppelstundenverlauf (2x 50min) des Szenarios 2 anhand der Themen und Inhalte, der Lehr- und Lernziele sowie der Lehr- und Lernorganisation aufgezeigt, welche sich schlussendlich in die zu verwendeten Methoden, Medien und Sozialformen aufgliedert.

Zeit:	Lehr- / Lernziele:	Thema / Inhalt:	Lehr- / Lernorganisation:		
			Methoden:	Medien:	Sozialform:
1. Std. 05 min	Herstellung von Ruhe und Aufmerksamkeit	Begrüßung, Organisatorisches	Vortrag		LSG
1. Std. 20 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	Moderne Webinterfaces mit HTML5 und CSS3	Vortrag, Reflexion	Computer, Internet, Projektor	LSG
1. Std. 05 min	Aktivierung des Vorwissens / Interesse wecken und Vertiefung	Musterbeispiel vorzeigen	Vortrag	Computer, Internet, Projektor, Musterbeispiel	LV
1. Std. 20 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Inbetriebnahme der Entwicklungsumgebung, Erstellung eines Grundgerüsts (index.html, layout.css)	Vortrag, Einzelarbeit	Computer, NetBeans, Projektor	LV, LSG
2. Std. 15 min	Wissensvermittlung / Vertiefung und Verständnisaufbau	Metadaten, HTML-Syntax und CSS-Syntax	Vortrag, Einzelarbeit	Computer, NetBeans, Projektor	LV, LSG
2. Std. 15 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Festlegung von header, nav, main und footer, Einbau eines fluiden Grid-Frameworks	Vortrag, Einzelarbeit	Computer, NetBeans, Projektor	LV, LSG
2. Std. 10 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Anpassung der Elemente mit CSS	Vortrag, Einzelarbeit	Computer, NetBeans, Projektor	LV, LSG
2. Std. 10 min	Festigung des Wissens / Aufgabenbewältigung durch erworbenes Wissen	Befüllen der Webseite mit Inhalten und weitere Individualisierung	Einzelarbeit	Computer, NetBeans	LSG

Tabelle 5.1.2: Planungsmatrix Szenario 2: Webinterfaces anhand aktueller Standards

²¹²s. Digi.Komp: Das Kompetenzmodell Informatik 5. Klasse 2013, (Stand: 29.11.2016).

In Bezug auf die Rückkopplung und (Selbst-)Reflexion können für das Unterrichtsszenario 2 „Webinterfaces anhand aktueller Webstandards“ die folgenden Faktoren zur Überprüfung herangezogen werden:

- Reaktionen der SchülerInnen
- Mitarbeit der SchülerInnen sowie zahlreich aufkommende Fragen
- Begeisterung durch kreative Inputs und Individualisierung
- Selbstreflexion der SchülerInnen

5.1.2.3 Anwendungsbeispiele

Für das zweite Unterrichtsszenario benötigt man, neben der vorhin bereits erwähnten Theorie sowie den notwendigen Internetquellen, zumindest einen Screenshot vom gewünschten Endresultat der Webseite (*Abb. 5.1.2*), um den SchülerInnen den Funktionsumfang sowie das Layout näher bringen zu können. Dadurch, dass die Wahl eines entsprechenden Browsers sowie einer geeigneten Entwicklungsumgebung oftmals als schwierig erscheint, kann in diesem Fall Google „Chrome“ sowie Oracle „NetBeans“ als kostenlose Anwendungssoftware empfohlen werden, welche für die Installation jeweils Administratorrechte voraussetzen.

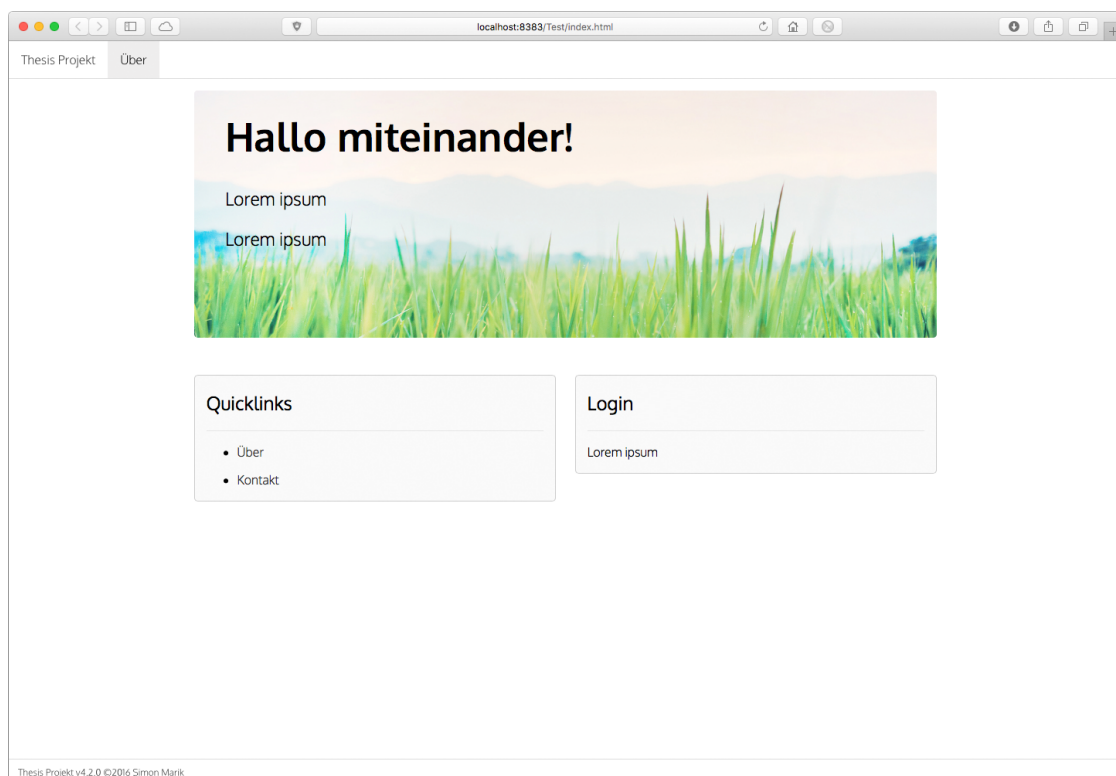


Abbildung 5.1.2: Ausgeführtes Musterbeispiel für Szenario 2: Webinterfaces anhand aktueller Standards

Weitere Unterrichtsmaterialien stellen beispielsweise Anhang A „Index HTML Seitenaufbau (index.html)“ sowie Anhang B „Layout Stylesheet (layout.css)“ dar, welche sich am Ende dieser Arbeit befinden.

5.2 Wahlpflichtfach Informatik (6. - 8. Klasse)

Für das Wahlpflichtfach Informatik wurden drei der insgesamt fünf Unterrichtsszenarien entwickelt, da sie eine Vertiefung in die Thematik darstellen sollen, erfahrenere Programmierfertigkeiten vorausgesetzt werden und sich somit ideal für alle 6. - 8. Klassen eignen. Der Lehrplanbezug für die Szenarien „Der Umstieg von HTML auf PHP“, „PHP Anmeldeskripts mit Base64 kodierten Cookies“ und „Authentifizierung anhand des Salted-Hash-Verfahrens“ wird mit folgenden Verweisen daraus abgedeckt:

- Datenbanken
- Konzepte von Programmiersprachen
- grundlegende Algorithmen und Datenstrukturen
- Informatik, Gesellschaft und Arbeitswelt
- Rechtsfragen²¹³

5.2.1 Szenario 3: Der Umstieg von HTML auf PHP

5.2.1.1 Themenbeschreibung & Ausarbeitung

Dem dritten Szenario des Unterrichtskonzeptes „Der Umstieg von HTML auf PHP“ liegt eine Kombination aus theoretischen Inputs, praktischen Übungsaufgaben (nur Einzelarbeiten) und berufstechnischen Bezügen zugrunde, mit denen der Übergang von HTML zu PHP für SchülerInnen fließend gestaltet wird. Zu Beginn erhalten die SchülerInnen durch das Vorzeigen des Musterbeispiels und den dazugehörigen Vorträgen die notwendigen Informationen um ihre eigene Webseite des letzten Szenarios auf PHP umzurüsten, wodurch der Konnex zu Szenario 2 gelegt wird. Zur Aufklärung werden PHP-Elemente wie Sessions, Funktionen oder Cookies besprochen, ehe man auf die Thematik rund um virtuelle Webserver wie z.B. „XAMPP“ oder „MAMP“ zu sprechen kommen kann. Durch die Realisierung eines ersten echo-Befehls, der Auslagerung von Seitenelementen sowie der Erstellung eines Anmeldeformulars einschließlich geschützter Login-Seite, können die SchülerInnen einen ersten oberflächlichen Eindruck von PHP gewinnen, ohne dass dabei jemand überfordert wird.

Als theoretische Grundlage bzw. Voraussetzung kann für das Unterrichtsszenario 3 „Der Umstieg von HTML auf PHP“ das Kapitel 2.3 „PHP“, aus dem Teil der theoretischen

²¹³s. Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 6.-8. Klasse 2004, (Stand: 18.11.2016).

Grundlagen dieser Diplomarbeit herangezogen werden. Im Weiteren sind die Unterkapitel 2.3.2 „Externer Dateibezug“, 2.3.3 „PHP Sessions“ sowie 2.3.4 „Cookies“ von besonderer Bedeutung, da hier explizit auf die Verwendung von PHP sowie auf die unterschiedlichen Anwendungen von Sessions und Cookies eingegangen wird.

Um über die Kompetenzorientierung dieses Unterrichtsszenarios Aufschluss geben zu können, wurden die zu behandelnden Themen und Inhalte mit den digitalen Grundkompetenzen für das Wahlpflichtfach Informatik vom Digi.Komp des BMB verglichen und analysiert, wodurch sich die folgenden Schnittmengen ergaben:

- Bedeutung von Informatik in der Gesellschaft:
 - Die SchülerInnen können Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen.
- Geschichte der Informatik:
 - Die SchülerInnen können Meilensteine in der Entwicklung der Informatik beschreiben und maßgebliche dahinterstehende Persönlichkeiten nennen.
- Berufliche Perspektiven:
 - Die SchülerInnen können ihr Wissen und ihre schulischen Erfahrungen im Zusammenhang mit IT für ihre künftige Erwerbsbiographie nutzen.
- Technische Grundlagen und Funktionsweisen:
 - Die SchülerInnen können Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären.
 - Die SchülerInnen können grundlegende technische Konzepte von Informatiksystemen erklären.
- Netzwerke:
 - Die SchülerInnen können verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten und Funktionsweisen beschreiben und erklären.
 - Die SchülerInnen können verschiedene Internetdienste nutzen.
 - Die SchülerInnen können die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.
- Produktion digitaler Medien:
 - Die SchülerInnen können digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.
- Kalkulationsmodelle und Visualisierung:
 - Die SchülerInnen können den (informatischen) Funktionsbegriff erklären.

- Konzepte der Informationsverarbeitung:
 - Die SchülerInnen können wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären.
- Algorithmen, Datenstrukturen und Programmierung:
 - Die SchülerInnen können Aufgaben mit Mitteln der Informatik modellieren.
 - Die SchülerInnen können die Schritte der Softwareentwicklung reflektieren.
 - Die SchülerInnen können die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
 - Die SchülerInnen können gezielt nach Programmfehlern suchen und diese korrigieren.²¹⁴

5.2.1.2 Planungsmatrix

In der folgenden Planungsmatrix wird der geplante Doppelstundenverlauf (2x 50min) des Szenarios 3 anhand der Themen und Inhalte, der Lehr- und Lernziele sowie der Lehr- und Lernorganisation aufgezeigt, welche sich schlussendlich in die zu verwendeten Methoden, Medien und Sozialformen aufgliedert.

Zeit:	Lehr- / Lernziele:	Thema / Inhalt:	Lehr- / Lernorganisation:		
			Methoden:	Medien:	Sozialform:
1. Std. 05 min	Herstellung von Ruhe und Aufmerksamkeit	Begrüßung, Organisatorisches	Vortrag		LSG
1. Std. 20 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	Sichere Webinterfaces mit PHP (Sessions, Cookies, Funktionen)	Vortrag	Computer, Internet, Projektor	LV
1. Std. 10 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	Virtuelle Webserver (XAMPP, MAMP)	Vortrag	Computer, Webserver, Projektor	LV
1. Std. 05 min	Aktivierung des Vorwissens / Interesse wecken und Vertiefung	Musterbeispiel vorzeigen / Anknüpfung an Szenario 2	Vortrag, Reflexion	Computer, NetBeans, Webserver, Projektor, Musterbeispiel	LV, LSG
1. Std. 10 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Inbetriebnahme des Webserver, Suffix Änderung von *.html auf *.php (index.html = index.php)	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 05 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	PHP-Syntax und erstes „echo“	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG

²¹⁴s. Digi.Komp: Das Kompetenzmodell Wahlpflichtfach Informatik 2013, (Stand: 29.11.2016).

2. Std. 25 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Auslagerung des Headers und Footers in externe Dateien (header.php, footer.php) inkl. Einbindung im Index	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 20 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Einbettung eines HTML Anmeldeformulars und Erstellung des geschützten Logins (dashboard.php)	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG

Tabelle 5.2.1: Planungsmatrix Szenario 3: Der Umstieg von HTML auf PHP

In Bezug auf die Rückkopplung und (Selbst-)Reflexion können für das Unterrichtsszenario 3 „Der Umstieg von HTML auf PHP“ die folgenden Faktoren zur Überprüfung herangezogen werden:

- Reaktionen der SchülerInnen
- Mitarbeit der SchülerInnen sowie zahlreich aufkommende Fragen
- Selbstreflexion der SchülerInnen

5.2.1.3 Anwendungsbeispiele

Für das dritte Unterrichtsszenario benötigt man, neben der vorhin bereits erwähnten Theorie sowie den notwendigen Internetquellen, zumindest einen Screenshot vom gewünschten Endresultat der PHP-Seite (*Abb. 5.2.1*), um den SchülerInnen den Funktionsumfang sowie die einzelnen Dateiteile erklären zu können. Dadurch, dass die Wahl eines entsprechenden virtuellen Webserver sowie einer geeigneten Entwicklungsumgebung oftmals als schwierig erscheint, kann in diesem Fall „XAMPP“ (Cross-Plattform) oder „MAMP“ (macOS) sowie Oracle „NetBeans“ als kostenlose Anwendungssoftware empfohlen werden, welche für die Installation jeweils Administratorrechte voraussetzen.

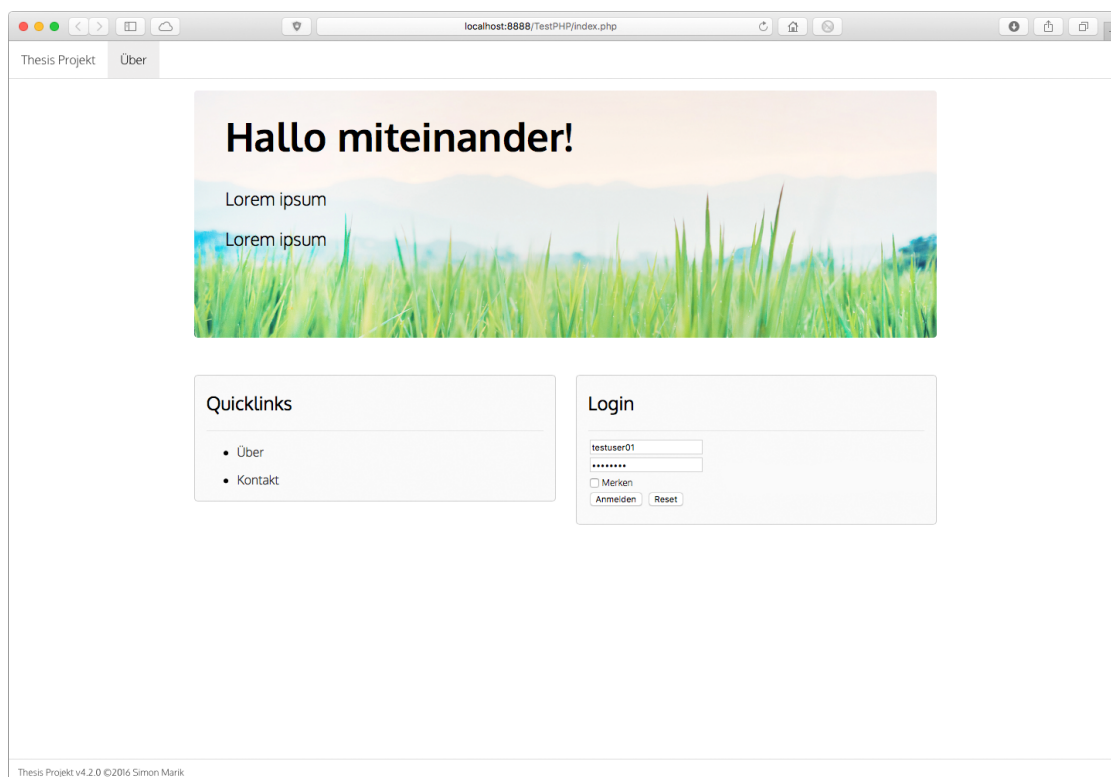


Abbildung 5.2.1: Ausgeführtes Musterbeispiel für Szenario 3: Der Umstieg von HTML auf PHP

Weitere Unterrichtsmaterialien stellen beispielsweise Anhang C „Index PHP Seitenaufbau (index.php)“ sowie Anhang D „Login Seitenaufbau (dashbd.php)“ dar, welche sich am Ende dieser Arbeit befinden.

5.2.2 Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies

5.2.2.1 Themenbeschreibung & Ausarbeitung

Dem vierten Szenario des Unterrichtskonzeptes „PHP Anmeldeskripts mit Base64 kodierten Cookies“ liegt eine Kombination aus theoretischen Inputs und Wiederholungen, praktischen Übungsaufgaben (nur Einzelarbeiten) und praxisnahen Bezügen zugrunde, mit denen die Funktionsweise von PHP-Skripte und deren Anwendung für SchülerInnen einfach zugänglich gemacht wird. Zu Beginn erhalten die SchülerInnen durch das Vorzeigen der Benutzerdatenbank und den dazugehörigen Vorträgen die notwendigen Informationen, wie eine Webseite mit Datenbanken überhaupt kommunizieren kann. Dabei soll auch der Konnex zur Zeichenkodierung und Ähnlichem gelegt werden, ehe man sich dem eigentlichen Verbindungsskript (scr.connect.php) widmen kann. Zur Vertiefung werden diverse Verschlüsselungsmethoden wie z.B. Base64 oder SHA1 vorgestellt, um diese gemeinsam im Anschluss innerhalb eines Anmeldeskripts (scr.signin.php) verbauen zu können. Durch die Überprüfung der Base64-Hashwerte innerhalb von Lookup- bzw. Rainbow-Tables wird den SchülerInnen dabei aufgezeigt, dass gängige

Verschlüsselungsalgorithmen über die Jahre weitgehend kompromittiert wurden und somit keinen effizienten Schutz mehr bieten.

Als theoretische Grundlage bzw. Voraussetzung kann für das Unterrichtsszenario 4 „PHP Anmeldeskripts mit Base64 kodierten Cookies“ das Kapitel 2.3.4 „Cookies“, aus dem Teil der theoretischen Grundlagen dieser Diplomarbeit herangezogen werden. Im Weiteren sind die Unterkapitel 2.3.4.3 „Verschlüsselung von Cookies“ sowie 2.3.4.4 „Salted-Hash-Cookies“ von besonderer Bedeutung, da hier die Grundlagen über gängige Cookie-Verschlüsselungsmethoden abgedeckt und diverse Abgrenzungen zu gesalzenen Anmeldesystemen aufgezeigt werden.

Um über die Kompetenzorientierung dieses Unterrichtsszenarios Aufschluss geben zu können, wurden die zu behandelnden Themen und Inhalte mit den digitalen Grundkompetenzen für das Wahlpflichtfach Informatik vom Digi.Komp des BMB verglichen und analysiert, wodurch sich die folgenden Schnittmengen ergaben:

- Bedeutung von Informatik in der Gesellschaft:
 - Die SchülerInnen können Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen.
 - Die SchülerInnen können den Einfluss von Informatiksystemen auf ihren Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen.
- Verantwortung, Datenschutz und Datensicherheit:
 - Die SchülerInnen können ihre Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären.
 - Die SchülerInnen können für den Schutz und die Sicherheit von Informatiksystemen, mit denen sie arbeiten, sorgen.
 - Die SchülerInnen können verschiedene Schutzmaßnahmen für Daten und IT-Systeme beurteilen und empfehlen.
 - Die SchülerInnen können die Rechtskonformität einer Website in Grundzügen bewerten.
- Technische Grundlagen und Funktionsweisen:
 - Die SchülerInnen können Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären.
 - Die SchülerInnen können grundlegende technische Konzepte von Informatiksystemen erklären.

- Netzwerke:
 - Die SchülerInnen können verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten und Funktionsweisen beschreiben und erklären.
 - Die SchülerInnen können verschiedene Internetdienste nutzen.
 - Die SchülerInnen können die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.
- Kalkulationsmodelle und Visualisierung:
 - Die SchülerInnen können den (informatischen) Funktionsbegriff erklären.
- Konzepte der Informationsverarbeitung:
 - Die SchülerInnen können wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären.
 - Die SchülerInnen können bei der Lösung konkreter Aufgaben Heuristiken, Grundprinzipien und Konzepte der Informatik anwenden und informatische Modelle gestalten.
 - Die SchülerInnen können unterschiedliche Lösungsansätze in Bezug auf zugrundeliegende Konzepte reflektieren und in konkreten Handlungssituationen bewerten.
- Algorithmen, Datenstrukturen und Programmierung:
 - Die SchülerInnen können den Algorithmusbegriff erklären.
 - Die SchülerInnen können Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben.
 - Die SchülerInnen können wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern.
 - Die SchülerInnen können Aufgaben mit Mitteln der Informatik modellieren.
 - Die SchülerInnen können Algorithmen entwerfen, diese formal darstellen, implementieren und testen.
 - Die SchülerInnen können die Schritte der Softwareentwicklung reflektieren.
 - Die SchülerInnen können die Effizienz von Algorithmen bewerten.
 - Die SchülerInnen können gezielt nach Programmfehlern suchen und diese korrigieren.
- Datenmodelle und Datenbanksysteme:
 - Die SchülerInnen können den Begriff Datenbanken und wichtige Fachbegriffe beschreiben und an Beispielen erklären.
 - Die SchülerInnen können Daten strukturiert (in Tabellen) erfassen, abfragen, auswerten sowie Datenbanken modellieren und einfache automatisierte Datenbanklösungen entwickeln.²¹⁵

²¹⁵s. *Digi.Komp*: Das Kompetenzmodell Wahlpflichtfach Informatik 2013, (Stand: 29.11.2016).

5.2.2.2 Planungsmatrix

In der folgenden Planungsmatrix wird der geplante Doppelstundenverlauf (2x 50min) des Szenarios 4 anhand der Themen und Inhalte, der Lehr- und Lernziele sowie der Lehr- und Lernorganisation aufgezeigt, welche sich schlussendlich in die zu verwendeten Methoden, Medien und Sozialformen aufgliedert.

Zeit:	Lehr- / Lernziele:	Thema / Inhalt:	Lehr- / Lernorganisation:		
			Methoden:	Medien:	Sozialform:
1. Std. 05 min	Herstellung von Ruhe und Aufmerksamkeit	Begrüßung, Organisatorisches	Vortrag		LSG
1. Std. 10 min	Aktivierung des Vorwissens / Interesse wecken und Vertiefung	Wiederholung PHP Sessions, Cookies, Funktionen, Dateieinbindung	Reflexion		LSG
1. Std. 05 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	PHP-Skripte (Auslagerung, Einbindung)	Vortrag	Computer, Webserver, Projektor	LV
1. Std. 10 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Erstellung eines Verbindungsskripts zu existierender Datenbank (scr.connect.php)	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
1. Std. 20 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Erstellung eines Anmeldeskripts mit Sessionvariablen und Cookies (scr.signin.php)	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 20 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	Verschlüsselung von Cookies (Base64, SHA1)	Vortrag	Computer, NetBeans, Webserver, Projektor	LV
2. Std. 15 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Implementierung der Base64 Kodierung im Anmeldeskript	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 15 min	Wissensvermittlung / Anregung zum Nachdenken	Lookup- bzw. Rainbow-Tables für Base64 Hashwerte suchen	Vortrag, Einzelarbeit	Computer, Internet, Projektor	LSG

Tabelle 5.2.2: Planungsmatrix Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies

In Bezug auf die Rückkopplung und (Selbst-)Reflexion können für das Unterrichtsszenario 4 „PHP Anmeldeskripts mit Base64 kodierten Cookies“ die folgenden Faktoren zur Überprüfung herangezogen werden:

- Ergebnis der Wiederholung
- Reaktionen der SchülerInnen
- Mitarbeit der SchülerInnen sowie zahlreich aufkommende Fragen
- Selbstreflexion der SchülerInnen

5.2.2.3 Anwendungsbeispiele

Für das vierte Unterrichtsszenario benötigt man, neben der vorhin bereits erwähnten Theorie sowie den notwendigen Internetquellen, die Codeauszüge der zu erstellenden PHP-Skripte (scr.connect.php & scr.signin.php), um den SchülerInnen den Funktionsumfang sowie die einzelnen Programmierschritte vorab erklären zu können. Durch die abschließende Eingabe diverser Base64-Hashwerte bei „base64decode.org“ (Abb. 5.2.2) oder ähnlichen Decodern wird aufgezeigt, dass sich die entsprechenden dekodierten Werte zur Repräsentation ohne Probleme finden lassen.

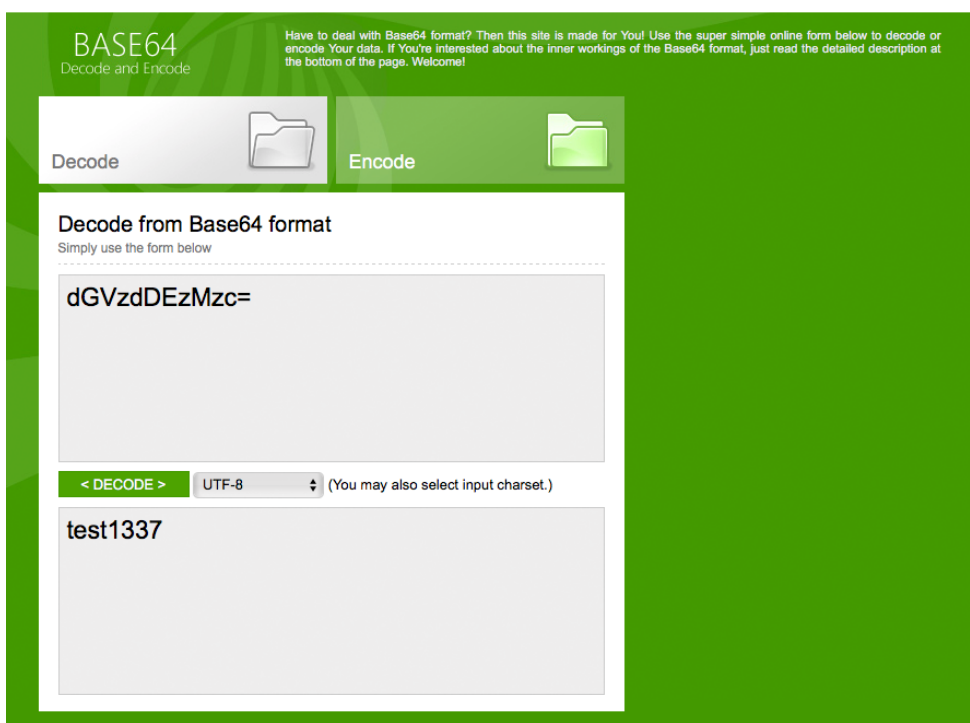


Abbildung 5.2.2: Lookup-Table Ergebnis für Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies

Weitere Unterrichtsmaterialien stellen beispielsweise Anhang E „Users Datenbank (thesis.sql)“, Anhang F „Connection Skript (scr.connect.php)“ sowie Anhang G „Sign in Skript (scr.signin.php)“ dar, welche sich am Ende dieser Arbeit befinden.

5.2.3 Szenario 5: Authentifizierung anhand des Salted-Hash-Verfahrens

5.2.3.1 Themenbeschreibung & Ausarbeitung

Dem fünften Szenario des Unterrichtskonzeptes „Authentifizierung anhand des Salted-Hash-Verfahrens“ liegt eine Kombination aus theoretischen Inputs und Wiederholungen, praktischen Übungsaufgaben (nur Einzelarbeiten) und praxisnahen Bezügen zugrunde,

mit denen die Funktionsweise von gesalzenen Authentifizierungsverfahren während eines Logins SchülerInnen näher gebracht wird. Zu Beginn sollen die SchülerInnen durch das Vorzeigen einer nachgestellten Cookie-Replay-Attacke sowie den dazugehörigen Vorträgen für korrekte Anmeldeauthentifizierungen sensibilisiert werden. Dabei sollen auch erneut Sessions, Cookies sowie Skripte wiederholt werden, bevor man sich dem eigentlichen Authentifizierungsskript (scr.auth.php) widmen kann. Zur Vertiefung werden die Verschlüsselungsmethoden des Blowfish-Algorithmus und des Salted-Hash-Verfahrens vorgestellt, um diese im Anschluss innerhalb des Authentifizierungs- und des Anmelde-skripts (scr.signin.php) verbauen zu können. Durch die Überprüfung der Salted-Hashwerte innerhalb von Lookup-Tables wird den SchülerInnen aufgezeigt, dass diese Form der Absicherung einen viel größeren Schutz bietet, da hier die Hashwerte nicht aufgefunden werden können.

Als theoretische Grundlage bzw. Voraussetzung kann für das Unterrichtsszenario 5 „Authentifizierung anhand des Salted-Hash-Verfahrens“ das Kapitel 2.3.4.4 „Salted-Hash-Cookies“, aus dem Teil der theoretischen Grundlagen dieser Diplomarbeit, herangezogen werden. Im Weiteren sind auch die Kapitel 2.3.4.3.2 „SHA1 En-/Decryption“ sowie 2.3.4.3.3 „BCrypt Blowfish Algorithmus“ von besonderer Bedeutung, da hier nochmals explizit auf die benötigten Verschlüsselungsmethoden wie BCrypt oder SHA1 sowie auf die korrekte Anwendung des Salted-Hash-Verfahrens eingegangen wird.

Um über die Kompetenzorientierung dieses Unterrichtsszenarios Aufschluss geben zu können, wurden die zu behandelnden Themen und Inhalte mit den digitalen Grundkompetenzen für das Wahlpflichtfach Informatik vom Digi.Komp des BMB verglichen und analysiert, wodurch sich die folgenden Schnittmengen ergaben:

- Bedeutung von Informatik in der Gesellschaft:
 - Die SchülerInnen können Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen.
 - Die SchülerInnen können den Einfluss von Informatiksystemen auf ihren Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen.
- Verantwortung, Datenschutz und Datensicherheit:
 - Die SchülerInnen können ihre Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären.
 - Die SchülerInnen können für den Schutz und die Sicherheit von Informatiksystemen, mit denen sie arbeiten, sorgen.
 - Die SchülerInnen können ihre Verantwortung beim Einsatz von Informatiksystemen sowohl in der Quantität als auch in der Qualität reflektieren.

- Die SchülerInnen können verschiedene Schutzmaßnahmen für Daten und IT-Systeme beurteilen und empfehlen.
- Die SchülerInnen können die Rechtskonformität einer Website in Grundzügen bewerten.
- Technische Grundlagen und Funktionsweisen:
 - Die SchülerInnen können Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären.
 - Die SchülerInnen können grundlegende technische Konzepte von Informatiksystemen erklären.
- Netzwerke:
 - Die SchülerInnen können verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten und Funktionsweisen beschreiben und erklären.
 - Die SchülerInnen können verschiedene Internetdienste nutzen.
 - Die SchülerInnen können die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.
- Konzepte der Informationsverarbeitung:
 - Die SchülerInnen können wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären.
 - Die SchülerInnen können bei der Lösung konkreter Aufgaben Heuristiken, Grundprinzipien und Konzepte der Informatik anwenden und informatische Modelle gestalten.
 - Die SchülerInnen können unterschiedliche Lösungsansätze in Bezug auf zugrundeliegende Konzepte reflektieren und in konkreten Handlungssituationen bewerten.
- Algorithmen, Datenstrukturen und Programmierung:
 - Die SchülerInnen können den Algorithmusbegriff erklären.
 - Die SchülerInnen können Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben.
 - Die SchülerInnen können wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern.
 - Die SchülerInnen können Aufgaben mit Mitteln der Informatik modellieren.
 - Die SchülerInnen können Algorithmen entwerfen, diese formal darstellen, implementieren und testen.
 - Die SchülerInnen können die Schritte der Softwareentwicklung reflektieren.
 - Die SchülerInnen können die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
 - Die SchülerInnen können die Effizienz von Algorithmen bewerten.

- Die SchülerInnen können gezielt nach Programmfehlern suchen und diese korrigieren.
- Datenmodelle und Datenbanksysteme:
 - Die SchülerInnen können Daten strukturiert (in Tabellen) erfassen, abfragen, auswerten sowie Datenbanken modellieren und einfache automatisierte Datenbanklösungen entwickeln.
 - Die SchülerInnen können Datenmodelle hinsichtlich der Datentypen, Redundanz, Integrität und Relevanz bewerten.²¹⁶

5.2.3.2 Planungsmatrix

In der folgenden Planungsmatrix wird der geplante Doppelstundenverlauf (2x 50min) des Szenarios 5 anhand der Themen und Inhalte, der Lehr- und Lernziele sowie der Lehr- und Lernorganisation aufgezeigt, welche sich schlussendlich in die zu verwendeten Methoden, Medien und Sozialformen aufgliedert.

Zeit:	Lehr- / Lernziele:	Thema / Inhalt:	Lehr- / Lernorganisation:		
			Methoden:	Medien:	Sozialform:
1. Std. 05 min	Herstellung von Ruhe und Aufmerksamkeit	Begrüßung, Organisatorisches	Vortrag		LSG
1. Std. 10 min	Aktivierung des Vorwissens / Interesse wecken und Vertiefung	Wiederholung PHP Sessions, Cookies, Skripte	Reflexion		LSG
1. Std. 05 min	Hinführung zum Thema und Wissensvermittlung / Interesse wecken	Authentifizierung in Anmeldesystemen (Cookie-Replay)	Vortrag	Computer, Internet, Projektor, Cookieeditor	LV
1. Std. 30 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Erstellung eines Authentifizierungsskripts mit Base64 kodierte Cookies (scr.auth.php)	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 15 min	Aktivierung des Vorwissens / Interesse wecken und Vertiefung	Wiederholung Verschlüsselung von Cookies (SHA1, BCrypt, Salted-Hash-Verfahren)	Reflexion	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 25 min	Wissensvermittlung inkl. Festigung durch gemeinsame Übung / Aufgabenbewältigung durch erworbenes Wissen	Implementierung des Salted-Hash-Verfahrens inkl. BCrypt-Token im Authentifizierungsskript	Vortrag, Einzelarbeit	Computer, NetBeans, Webserver, Projektor	LV, LSG
2. Std. 10 min	Wissensvermittlung / Anregung zum Nachdenken	Lookup- bzw. Rainbow-Tables für die Salted-Hashwerte suchen	Vortrag, Einzelarbeit	Computer, Internet, Projektor	LSG

Tabelle 5.2.3: Planungsmatrix Szenario 5: Authentifizierung anhand des Salted-Hash-Verfahrens

²¹⁶s. *Digi.Komp*: Das Kompetenzmodell Wahlpflichtfach Informatik 2013, (Stand: 29.11.2016).

In Bezug auf die Rückkopplung und (Selbst-)Reflexion können für das Unterrichtsszenario 5 „Authentifizierung anhand des Salted-Hash-Verfahrens“ die folgenden Faktoren zur Überprüfung herangezogen werden:

- Ergebnisse der Wiederholungen
- Reaktionen der SchülerInnen
- Mitarbeit der SchülerInnen sowie zahlreich aufkommende Fragen
- Selbstreflexion der SchülerInnen

5.2.3.3 Anwendungsbeispiele

Für das fünfte Unterrichtsszenario benötigt man, neben der vorhin bereits erwähnten Theorie sowie den notwendigen Internetquellen, die Codeauszüge der zu erstellenden PHP-Skripte (scr.auth.php & scr.signin.php), um den SchülerInnen den Funktionsumfang sowie die einzelnen Programmierschritte vorab erklären zu können. Durch die abschließende Eingabe diverser Salted-Hashwerte bei „webnet77.net/cgi-bin/helpers/blowfish.pl“ (Abb. 5.2.3) oder ähnlichen Decodern wird aufgezeigt, dass die entsprechenden dekodierten Werte nicht mit dem Ursprungsdatensatz übereinstimmen.

BLOWFISH

Blowfish is capable of strong encryption and can use key sizes up to 56 bytes (a 448 bit key). The key must be a multiple of 8 bytes (up to a maximum of 56). This example will automatically pad and unpad the key to size. Because Blowfish creates blocks of 8 byte encrypted output, the output is also padded and unpadded to multiples of 8 bytes.

Encrypt/Decrypt ☒ Encrypt ☐ Decrypt

Break at **Characters**

Blowfish Key **MAX 56 Bytes** padded with 6 bytes

Blowfish Plain (or ASCII HEX if Encrypted)

Blowfish Encrypted Text (Hexadecimal) padded with 5 bytes

7213CA2A058037D089004AC49434DA75 E272C19D56D32B3296D84ACAB98207C7 DBE1F9824C4D116B636EB5278ED03FFE C3E9B225C9F0612B18679B8DB256354E	7213ca2a058037d089004ac49434da75 e272c19d56d32b3296d84acab98207c7 dbe1f9824c4d116b636eb5278ed03ffe c3e9b225c9f0612b18679b8db256354e
--	--

Abbildung 5.2.3: Lookup-Table Ergebnis für Szenario 5: Authentifizierung anhand des Salted-Hash-Verfahrens

Weitere Unterrichtsmaterialien stellen beispielsweise Anhang G „Sign in Skript (scr.signin.php)“ sowie Anhang H „Authentication Skript (scr.auth.php)“ dar, welche sich am Ende dieser Arbeit befinden.

Kapitel 6

Proof of Concept

Um das Unterrichtskonzept „Kekse ohne Salz schmecken nicht“ empirisch belegen sowie analysieren zu können, wurde für die Überprüfung ein Experteninterview mit einem Vertreter des Informatikunterrichts herangezogen. Dabei wurde der ehemalige Betreuungslehrer des eigenen fachbezogenen Praktikums in Informatik, Herr Mag. Werner Jägersberger vom BGRg 13 kontaktiert, welcher sich ohne zu zögern dafür bereit erklärte, an dem Interview teilzunehmen und dieses mit seiner langjährigen Erfahrung nach bestem Wissen und Gewissen zu beantworten.

Betrachtet man das charakteristische Umfeld des BGRg 13 in der Wenzgasse 7 im 13. Wiener Gemeindebezirk etwas näher, so fällt einem auf, dass diese Schule mit seinen fast 100 Lehrbediensteten eine eher große Bildungseinrichtung, verglichen zu anderen Wiener Gymnasien darstellt, welche mit einem breit gefächerten Programm beziehungsweise schulischem Angebot auftrumpfen kann. Dabei unterteilt sich das BGRg 13 in drei unterschiedliche Schulformen, welche nun kurz im Detail beschrieben werden:

- Bundesgymnasium: Schwerpunkt liegt auf Sprachen („learning by doing“-Konzept: ab der 3. Klasse Wahl zwischen Latein oder Französisch und ab der 5. Klasse eine dritte Fremdsprache wie Spanisch, Latein oder Italienisch als Vertiefung sowie Festigung von Fertigkeiten im freien Reden)
- Realgymnasium: Schwerpunkt liegt auf Naturwissenschaften (fundierte Grundlage für eine eventuell darauffolgende universitäre Ausbildung, wodurch Gegenstände wie Mathematik, Physik oder Geometrisches Zeichnen sowie eine zusätzliche Fremdsprache in den Vordergrund rücken)
- wirtschaftskundliches Realgymnasium: praktische Sparte der Wenzgasse (erhöhtes Pensum an Werkerziehung, Chemie, Kochen oder Haushaltsökonomie, aber auch Fremdsprachen und Psychologie)²¹⁷

Wie vorhin bereits erwähnt ist das BGRg 13 ein sehr großes Schulgebäude, um sei-

²¹⁷ s. *BGRg 13 Wenzgasse: Schulformen*, (Stand: 30.11.2016).

nen drei Schulformen genügend Platz bieten zu können. Zusätzlich zu den Klassen- und Funktionsräumlichkeiten bietet die Schule ein vielfältiges Angebot. Neben einem klassischen Buffet im Erdgeschoss, einer eigenen psychologischen Betreuung und zwei Schulärztinnen gibt es eine Bibliothek und einen aus besonderem Grund angelegten Schulgarten. Dieser Schulgarten wurde unter dem Leitbild "(an)schauen – (dazu)lernen – (er)leben – (los)lassen" entwickelt und wird von den SchülerInnen eigenhändig betreut. Dort findet man eine große Vielfalt verschiedenster Pflanzen und Faunen wie z.B. den „Wienerwaldrand“ oder die „Pannonische Ecke“. Auf diese Art und Weise soll es den Jugendlichen ermöglicht werden, im Biologieunterricht hautnahe Erlebnisse machen zu können.²¹⁸

Auch in der eigens angelegten Tagesbetreuung der Schule können die SchülerInnen diesen Garten nutzen, um ihren Bedürfnissen, neben den geregelten Nachmittagsabläufen, nachzugehen. Hier sollen innerhalb einer gemütlich angelegten Atmosphäre die Interessen der SchülerInnen nicht zu kurz kommen und selbstverständlich auch die Lern- und Freizeit bestmöglich durch einen guten Umgang miteinander gefördert werden. Auch der notwendige Spielraum wird gegeben, um bei konkreteren Problemen weiterzuhelfen, indem beispielsweise unter Einbeziehung der Eltern mit den SchülerbetreuerInnen zusammengearbeitet wird.²¹⁹

6.1 Experteninterview

Der hier ausgewählte Experte Mag. Werner Jägersberger, der im Zuge des Interviews als Gesprächspartner zur Verfügung stand, arbeitet als Professor für Informatik, Mathematik und Physik am BGRg 13 und ist derzeit als Informatik- und Physikprofessor in den Klassen tätig. Zur Zeit unterrichtet er insgesamt 16 Klassen in denen die SchülerInnen nach eigenen Angaben alle größtenteils sehr motiviert und besonders arbeitsfreudig sind. Innerhalb seines Unterrichts verwaltet jede SchülerIn sogar einen eigenen Webpace, was heutzutage nach wie vor sehr selten an österreichischen Schulen ist. Außerdem evaluiert Herr Mag. Werner Jägersberger nach jeder Einheit seine SchülerInnen über die eigens programmierte Lernplattform, um ihnen somit eine größtmögliche Transparenz bieten zu können.²²⁰

In den folgenden Unterkapiteln wird nun das gesamte Experteninterview mit Herrn Mag. Werner Jägersberger zum Unterrichtskonzept „Kekse ohne Salz schmecken nicht“ aufgelistet. Dabei wird zuerst näher auf die entwickelten Fragestellungen eingegangen, bevor diese anschließend in beantworteter Form abgebildet werden.

²¹⁸s. BGRg 13 Wenzgasse: Unser Schulgarten, (Stand: 30.11.2016).

²¹⁹s. BGRg 13 Wenzgasse: Tagesbetreuung, (Stand: 30.11.2016).

²²⁰s. BGRg 13 Wenzgasse: Mag. Werner Jägersberger, (Stand: 30.11.2016).

6.1.1 Fragebogen

Im Folgenden wird nun der erstellte Fragebogen des Experteninterviews einschließlich der eigens dafür entwickelten Fragestellungen angeführt. Im Zuge der Erstellung wurde besonderer Wert darauf gelegt, die Intensität aller Anforderungsbereiche des kompetenzorientierten Unterrichts abzufragen, um Aufschluss über die Reproduktions-, Transfer-, Reflexions- oder Problemlösungsleistung²²¹ innerhalb des gesamten Unterrichtskonzeptes geben zu können. Dabei wurden die folgenden Fragestellungen ausgearbeitet:

1. Geben Sie Aufschluss über die innere Kohärenz eines jeden Unterrichtsszenarios in Bezug auf den Unterrichtseinstieg, dem methodischen sowie inhaltlichen Aufbau, den Transfer- sowie Übungsphasen inkl. den dazugehörigen Musterbeispielen und Rückkopplungen. Zieht sich ein roter Faden durch das gesamte Unterrichtskonzept?
2. Wie kann dieses Unterrichtskonzept in den jeweiligen Jahresplanungen untergebracht werden? Würden Sie die Szenarien der Schulstufe nach anders anordnen?
3. Wie realistisch ist das Zeitmanagement gewählt? Sehen Sie Engpässe bzw. Zeitüberschüsse? Würden Sie bestimmte Phasen zeitlich anders strukturieren?
4. Kann das Unterrichtskonzept auch anderweitig angewandt werden bzw. können die SchülerInnen abseits des Unterrichts davon profitieren oder stellt es eine reine Reproduktionsleistung dar?
5. Motivieren die einzelnen Szenarien bzw. könnten sie Ihrer Meinung nach an gewissen Stellen anregender gestaltet sein? Wie schätzen Sie dabei den allgemeinen Schwierigkeitsgrad ein bzw. passt dieser zur jeweils gewählten Schulstufe?
6. Sind die digitalen Grundkompetenzen der jeweiligen Szenarien realisierbar? Nennen Sie 2-3 digitale Grundkompetenzen je Einheit, welche besonders zur Geltung kommen.
7. Benoten Sie bitte abschließend das gesamte Unterrichtskonzept „Kekse ohne Salz schmecken nicht“ nach dem Schulnotensystem und begründen Sie Ihre Wahl.

6.1.2 Ergebnis

Während der Durchführung des Experteninterviews wurden die gegebenen Antworten mittels Tonaufzeichnung aufgenommen und anschließend in ihrer Originalfassung transkribiert. Die aus den Tonaufnahmen resultierende Transkription repräsentiert dabei das Ergebnis des Experteninterviews:

1. Die innere Kohärenz sowie der inhaltliche Aufbau des Unterrichtskonzeptes „Kekse ohne Salz schmecken nicht“ wird durch den klar zu erkennenden roten Faden legitimiert, welcher sich von HTML über PHP bis hin zu Websicherheit zieht. Durch die Erstreckung über mehrere Unterrichtsjahre wird auch die modulare Aufbauform

²²¹ s. *Rechtsinformationssystem*: Gesamte Rechtsvorschrift für Prüfungsordnung AHS 2012, (Stand: 30.11.2016).

widergespiegelt, welche sich für den Einsatz an Allgemein höher bildenden Schulen ideal eignet. Lediglich gegen Ende des Konzeptes ist eventuell die Frage zu stellen, ob der rote Faden überhaupt so weit reichen sollte, da der Experte der Ansicht ist, dass PHP-Datenbanken immer mehr in den Hintergrund geraten werden und das Benutzermanagement zunehmend vom Webserver, in Form von sicheren LDAP-Verbindungen oder ähnlichem übernommen wird, wodurch unter anderem der Zugang zu mehreren Datenbanken durch ein einziges Passwort ermöglicht wird. Die geplanten Transfer- und Übungsphasen bieten im gesamten Unterrichtskonzept viel Platz für die Selbstreflexion der SchülerInnen, wodurch auch von sonstigen Anordnungsmöglichkeiten in diesem Bereich abgeraten und das Konzept im Ganzen als gut inszeniert befunden wurde.

2. Die Szenarien sind allesamt der Schulstufe entsprechend nach angeordnet und es bedarf keiner sonstigen Anpassung. Der Experte zeigte sich darüber sehr begeistert, da er selbst in der 6. Klasse im ersten Halbjahr Programmieren sowie im zweiten Halbjahr Datenbanken unterrichtet, wodurch sich die Szenarien 3 und 4 des Konzeptes ideal einbinden lassen würden. Das fünfte Szenario würde er eher in der 7. Klasse heranziehen, da man hierbei dann auch ausführlicher über Tabellenrechte und dergleichen mit den SchülerInnen sprechen könnte.
3. Der korrekte Ablauf des gewählten Zeitmanagements ist bei diesem Konzept stark von der zu unterrichtenden Klasse abhängig. So entscheidet das Leistungspotential der Klasse darüber, ob die Szenarien 2 bis 5 so durchführbar sind wie sie innerhalb der Planungsmatrizen geplant sind. Prinzipiell sieht der Experte aber dadurch keine sonderlich große Schwäche innerhalb des Konzeptes, da in der Praxis die Auslagerungsmöglichkeit nach wie vor herangezogen werden kann, um Punkte, welche sonst auf der Strecke bleiben würden, in einer Folgestunde nachzuholen. Durch diese Methode kann auch in weiterer Folge sichergestellt werden, dass die äußerst schön gestalteten Übungs- sowie Reflexionsphasen für die SchülerInnen zeitlich nicht darunter leiden. Allerdings soll das erste Szenario bezüglich dem Zeitmanagement besonders hervorgehoben werden, da dieses Szenario von seinen zeitlichen Rahmenbedingungen her am realistischsten gestaltet ist.
4. Eine reine Reproduktionsleistung seitens der SchülerInnen kann während des gesamten Konzeptes in keiner Form erkannt werden, wodurch es in seiner Abstraktion sehr multipel einsetzbar und vor allem extrem profitabel für eventuelle spätere berufliche Werdegänge sein kann. Betrachtet man alleine die allgemeinen sowie reflexiven Kompetenzstufen des Schulunterrichts, wird während der fünf Szenarien wirklich bis zur letzten Kompetenz alles ausgeschöpft, was sich als sehr vorteilhaft gegenüber herkömmlichen Annäherungsmethoden auswirkt. Wie vorhin bereits angemerkt, sollte man eventuell die Detailliertheit in manchen Klassen etwas anpassen, sodass die kompetenzorientierte Vielfalt des Unterrichtskonzepts nicht an der Gesamtleistung

der jeweiligen Klasse leidet. Selbstverständlich ist diese Entscheidung stark vom Schultyp abhängig, wodurch manche Subthemen beispielsweise in einer AHS einfach weggelassen werden können, sofern diese nicht allzu elementar sind. Als besonders spannend entpuppten sich die wirklich schön verzierten Inputs sowie die stets darauffolgenden Reflexionsphasen, durch die ebenfalls eine Wissensvermittlung durch Nachdenken stattfinden kann. Diesen Ansatz fand der Experte äußerst gelungen und wollte ihn an dieser Stelle noch besonders hervorheben.

5. Die Frage der Motivation ist selbstverständlich stark von der jeweiligen SchülerIn abhängig. Nimmt man als Anhaltspunkt eine SchülerIn, welche sich ausnehmend für das Unterrichtsfach Informatik begeistert, sind die Inputs, welche jeweils am Schluss der jeweiligen Szenarien angesiedelt sind, äußerst informativ und motivierend gestaltet. Durch die stetige Reflexion und den Nachdenkphasen der SchülerInnen, werden diese jeweils zur Verbesserung gefordert sowie in ihren kombinatorischen Fähigkeiten durch explizite Transferphasen optimal geschult. Auch das Salted-Hash-Verfahren, welches sich sehr schön im Arbeitstitel des Konzeptes widerspiegelt und vor allem ein äußerst wichtiges Verfahren darstellt, bietet einen wirklich aktuellen Zugang für SchülerInnen, um die Thematik der Websicherheit einmal von einer komplett neuen Seite aus aufzugreifen und zu betrachten.
6. Der Experte möchte sich bei dieser Frage viel eher auf den Terminus „informatische Kompetenzen“ einigen, da er der Meinung ist, dass digitale Grundkompetenzen eigentlich mit grundlegenden informatischen Begriffen gedeckt sein sollten und diese im Prinzip in der 5. Klasse abgearbeitet werden. Deswegen sieht er auch nur beim ersten Szenario einen wirklichen Konnex zu „Bedeutung von Informatik in der Gesellschaft“, wobei alle anderen Grundkompetenzen in den jeweiligen Szenarien ohne Weiteres erreicht werden können. Dem Experten ist es aber dennoch wichtig anzumerken, dass die drei „informatischen Kompetenzen“ Sicherheit, Gestaltung und Datenbanken, welche sich durch das gesamte Konzept erstrecken, solche Unterrichtsszenarien viel exakter beschreiben können als jene, welche vom Bildungsministerium zur Verfügung gestellt werden.
7. Das gesamte Unterrichtskonzept ist dem Experten nach mit einem „Gut“ zu beurteilen. Diese Note erschließt sich seiner Meinung nach aus zwei einfachen Gründen. Auf der einen Seite sind die einzelnen Szenarien allesamt inhaltlich sehr interessant und spannend gestaltet sowie didaktisch und methodisch unglaublich gut aufbereitet. Andererseits könnten beim Zeitmanagement eventuelle Engpässe auftreten, welche allerdings durch zusätzliche Stundenadaptionen ausgeglichen werden können und in der Praxis relativ einfach umsetzbar sind. Durch das breite Spektrum an „informatischen Kompetenzen“ sowie durch die digitalen Grundkompetenzen für die 5. Klasse, kann eine sehr kompetenzorientierte Unterrichtsreihe gestaltet werden, welche SchülerInnen begeistern und für diese Thematik zunehmend sensibilisieren wird.

6.2 SWOT-Analyse

Um die jeweiligen Stärken und Schwächen des Unterrichtskonzeptes besser aufzeigen und veranschaulichen zu können, wurde die weiterentwickelte Form der „Stärken-Schwächen-Analyse“, die sogenannte „SWOT-Analyse“, als Analysetool herangezogen. Dabei stand vor allem jene Abwägung im Vordergrund, ob das Unterrichtskonzept als solches auch tatsächlich für den Unterrichtseinsatz geeignet ist bzw. welche Komponenten daran verbesserungswürdig erscheinen.

Ursprünglich entwickelte sich die SWOT-Analyse, welche sich aus den Stärken und Schwächen (i.e. „Strengths and Weaknesses“) sowie Chancen und Risiken (i.e. „Opportunities and Threads“) zusammensetzt, aus der Betriebswirtschaft und führte im Gegensatz zur „Stärken-Schwächen-Analyse“ (nur unternehmensbezogene Seite) Markt- und Unternehmensanalyse in einer integrativen Betrachtung zusammen.²²² Diese werden vorab getrennt voneinander durchgeführt, um im Anschluss daran, durch eine Gegenüberstellung beider, ein Handlungsportfolio zu erstellen, aus dem sich weitere Handlungsstrategien ableiten lassen (Abb. 6.2.1).²²³

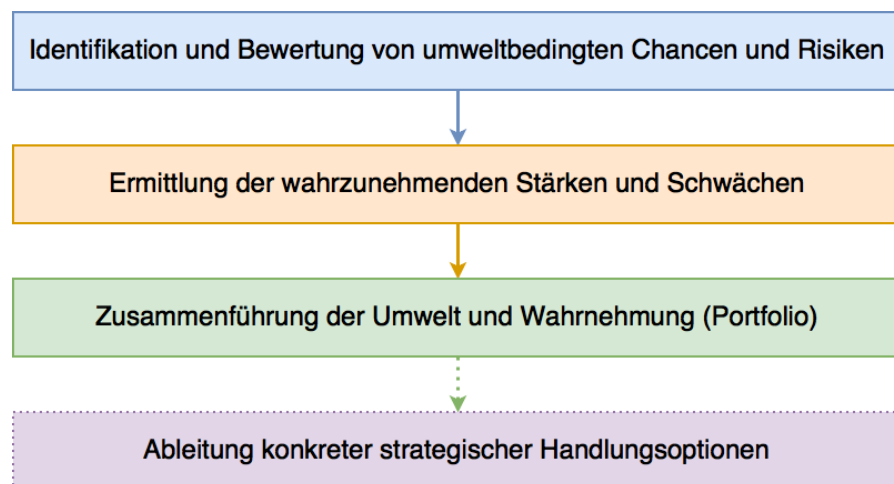


Abbildung 6.2.1: Ablaufdiagramm einer SWOT-Analyse ²²⁴

Münzt man die gesamte Thematik von der Betriebswirtschaftslehre auf die hier vorliegende empirische Studie um, so müssen zunächst einmal die allgemeinen und schul- bzw. klassenspezifischen Umweltgegebenheiten analysiert werden, um auf die diversen Chancen und Risiken des Unterrichtskonzeptes zu stoßen. Dabei ist besonders die Bewertung der Wichtigkeit der einzelnen Chancen und Risiken vonnöten, um sich auf die relevanten unterrichtsbezogenen Aspekte konzentrieren zu können. Bei der Analyse der wahrzu-

²²²vgl. Piercy (1992), S.260.

²²³vgl. Herrmann & Huber (1998), S.76 f.

²²⁴ebd., S.77.

nehmenden Eigenschaften liegt das Hauptaugenmerk auf dem Sammeln von Stärken und Schwächen des Untersuchungsbereiches, bezogen auf konkrete Teilprozesse wie z.B. didaktischer Aufbau, Inhalt oder Methodik. Schlussendlich werden die Stärken und Schwächen mit den zuvor identifizierten Chancen und Risiken hinsichtlich ihrer Relevanz bewertet.²²⁵ Durch die Gegenüberstellung aller Punkte kann ein entsprechendes Portfolio abgeleitet werden (Abb. 6.2.2), von denen folgende Handlungsstrategien resultieren können:

- Matching-Strategien: Diese können aus einer Stärke und einer dazu passenden Chance abgeleitet werden. Dazu muss vorab untersucht werden, ob es Chancen gibt, die zu einer Stärke passen. Eine Strategie ergibt sich aus der Überlegung, wie die Stärke im Hinblick auf die Chance umgesetzt werden kann.
- Umwandlungs- & Neutralisationsstrategien: Dazu werden Strategien formuliert, die Schwächen in Stärken umwandeln oder zumindest neutralisieren können. Gleiches gilt für Risiken, die sich entweder durch Strategien in Chancen umwandeln oder neutralisieren lassen.²²⁶

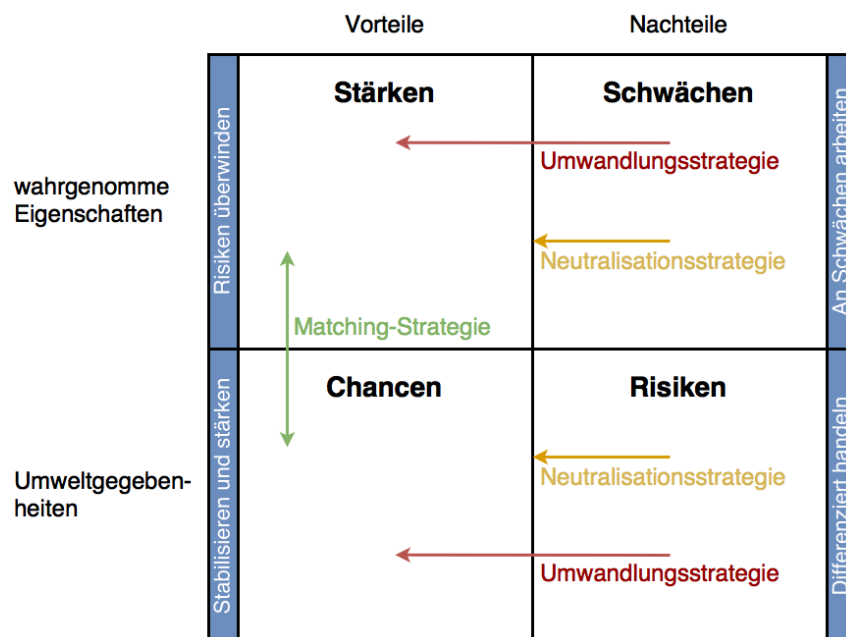


Abbildung 6.2.2: Strategieentwicklungs- bzw. Handlungsportfolio einer SWOT-Analyse²²⁷

Ein großer Vorteil der SWOT-Analyse liegt sicherlich in ihrer Einfachheit sowie Strukturiertheit. Die Analyse der oben bereits definierten Stärken, Schwächen, Chancen und Risiken kann dabei mittels Checklisten bzw. sogenannten Aktionsplänen unterstützt werden.²²⁸ Außerdem stellt die SWOT-Analyse eine sehr gute Basis für die Bestimmung von

²²⁵vgl. Herrmann & Huber (1998), S.77 f.

²²⁶vgl. Homburg (2000), S.134.

²²⁷Piercy (1992), S.260.

²²⁸vgl. Homburg (2000), S.135.

Kernfähigkeiten und -kompetenzen dar, wodurch Argumente und kulturelle Schlüssel für Veränderungen aufgezeigt werden und somit ein Anreiz für die Partizipation geschaffen werden kann.²²⁹ Allerdings liegt die Schwäche dieses analytischen Ansatzes in der Offenheit des Ableitungsvorganges der möglichen Strategien. So existieren keinerlei Leitlinien wie die strategischen Stoßrichtungen schlussendlich abgeleitet werden können, wodurch eher die Zusammenfassung der vorausgegangenen Analyseschritte im Vordergrund steht.²³⁰

6.2.1 Analyseergebnis

Um das Ergebnis des Experteninterviews mit Herrn Mag. Werner Jägersberger besser analysieren sowie veranschaulichen zu können, wurde in der folgenden Tabelle (Tab. 6.2.1) das Handlungsportfolio der SWOT-Analyse nachgebildet, um die jeweiligen Stärken, Schwächen, Chancen und Risiken des Unterrichtskonzeptes in strukturierter Form nochmals abbilden zu können.

		Vorteile	Nachteile
wahrgenommene Eigenschaften	Stärken	• roter Faden ist deutlich erkennbar • allgemeine sowie reflexive Kompetenzstufen werden ausnahmslos erreicht • viele „informatische Kompetenzen“ erreichbar • deckt breites Spektrum der Informatik ab • methodisch und didaktisch sehr gut aufgebaut • viele Selbstreflexionsphasen der SchülerInnen • ideal geeignet für den Unterrichtseinsatz • wunderschön verzierte Inputs und Übungen • äußerst informative und spannende Inputs • Reflexion mit vertiefender Wissensvermittlung • kombinatorische Fähigkeiten im Transfer	Schwächen • Zeitmanagement ab Szenario 3 zu dicht • (digitale Grundkompetenzen nur bis zur 5. Klasse sinnvoll)
	Chancen	• praktische modulare Aufbauform • Vertiefung durch Stundenadaptionen • entsprechende Anordnung nach Schulstufe • keine reine Reproduktionsleistung • multiple Einsatzmöglichkeiten • stetige Forderung nach Verbesserung • sehr aktuelles und wichtiges Thema	Risiken • Detailliertheit der Subthemen • Leistungspotential der Klasse • Motivation der SchülerInnen
Umweltgegebenheiten			

Tabelle 6.2.1: Ergebnis der SWOT-Analyse im Handlungsportfolio

²²⁹vgl. Handelsblatt Management Bibliothek (2005), S.128 f.

²³⁰vgl. Homburg (2000), S.135.

Beim Handlungsportfolio der SWOT-Analyse fallen sofort die zahlreichen Stärken und Chancen des Unterrichtskonzeptes auf, welche es mittels zu entwickelnder Matching-Strategien noch mehr in Szene zu setzen gilt. So können beispielsweise die Chancen der praktischen modularen Aufbauform sowie der entsprechenden Anordnung der Schulstufe nach aufgegriffen werden, um die Stärken des klar erkennbaren roten Fadens sowie der Abdeckung eines breiten Spektrums umsetzen zu können. Im weiteren Verlauf kann noch eine andere Matching-Strategie erkannt werden, in welcher die Stärken der erreichbaren Kompetenzstufen und „informatischen Kompetenzen“ sowie den umfassenden Reflexionsphasen, durch die Chancen der nicht vorhandenen Reproduktionsleistung sowie der stetigen Forderung nach Verbesserung und den multiplen Einsatzmöglichkeiten umgesetzt werden können. Man sieht also, dass die paarweise Zusammenführung von Stärken und Chancen zu einer Vielzahl an Matching-Strategien führt, welche die einzelnen Punkte nochmals explizit begründen und die genauen Vorteile besser veranschaulichen können.

Sieht man sich nun die Schwächen und Risiken des Handlungsportfolios an, so können diverse Umwandlungs- bzw. Neutralisationsstrategien daraus abgeleitet werden, wodurch wiederum neue Stärken und Chancen des Unterrichtskonzeptes entstehen. Eine erste Neutralisationsstrategie für die potentielle Schwäche des etwas zu dicht gewählten Zeitmanagements stellt beispielsweise die Stundenadaption zusätzlicher Einheiten dar, welche ebenfalls als eigene Chance innerhalb des Handlungsportfolios angeführt ist. Dadurch könnte die notwendige Vertiefung gewährleistet werden, wodurch der Sensibilisierungsgrad der SchülerInnen signifikant gesteigert werden könnte. Außerdem erweitern bzw. übernehmen die „informatischen Kompetenzen“ das Kompetenzraster des Unterrichtskonzeptes ab der 6. Klasse, wodurch die vermeintliche Schwäche der digitalen Grundkompetenzen mittels der „informatischen Kompetenzen“ in eine weitere Stärke umgewandelt werden kann und somit an und für sich nur in Klammer zu sehen ist.

Als weitere Umwandlungsstrategie kann für die Detailliertheit der einzelnen Subthemen, welche ein potentielles Risiko darstellen, Auslassen bzw. Überspringen diverser Themen in Erwägung gezogen werden. So ist es z.B. ratsam das Thema „responsives Webdesign“ oder „PHP“ etwas zu kürzen, um beispielsweise mehr Platz für etwaige vertiefende Reflexionsphasen der SchülerInnen zu schaffen. Durch die Umwandlung dieses Risikos zu einer neuen Chance würde sich auch das Leistungspotential der zu unterrichtenden Klasse, welches ebenfalls ein weiteres potentielles Risiko darstellt, neutralisieren lassen, wodurch der Großteil an zu erwartenden Risiken innerhalb des Unterrichtskonzeptes nachhaltig beseitigt wäre.

6.3 Adaption des Unterrichtskonzeptes

Damit das Ergebnis der SWOT-Analyse und die Erkenntnisse aus der Strategieentwicklung eine Erweiterung für das Unterrichtskonzept darstellen, werden im Folgenden nun die bestehenden Kritikpunkte und möglichen Risiken der Unterrichtsszenarien in ihrem Wesen näher analysiert, sodass abschließende Adaptionen der jeweiligen Szenarien definiert werden können.

Aufgrund des Zeitmanagementproblems ist es ratsam, die einzelnen Einheiten strukturell etwas aufzulockern um noch mehr Zeit für die Selbstreflexionsphasen der SchülerInnen zu finden. So wäre es beispielsweise vorteilhaft das Thema „PHP“ kürzer zu gestalten und auch weniger Zeit bei „responsive Webdesigns“ zu verschwenden, da diese Themen nur sehr marginal mit verschlüsselten Webdaten in Verbindung stehen. Die Themenadaptionen sollten nach Leistungspotential der Klasse, der Motivation der darin befindlichen SchülerInnen sowie der eigenen Lehrerfahrung durchgeführt werden, sodass Lehrende und Lernende gleichermaßen davon profitieren können.

Die geschaffene Zeit kann außerdem für weitere wichtige Subthemen genutzt werden, welche von ihrer Thematik her ebenfalls dicht an dem breiten Spektrum des Unterrichtskonzeptes angesiedelt sind. Als Beispiele lassen sich Themen wie „XHTML“ bzw. „XML“, „Bootstrap“ oder „JavaScript“ nennen, welche in der Webentwicklung eine sehr große Rolle spielen und deswegen auch innerhalb dieses Konzeptes einen geeigneten Platz während den Vortrags- bzw. Übungsphasen finden würden. Durch die Adaption dieser zusätzlichen Themen könnte das Unterrichtskonzept noch wesentlich vielfältiger gestaltet erscheinen und den SchülerInnen einen noch größeren Pool an zu vertiefenden Themenfeldern während den Selbstreflexionsphasen bieten.

Abschließend lässt sich aufgrund dessen sagen, dass das Unterrichtskonzept sehr viel Spielraum für weitere mögliche Adaptionen bereithält und den entsprechenden Vorstellungen nach erweitert bzw. verkürzt werden kann. Das Unterrichtskonzept „Kekse ohne Salz schmecken nicht“ kann somit problemlos in jeder Jahresplanung aufgenommen werden um die eigenen SchülerInnen für Security im Webdatenbereich schnellstmöglich zu sensibilisieren und zwar ganz unabhängig davon, wie es um irgendwelche Umweltgegebenheiten steht.

Kapitel 7

Conclusio

Diese Diplomarbeit beschäftigt sich mit der Erstellung eines völlig neu entwickelten Zuganges für die Thematik rund um Security im Webdatenbereich, welcher für alle Lehrenden und Lernenden der Sekundarstufe II, aber auch im Studiengang Informatik als innovatives Unterrichtskonzept theoretisch, sowie empirisch aufbereitet und veranschaulicht wurde.

Während die Theorie eine fundierte Grundlage über aktuelle Standards in der Webentwicklung (d.h. HTML5, CSS3, PHP5.6, MySQLi), Abgrenzungen zu anderen Technologien (d.h. SSL) sowie die Relevanz für einen kompetenzorientierten Informatikunterricht bildet, umfasst die Empirie fünf modular aufbauend gestaltete Unterrichtsszenarien, welche sich in den Vortragsphasen jeweils auf die theoretischen Grundlagen und in den Übungsphasen auf das eigenständig entwickelte PHP Session-Login-System Musterbeispiel dieser Diplomarbeit beziehen. Durch das Heranziehen eines Experteninterviews am Wiener BGRg 13 und abschließender SWOT-Analyse wurde das Unterrichtskonzept abschließend noch empirisch belegt und abgerundet.

Durch die Kontaktaufnahme mit dem ehemaligen Betreuungslehrer des eigenen fachbezogenen Praktikums in Informatik, Herrn Mag. Werner Jägersberger, konnten im Zuge eines Interviews ein zuvor eigens erstellter Fragebogen zum Unterrichtskonzept beantwortet werden, indem der Experte jeweils auf seine eigene langjährige Erfahrung Bezug nimmt. Durch die Aufnahme des Gesprächs mittels Diktiergerät wurde das Interview im Anschluss transkribiert, sodass in der Folge daraus eine SWOT-Analyse erstellt werden konnte. Diese Analyse gibt gebührend Aufschluss über die genauen Vor- und Nachteile des Unterrichtskonzeptes, wodurch ein besserer Einblick in die Unterrichtstauglichkeit gewährt wird.

So erbrachte die SWOT-Analyse des Experteninterviews mit Herrn Mag. Werner Jägersberger das Ergebnis, dass die Anzahl der Stärken und Chancen des Unterrichtskonzeptes weit über jener der potentiellen Schwächen und Risiken lagen und

somit das Konzept mit einer guten Endnote seitens des Experten zu beurteilen war. So dominiert ganz klar der sich stetig durchziehende rote Faden, verbunden mit seinem groß angelegten Themenspektrum und der Erreichbarkeit etlicher allgemeiner und reflexiver Kompetenzstufen sowie zahlreicher „informatischer Kompetenzen“. Durch die praktische modulare Aufbauform und die entsprechende Anordnung nach der Schulstufe werden dabei die großzügig eingebauten Reflexionsphasen der SchülerInnen gewährleistet, welche im späteren Verlauf durch vielfältige Einsatzmöglichkeiten sowie in Transferphasen herangezogen werden können. Lediglich das Zeitmanagement bzw. die Detailliertheit einzelner Subthemen, welche sich allerdings durch eigens entwickelte Umwandlungs- bzw. Neutralisationsstrategien beseitigen lassen, können als potentielle Schwächen bzw. Risiken innerhalb des Konzeptes festgemacht werden, wodurch sich auch die Gesamtnote eines „Gut“ seitens des Experten begründen lässt.

Das Unterrichtskonzept „Kekse ohne Salz schmecken nicht“ stellt damit ein gutes innovatives Unterrichtskonzept für alle Lehrenden und Lernenden der Sekundarstufe II oder des Studienganges Informatik dar, um die Thematik Webdatensicherheit einmal von einer anderen, sehr aktuellen Seite aus zu betrachten. Durch die Berücksichtigung vieler Subelemente, welche durch den modularen Aufbau und dem eigens programmierten Musterbeispiel für persönliche Zwecke adaptiert werden können, stellt das Unterrichtskonzept eine fundierte Basis für ein gesalzenes aber zugleich methodisch sowie didaktisch wundervoll verziertes Beispiel, wie vielfältig und realitätsnahe der Informatikunterricht im Bildungsbereich heutzutage gedacht werden sollte.

Lebenslauf

Curriculum Vitae

Persönliche Daten:

Name: Simon Marik
Wohnsitz: A-1170 Wien
E-Mail: simon.marik@me.com
geb. am: 07.04.1989 in Wien

Familie:

Eltern: Elisabeth und Peter Marik - selbstständig
(PMA Mobile Electronics GmbH)
Geschwister: Mag. Petra Marik, MA - Pädagogin

Schulbildung:

1995–1999 Volksschule Kitzbühel
1999–2003 Bundesgymnasium St. Johann in Tirol
2003–2004 Höhere Technische Bundeslehr- und Versuchsanstalt Saalfelden
Abteilung: Elektrotechnik
2004–2009 Höhere Technische Bundeslehr- und Versuchsanstalt Mödling
Abteilung: Wirtschaftsingenieurwesen

Studium:

2010–2017 Universität Wien: Lehramtsstudium Informatik und Informatikmanagement & Geschichte, Sozialkunde und Politische Bildung

Wien, 2017

Literatur- & Quellenverzeichnis

Anderson, Ross J. (2008): Security Engineering - A Guide to Building Dependable Distributed Systems; 2. Auflage, Wiley Publishing, Indianapolis.

Beighley, Lynn & Morrison, Michael (2009): PHP & MySQL von Kopf bis Fuß; übersetzt von: Schulten, Lars; 1. Auflage, O'Reilly Verlag, Köln.

Brandl, Ernst O. & Mayer-Schönberger, Viktor (1999): CPU-IDs, Cookies und Internet-Datenschutz; In: Ecolex, Heft Nr. 5, Manz Verlag, Wien.

Brikman, Yevgeniy (2016): Hello, Startup - A Programmer's Guide to Building Products, Technologies, and Teams; 1. Auflage, O'Reilly Verlag, Sebastopol.

Christl, Alexander (2014): Datenschutz im Internet - Cookies, Web-Logs, Location Based Services, eMail, Webbugs, Spyware; 1. Auflage, Disserta Verlag, Hamburg.

Clarke, Justin (2012): SQL Injection Attacks and Defense; 2. Auflage, Elsevier Verlag, Waltham.

Ebner, Alexander; Bold, Max (Hrsg.) (2013): Webdesign mit CSS3 - Workshops und Praxistipps für das Webdesign mit CSS3; 1. Auflage, epubli Verlag, Berlin.

Fuchs, Florian (2001): Sicherheit von Internet- und Intranetdiensten; Universität Klagenfurt, Klagenfurt.

Günster, Kai (2013): Schrödinger lernt HTML5, CSS3 und JavaScript - Das etwas andere Fachbuch; 1. Auflage, Rheinwerk Verlag, Bonn.

Hagelkruys, Dominik (2012): Barrierefreiheit als Thematik im österreichischen Informatikunterricht; Universität Wien, Wien.

Hammer, Norbert (2008): Mediendesign für Studium und Beruf - Grundlagenwissen und Entwurfssystematik in Layout, Typografie und Farbgestaltung; 1. Auflage, Springer Verlag, Berlin.

Handelsblatt Management Bibliothek (2005): Die besten Checklisten für Manager L - Z; Band 12, Campus Verlag, Frankfurt.

Heller, Stephan (2012): Workshop HTML5 & CSS3 - Weblayouts professionell umsetzen - ein Einstieg in die Frontendentwicklung; 1. Auflage, dpunkt Verlag, Heidelberg.

Herrmann, Andreas & Huber, Frank (1998): Produktmanagement - Grundlagen, Methoden, Beispiele; 3. Auflage, Gabler Verlag, Wiesbaden.

Hogan, Brian P. (2011): HTML5 & CSS3 - Webentwicklung mit den Standards von morgen; übersetzt von: Fröhlich, Stefan; 1. Auflage, O'Reilly Verlag, Köln.

Homburg, Christian (2000): Quantitative Betriebswirtschaftslehre - Entscheidungsunterstützung durch Modelle; Gabler Verlag, Wiesbaden, 3. Auflage.

Horn, Janine (2015): Rechtliche Aspekte digitaler Medien an Hochschulen; 1. Auflage, Waxmann Verlag, Münster.

Hudson, Paul (2005): PHP in a nutshell; übersetzt von: Speidel, Sigrid & Ulrich; 1. Auflage, O'Reilly Verlag, Köln.

Jahnel, Dietmar (2001): Datenschutz im Internet - Rechtsgrundlagen, Cookies und Web-Logs; In: Ecolex, Heft Nr. 1, Manz Verlag, Wien.

Jahnel, Dietmar (2003): Spamming, Cookies, Web-Logs, LBS und die Datenschutzrichtlinie für elektronische Kommunikation; In: Wirtschaftsrechtliche Blätter, Heft Nr. 3, Manz Verlag, Wien.

Jahnel, Dietmar (2004): Spamming, Cookies, Logfiles und Location Based Services im TKG 2003; In: Österreichische Juristen Zeitung, Heft Nr. 9, Manz Verlag, Wien.

Kadlec, Tim (2013): Praxiswissen Responsive Webdesign; übersetzt von: Fröhlich, Stefan; 1. Auflage, O'Reilly Verlag, Köln.

Knuckles, Craig D. & Yuen, David S. (2005): Web Applications - Concepts & Real World Design; 1. Auflage, Wiley Publishing, Michigan.

Kofler, Michael & Öggel, Bernd (2010): PHP 5.3 & MySQL 5.4 - Programmierung, Administration, Praxisprojekte; 1. Auflage, Addison-Wesley Verlag, München.

Khu-smith, Vorapranee & Mitchell, Chris; Kim, Kwangjo (Hrsg.) (2002): Information Security and Cryptology - ICISC 2001; 1. Auflage, Springer Verlag, Seoul.

Lassmann, Wolfgang (2006): Wirtschaftsinformatik - Nachschlagewerk für Studium und Praxis; 1. Auflage, Gabler Verlag, Wiesbaden.

Mahle, Niko (2004): Das technische und strategische Potential von SVG als standardisierte Grafik-Technologie für grafikorientierte Browser-Anwendungen bei DEKRA; Berufsakademie Stuttgart, Stuttgart.

Maurice, Florence (2012): Mobile Webseiten - Strategien, Techniken, Dos und Don'ts für Webentwickler; 1. Auflage, Carl Hanser Verlag, München.

Maurice, Florence (2015): PHP 5.6 und MySQL 5.7 - Ihr praktischer Einstieg in die Programmierung dynamischer Websites; 4. Auflage, Dpunkt Verlag, Heidelberg.

Mogollon, Manuel (2007): Cryptography and Security Services - Mechanisms and Applications; 1. Auflage, Cybertech Publishing, Hershey.

Müller, Peter (2013): Flexible Boxes - Eine Einführung in moderne Websites; 1. Auflage, Rheinwerk Verlag, Bonn.

Peyton, Christine (2002): Das neue PC Lexikon für Alle; 1. Auflage, Sybex Verlag, Düsseldorf.

Piercy, Nigel (1992): Market-Led Strategic Change - Making Marketing Happen in your Organization; 2. Auflage, Butterworth-Heinemann Ltd., Oxford.

Pomaska, Günter (2005): Grundkurs Web-Programmierung - Interaktion, Grafik und Dynamik: Mit XHTML und CSS, XML, JavaScript, Applets, SVG, PHP; 1. Auflage, Vieweg & Sohn Verlag, Wiesbaden.

Potschien, Denis (2013): Pure HTML5 und CSS3; 1. Auflage, Franzis Verlag, München.

Rhee, Man Young (2003): Internet Security - Cryptographic Principles, Algorithms and Protocols; 1. Auflage, Wiley Publishing, Chichester.

Niederst Robbins, Jennifer (2014): HTML5 - kurz & gut; übersetzt von: Lang, Jörgen W.; 5. Auflage, O'Reilly Verlag, Köln.

Schäfers, Tim Philipp (2016): Hacking im Web - Denken Sie wie ein Hacker und schließen Sie die Lücken in ihrer Webapplikation, bevor diese zum Einfallstor für Angreifer wird; 1. Auflage, Franzis Verlag, München.

Schlossnagle, George (2006): Professionelle PHP 5 - Programmierung; 1. Auflage, Addison-Wesley Verlag, München.

Splaine, Steven (2002): Testing Web Security - Assessing the Security of Web Sites and Applications; 1. Auflage, Wiley Publishing, Indianapolis.

Theis, Thomas (2014): Einstieg in PHP 5.6 und MySQL 5.6; 10. Auflage, Rheinwerk Verlag, Bonn.

Voss, Andreas (2003): Das große PC & Internet Lexikon; 1. Auflage, Data Becker Verlag, Düsseldorf.

Wassermann, Tobias (2007): Sichere Webanwendungen mit PHP; 1. Auflage, Mitp Verlag, Heidelberg.

Williams, Hugh E. & Lane, David (2005): Webdatenbank-Applikationen mit PHP und MySQL; 2. Auflage, O'Reilly Verlag, Köln.

Zacharias, Mark (2002): Ecommerce Grundlagen in der Praxis; 1. Auflage, Mazscreen Media, Hamburg.

Zander, Tobias (2014): Security im E-Commerce - Absicherung von Shopsystemen wie Magento, Shopware und OXID; 1. Auflage, Entwickler Press, Frankfurt.

Allsopp, John: A Dao of Web Design; siehe online unter: <http://alistapart.com/article/dao> (Stand: 24.09.2016).

Artworkz: Die Cookie-Richtlinie der EU; siehe online unter: <http://www.artworx.at/die-cookie-richtlinie-der-eu-in-oesterreich-ist-opt-in-fuer-personenbezogene-daten-gesetz/> (Stand: 05.10.2016).

BGRg 13 Wenzgasse: Mag. Werner Jägersberger; siehe online unter: <http://www.wenzgasse.at/index.php?ID=33&KL=24> (Stand: 30.11.2016).

BGRg 13 Wenzgasse: Schulformen; siehe online unter: <http://www.wenzgasse.at/index.php?ID=21> (Stand: 30.11.2016).

BGRg 13 Wenzgasse: Tagesbetreuung; siehe online unter: <http://www.wenzgasse.at/index.php?ID=51> (Stand: 30.11.2016).

BGRg 13 Wenzgasse: Unser Schulgarten; siehe online unter: <http://schulgarten.wenzgasse.at> (Stand: 30.11.2016).

Blogstories: Vor- und Nachteile einer SSL-Verschlüsselung; siehe online unter: <http://blogstories.de/vor-und-nachteile-ssl/> (Stand: 18.10.2016).

Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 5. Klasse 2004; siehe online unter: https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf (Stand: 24.10.2016).

Bundesministerium für Bildung: Informatik Lehrplan - Sekundarstufe II. - 6.-8. Klasse 2004; siehe online unter: https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_21_11876.pdf (Stand: 24.10.2016).

Bundesministerium für Bildung: Lehrpläne der AHS-Oberstufe Überblick; siehe online unter: https://www.bmb.gv.at/schulen/unterricht/lp/lp_ahs_oberstufe.html (Stand: 21.10.2016).

Bundesministerium für Bildung: Schulrecht Überblick; siehe online unter: <https://www.bmb.gv.at/schulen/recht/index.html> (Stand: 20.10.2016).

Bundesministerium für Bildung: Verordnung über die Änderung der AHS Lehrpläne 2004; siehe online unter: https://www.bmb.gv.at/schulen/unterricht/lp/11668_11668.pdf (Stand: 21.10.2016).

Brünner, Arndt: Der Kodierungsstandard Base64; siehe online unter: <http://www.arndt-bruenner.de/mathe/scripts/base64.htm> (Stand: 11.10.2016).

Code-Bude: Sicheres Passwort Hashing mit Salts; siehe online unter: <http://code-bude.net/2015/03/30/grundlagen-sicheres-passwort-hashing-mit-salts/> (Stand: 12.10.2016).

Cookie Central: The Cookie Concept; siehe online unter: http://www.cookiecentral.com/c_concept.htm (Stand: 04.10.2016).

Der Standard: Bildungsreform - Mehr Macht für Direktoren; siehe online unter: <http://derstandard.at/2000045996534/Bildungsreform-Mehr-Macht-fuer-Direktoren> (Stand: 20.10.2016).

Digi.Komp: Das Kompetenzmodell Informatik 5. Klasse 2013; siehe online unter: <http://digikomp.at/praxis/portale/digitale-kompetenzen/digikomp12ahs/kompetenzmodelle/informatik-5-klasse.html> (Stand: 19.09.2016).

Digi.Komp: Das Kompetenzmodell Wahlpflichtfach Informatik 2013; siehe online unter: <http://digikomp.at/praxis/portale/digitale-kompetenzen/digikomp12ahs/kompetenzmodelle/wahlpflichtfach-informatik.html> (Stand: 19.09.2016).

Doumack, Andrej: Symmetrische Verschlüsselung mit Blowfish-Algorithmus; siehe online unter: http://www.gm.fh-koeln.de/~hk/lehre/ala/ws0506/Praktikum/Projekt/C_rot/Blowfish.pdf (Stand: 12.10.2016).

Draw.io: Flowchart Maker & Online Diagram Software; siehe online unter: <https://www.draw.io> (Stand: 24.09.2016).

Envatotuts Plus: PHP 5.6 What's New; siehe online unter: <http://code.tutsplus.com/articles/php-56-whats-new--cms-22101> (Stand: 28.09.2016).

Gatliff, Bill: Encrypting Data with the Blowfish Algorithm; siehe online unter: <http://www.embedded.com/design/configurable-systems/4024599/Encrypting-data-with-the-Blowfish-algorithm> (Stand: 12.10.2016).

Internet Assigned Numbers Authority: Service Name and TCP Port Number Registry; siehe online unter: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt> (Stand: 18.10.2016).

ITWissen: SHA (Secure Hash Algorithm); siehe online unter: <http://www.itwissen.info/definition/lexikon/secure-hash-algorithm-SHA-SHA-Algorithmus.html> (Stand: 11.10.2016).

Jusline: §11 (Abs. 3a) SchUG 1986 Pflichtgegenstände und verbindliche Übungen; siehe online unter: <https://www.jusline.at/index.php?cpid=ba688068a8c8a95352ed951ddb88783e&lawid=93&paid=11> (Stand: 24.10.2016).

Jusline: §17 (Abs. 1) SchUG 1986 Unterrichtsarbeit; siehe online unter: <https://www.jusline.at/index.php?cpid=ba688068a8c8a95352ed951ddb88783e&lawid=93&paid=17> (Stand: 20.10.2016).

Jusline: §57b SchUG 1986 Schülerinnen- bzw. Schülerkarte; siehe online unter: <https://www.jusline.at/index.php?cpid=fdfdb9555e1b818d0e269bad11308871&lawid=93&paid=57b> (Stand: 20.10.2016).

Jusline: §77 SchUG 1986 Schulverzeichnisse, Protokolle, Formblätter; siehe online unter: <https://www.jusline.at/index.php?cpid=ba688068a8c8a95352ed951ddb88783e&lawid=93&paid=77> (Stand: 20.10.2016).

Jusline: §77a SchUG 1986 Formblätter und Datenmuster; siehe online unter: <https://www.jusline.at/index.php?cpid=ba688068a8c8a95352ed951ddb88783e&lawid=93&paid=77a> (Stand: 20.10.2016).

Kröner, Peter: Die Karte des HTML5-Universums; siehe online unter: <http://www.peterkroener.de/die-karte-des-html5-universums> (Stand: 25.08.2016).

Loew, Markus: Das Schulorganisationsgesetz 1962; siehe online unter: http://austria-forum.org/af/Wissenssammlungen/Essays/Bildung/Das_Schulorganisationsgesetz_1962 (Stand: 20.10.2016).

Marcotte, Ethan: Responsive Web Design; siehe online unter: <http://alistapart.com/article/responsive-web-design> (Stand: 27.09.2016).

Meine-Cookies: Rechtliche Aspekte; siehe online unter: http://www.meine-cookies.org/alles_ueber_cookies/rechtliche_aspekte.html (Stand: 05.10.2016).

Mozilla Developer Network: Liste der HTML5-Elemente; siehe online unter: https://developer.mozilla.org/de/docs/Web/HTML/HTML5/HTML5_element_list (Stand: 25.09.2016).

MySQL: Encryption Functions; siehe online unter: http://dev.mysql.com/doc/refman/5.7/en/encryption-functions.html#function_sha1 (Stand: 11.10.2016).

OnPageWiki: SSL; siehe online unter: <https://de.onpage.org/wiki/SSL> (Stand: 18.10.2016).

PHP Gangsta: Schöner hashen mit BCrypt; siehe online unter: <http://www.phpgangsta.de/schoener-hashen-mit-bcrypt> (Stand: 12.10.2016).

PHP.net: Funktion crypt; siehe online unter: <http://php.net/manual/de/function.crypt.php> (Stand: 12.10.2016).

PHP.net: Funktion password_hash; siehe online unter: <http://jp2.php.net/manual/en/function.password-hash.php> (Stand: 10.10.2016).

PHP.net: Funktion password_verify; siehe online unter: <http://jp2.php.net/manual/en/function.password-verify.php> (Stand: 10.10.2016).

PHP.net: Funktion require_once; siehe online unter: <http://php.net/manual/de/function.require-once.php> (Stand: 29.09.2016).

PHP.net: Migrating from PHP 5.6.x to PHP 7.0.x; siehe online unter: <http://php.net/manual/de/migration70.php> (Stand: 28.09.2016).

PHP.net: PHP 5.6 Changed Functions; siehe online unter: <http://php.net/manual/en/migration56.changed-functions.php> (Stand: 28.09.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Datenschutzgesetz 2000; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=bundesnormen&Gesetzesnummer=10001597> (Stand: 05.10.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für LBVO 1974; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009375> (Stand: 20.10.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Lehrpläne - AHS 1984; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568> (Stand: 21.10.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Prüfungsordnung AHS 2012; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20007845> (Stand: 18.11.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Schulorganisationsgesetz 1962; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009265> (Stand: 20.10.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Schulunterrichtsgesetz 1986; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009600> (Stand: 20.10.2016).

Rechtsinformationssystem: Gesamte Rechtsvorschrift für Telekommunikationsgesetz 2003; siehe online unter: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20002849> (Stand: 05.10.2016).

Schülerunion: Leistungsbeurteilung; siehe online unter: <http://schuelerunion.at/leistungsbeurteilung/> (Stand: 20.10.2016).

Schülerunion: Schulunterrichtsgesetz 1986; siehe online unter: <http://schuelerunion.at/schulunterrichtsgesetz/> (Stand: 20.10.2016).

SelfHTML Wiki: CSS/Box-Modell; siehe online unter: <https://wiki.selfhtml.org/wiki/CSS/Box-Modell> (Stand: 26.09.2016).

SelfHTML Wiki: CSS/Media Queries; siehe online unter: https://wiki.selfhtml.org/wiki/CSS/Media_Queries (Stand: 27.09.2016).

SSL Webhosting: SSL Proxy - Vor- und Nachteile und Einsatzmöglichkeiten; siehe online unter: <http://sslwebhosting.de/ssl-proxy-vor-und-nachteile-und-einsatzmoeglichkeiten/> (Stand: 19.09.2016).

The Web Standards Project: Working together for Standards; siehe online unter: <http://www.webstandards.org/about/> (Stand: 24.09.2016).

W3C: CSS Current Status; siehe online unter: https://www.w3.org/standards/techs/css#w3c_all (Stand: 26.09.2016).

W3C: HTML5 Recommendation 2014; siehe online unter: <https://www.w3.org/TR/html5/> (Stand: 24.09.2016).

W3C: Web Content Accessibility Guidelines (WCAG) 2.0 2009; siehe online unter: <https://www.w3.org/Translations/WCAG20-de/> (Stand: 24.09.2016).

Wikiquote: Bruce Lee - Quotes; siehe online unter: https://en.wikiquote.org/wiki/Bruce_Lee (Stand: 27.09.2016).

Abbildungsverzeichnis

Abb. 2.1.1: Offizieller HTML5-Sticker inkl. Logo des W3C; siehe online unter: https://www.w3.org/html/logo/downloads/HTML5_sticker.png (Stand: 22.09.2016)	23
Abb. 2.1.2: Die Karte des „HTML5-Universums“ nach Kröner; siehe online unter: http://cdn2.peterkroener.de/uploads/2012/graph.w.png (Stand: 23.09.2016)	25
Abb. 2.1.3: Minimalbeispiel der neuen HTML5 DOCTYPE-Deklaration	27
Abb. 2.1.4: Auszeichnung der Layoutbereiche von HTML4 nach HTML5	29
Abb. 2.1.5: Das Periodensystem der HTML5-Elemente nach Duck; siehe online unter: http://jpolete.me/attachments/Periodic-Table-of-HTML-Elements.png (Stand: 24.09.2016)	31
Abb. 2.2.1: Offizielles CSS3-Klassenbild inkl. Logo des W3C; siehe online unter: https://www.w3.org/html/logo/img/class-header-css3.jpg (Stand: 22.09.2016)	32
Abb. 2.2.2: Alle gängigen Vendor-Präfixe für die CSS-Eigenschaft <code>border-radius</code>	33
Abb. 2.2.3: Neu konzipiertes Box-Modell des CSS3-Standards	34
Abb. 2.2.4: Inhalt ist wie Wasser (i.e. “Content is like water”) nach Walter; siehe online unter: https://blog.stephaniewalter.fr/wp-content/uploads/2013/06/content-is-like-water-12401.jpg (Stand: 27.09.2016)	35
Abb. 2.2.5: Anordnungsmöglichkeiten durch verschiedene Media Queries nach Pixelbreite	37
Abb. 2.2.6: Beschränkung der Styles auf die Druckausgabe per Media Query bzw. <code>meta-viewport</code>	38
Abb. 2.2.7: Offizielles Scalable Vector Graphics Logo des W3C; siehe online unter: https://www.w3.org/Icons/SVG/svg-logo-v.svg (Stand: 27.09.2016)	39
Abb. 2.3.1: Offizielles PHP Hypertext Preprocessor Logo von PHP.net; siehe online unter: http://php.net/images/logos/php-logo.eps (Stand: 22.09.2016)	41
Abb. 2.3.2: Minimalbeispiel zur Verdeutlichung der Einbettung von PHP-Code in HTML	43
Abb. 2.3.3: Mehrfacheinbindung von PHP-Skripte durch <code>include</code> oder <code>require</code>	46
Abb. 2.3.4: Ablaufplan eines PHP Session-Handlers	48
Abb. 2.3.5: Speicherung & Indizierung von Sessionvariablen anhand der Session-ID	50
Abb. 2.3.6: Minimalbeispiel zu den einzelnen Variablenoperationen innerhalb von Sessions	51
Abb. 2.3.7: Abfrage von Cookieinformationen durch Drittanbieter anderer Domänen	53

Abb. 2.3.8: Minimalbeispiel zur Erstellung und Entfernung eines Cookies	55
Abb. 2.3.9: Vorgehensweise von Angreifern bei einer „Cookie-Steal“- bzw. „Cookie-Replay-Attack“	62
Abb. 2.3.10: Kodierung einer Zeichenkette durch Base64	65
Abb. 2.3.11: Schematische Darstellung eines SHA1 Verschlüsselungszyklus	67
Abb. 2.3.12: Ablaufdiagramm eines Verschlüsselungszyklus des Blowfish-Algorithmus	69
Abb. 2.3.13: Minimalbeispiel zur Erstellung und Überprüfung eines „salted“ Hashwerts	71
Abb. 2.3.14: Salted-Hash-Verfahren bei gespeicherten Benutzerpasswörtern	71
Abb. 2.4.1: Offizielles MySQL Logo von Oracle; siehe online unter: https://www.mysql.com/common/logos/logo-mysql.eps (Stand: 22.09.2016)	73
Abb. 2.4.2: Minimalbeispiel einer SQL Abfrage inkl. DB Verbindungsaufbau	74
Abb. 2.4.3: Schematische Vorgangsweise von Angreifern bei einer „SQL Injection“	76
Abb. 2.5.1: Servernetzwerkstruktur inkl. DNS zur Darstellung einer Internetseite	80
Abb. 3.1.1: SSL Architektur inkl. Protokollaufbau	85
Abb. 4.2.1: „Word Cloud“ über den Lehrstoff des Unterrichtsfachs Informatik	94
Abb. 4.2.2: Das österreichische Schulsystem	96
Abb. 5.1.1: Diverse Cookieeditoren (Chrome & Safari) für Speicheranalyse und Bearbeitung	107
Abb. 5.1.2: Ausgeführtes Musterbeispiel für Szenario 2: Webinterfaces anhand aktueller Standards	111
Abb. 5.2.1: Ausgeführtes Musterbeispiel für Szenario 3: Der Umstieg von HTML auf PHP	116
Abb. 5.2.2: Lookup-Table Ergebnis für Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies	120
Abb. 5.2.3: Lookup-Table Ergebnis für Szenario 5: Authentifizierung mittels Salted-Hash-Verfahren	124
Abb. 6.2.1: Ablaufdiagramm einer SWOT-Analyse	130
Abb. 6.2.2: Strategieentwicklungs- bzw. Handlungsportfolio einer SWOT-Analyse	131

Tabellenverzeichnis

Tab. 2.1.1:	Ein Auszug der neu unterstützten HTML5-Elemente	29
Tab. 2.2.1:	Übersicht der Medientypen in CSS; siehe online unter: https://wiki.selfhtml.org/wiki/CSS/Media_Queries (Stand: 27.09.2016)	38
Tab. 2.2.2:	Unterscheidungsmerkmale von Vektor- & Pixelgrafiken	40
Tab. 2.3.1:	Wichtige Paragraphen zum DSG über den Schutz personenbezogener Daten; siehe online unter: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=bundesnormen&Gesetzesnummer=10001597 (Stand: 05.10.2016)	57
Tab. 2.3.2:	Das komplette Base64 Alphabet	65
Tab. 2.5.1:	„Changeable“ Spezifikation der PHP-Dokumentation	79
Tab. 2.5.2:	HTTP-Statuscodes inkl. deren Bedeutung für BenutzerInnen	81
Tab. 3.1.1:	Ports der unterschiedlichen SSL-Protokolle	84
Tab. 5.1.1:	Planungsmatrix Szenario 1: Cookies im Webdatenbereich	107
Tab. 5.1.2:	Planungsmatrix Szenario 2: Webinterfaces anhand aktueller Standards	110
Tab. 5.2.1:	Planungsmatrix Szenario 3: Der Umstieg von HTML auf PHP	115
Tab. 5.2.2:	Planungsmatrix Szenario 4: PHP Anmeldeskripts mit Base64 kodierten Cookies	119
Tab. 5.2.3:	Planungsmatrix Szenario 5: Authentifizierung anhand des Salted-Hash-Verfahrens	123
Tab. 6.2.1:	Ergebnis der SWOT-Analyse im Handlungsportfolio	132
Tab. E.0.1:	Auszug aus der Datenbanktabelle „thesis_users“	167

Abkürzungsverzeichnis

Abb.	Abbildung
Abs.	Absatz
AGB	Allgemeine Geschäftsbedingungen
AHS	Allgemein bildende höhere Schule
API	Application Programming Interface
ApiGen	Dokumentationsgenerator für PHP
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
BHS	Berufsbildende höhere Schule
BMB	Bundesministerium für Bildung
BMS	Berufsbildende mittlere Schule
BY-SA	Attribution - ShareAlike
bzw.	beziehungsweise
C#	C-Sharp
ca.	cirka
CC	Creative Commons
CGM	Computer Graphics Metafile
CORS	Cross-Origin Resource Sharing
CRL	Certificate Revocation List
CSS	Cascading Style Sheet
CSS3	Cascading Style Sheet Version 3
DB	Datenbank
DBMS	Datenbankmanagementsystem
d.h.	das heißt
Digi.Komp	Digitale Kompetenzen der Informatischen Bildung
DIN	Standard des Deutschen Instituts für Normung
DNS	Domain Name System
DSG	Datenschutzgesetz
DTD	Document Type Definition

ebd.	ebenda
ERP	Enterprise-Resource-Planning
EU	Europäische Union
f.	folgend
ff.	Plural von folgend
FTP	File Transfer Protocol
GIF	Graphical Interchange Format
GNU	GNU's Not Unix
GPL	General Public Licence
HTML	HyperText Markup Language
HTML5	HyperText Markup Language Version 5
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ID	Identifier
i.e.	id est
IIOP	Internet Inter-ORB Protocol
IMAP	Internet Message Access Protocol
inkl.	inklusive
IP	Internet Protocol
IRC	Internet Relay Chat
ISO	International Standards Organization
JPEG	Joint Photographic Experts Group
KGB	Russisches Komitee für Staatssicherheit
LBVO	Leistungsbeurteilungsverordnung
LDAP	Lightweight Directory Access Protocol
LV	LehrerInnenvortrag
LSG	LehrerInnen-SchülerInnen-Gespräch
MAC	Message Authentication Code
macOS	Macintosh Operating System
MAMP	Macintosh, Apache, MySQL, PHP
MD5	Message-Digest Algorithm 5
MIME	Multipurpose Internet Mail Extension
MySQL	My Structured Query Language
MySQLi	My Structured Query Language improved
NNTP	Network News Transfer Protocol
PHP	PHP Hypertext Preprocessor

PHP5.6	PHP Hypertext Preprocessor Version 5.6
PNG	Portable Network Graphics
POP3	Post Office Protocol Version 3
s.	siehe
SchOG	Schulorganisationsgesetz
SchUG	Schulunterrichtsgesetz
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSG	SchülerInnen-SchülerInnen-Gespräch
SSL	Secure Sockets Layer
SVG	Scalable Vector Graphics
SWOT	Strengths, Weaknesses, Opportunities und Threats
Tab.	Tabelle
TCP	Transmission Control Protocol
TKG	Telekommunikationsgesetz
TLD	Top-Level-Domain
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USA	United States of America
usw.	und so weiter
UTF	Unicode Transformation Format Resource Locator
vgl.	vergleiche
VML	Vector Markup Language
vs.	versus
WCAG	Web Content Accessibility Guidelines
WHATWG	Web Hypertext Application Technology Working Group
W3C	World Wide Web Consortium
WWW	World Wide Web
XAMPP	Cross-Plattform, Apache, MySQL, PHP, Perl
XML	Extensible Markup Language
XSS	Cross-Site Scripting
z.B.	zum Beispiel

Anhang A

„Index“ HTML Seitenaufbau (index.html)

Im folgenden Anhang wird der Seitenaufbau der HTML-Datei „index.html“ in Codeform dargestellt, welcher zu Beginn wichtige Header-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 2 „Webinterfaces anhand aktueller Standards“ als Musterbeispiel.

```
1  <!DOCTYPE html>
2  <!--
3      index.html v4.2.0 (for Thesis Project)
4      Copyright (C) 2016 Simon Marik <simon.marik@me.com>
5      Licensed under the GNU General Public License v3.0
6  -->
7  <html>
8      <head>
9          <meta http-equiv="content-type" content="text/html; charset=utf-8">
10         <meta name="author" content="Simon Marik">
11         <meta name="viewport" content="width=device-width, initial-scale=1.0,
12             user-scalable=no">
13         <title>Thesis Projekt</title>
14         <link rel="stylesheet" type="text/css" href="layout.css">
15     </head>
16     <body>
17         <nav>
18             <ul>
19                 <li><a href="index.html">Thesis Projekt</a></li>
20                 <li><a href="about.html">Über</a></li>
21             </ul>
22         </nav>
23         <main>
24             <div class="container">
25                 <div class="row">
26                     <div class="col-12">
27                         <div class="jumbotron">
28                             <h1>Hallo miteinander!</h1>
29                             <p>Lorem ipsum</p>
30                             <p>Lorem ipsum</p>
31                         </div>
32                     </div>
33                 </div>
34             </div>
35         </main>
36     </body>
37 </html>
```

```

34         <div class="col-6">
35             <div class="well">
36                 <h2>Quicklinks</h2>
37                 <hr>
38                 <ul>
39                     <li><p><a href="about.html">Über</a></p></li>
40                     <li><p><a
41                         href="mailto:simon.marik@me.com">Kontakt</a></p></li>
42                 </ul>
43             </div>
44         </div>
45         <div class="col-6">
46             <div class="well">
47                 <h2>Login</h2>
48                 <hr>
49                 <p>Lorem ipsum</p>
50             </div>
51         </div>
52     </div>
53 </main>
54 <footer>
55     <p>Thesis Projekt v4.2.0 &copy;2016 <a
56         href="mailto:simon.marik@me.com">Simon Marik</a></p>
57 </footer>
58 </body>
</html>

```

Anhang B

„Layout“ Stylesheet (layout.css)

Im folgenden Anhang wird der Cascading Stylesheet der Datei „layout.css“ in Codeform dargestellt, welcher zu Beginn wichtige Header-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 2 „Webinterfaces anhand aktueller Standards“ als Musterbeispiel.

```
1  /*!  
2   * layout.css v4.2.0 (for Thesis Project)  
3   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>  
4   * Licensed under the GNU General Public License v3.0  
5   */  
6  
7  @import url("http://fonts.googleapis.com/css?family=Oxygen:300,400,700");  
8  
9  html {  
10     font-family: "Oxygen";  
11 }  
12  
13 * {  
14     box-sizing: border-box;  
15 }  
16  
17 body {  
18     margin: 0;  
19 }  
20  
21 h2, h3, h4 {  
22     font-weight: 400;  
23 }  
24  
25 p {  
26     font-weight: 300;  
27 }  
28  
29 main {  
30     margin: 4em 1em 2em 1em;  
31 }  
32  
33 .container {  
34     margin: 0 15% 0 15%;
```

```
35 }
36
37 nav {
38     position: fixed;
39     top: 0;
40     left: 0;
41     z-index: 9999;
42     width: 100%;
43     height: 3em;
44 }
45
46 nav ul {
47     list-style-type: none;
48     margin: 0;
49     padding: 0;
50     overflow: hidden;
51     background-color: #fff;
52     border-bottom: 1px #ddd solid;
53 }
54
55 ul a {
56     text-decoration: none;
57 }
58
59 nav li {
60     float: left;
61 }
62
63 nav li a {
64     display: block;
65     color: #333;
66     font-weight: 300;
67     text-align: center;
68     padding: 14px 16px;
69     text-decoration: none;
70 }
71
72 nav li a:hover {
73     background-color: #eee;
74     color: #000;
75     transition: 0.3s;
76 }
77
78 .nav-active {
79     background-color: #eee;
80     color: #000;
81     cursor: dflt;
82     box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);
83 }
84
85 .nav-active a {
86     color: #55c801;
87 }
88
89 .nav-active a:hover {
90     box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);
```

```
91     transition: 0s;
92 }
93
94 footer {
95     position: fixed;
96     bottom: 0;
97     width: 100%;
98     height: 2.5em;
99     border-top: 1px #ddd solid;
100     font-weight: 300;
101     font-size: 10pt;
102     z-index: 9999;
103     background-color: #fff;
104 }
105
106 footer p {
107     margin: 0.8em 1em 0 1em;
108     font-size: 9pt;
109     color: #333;
110 }
111
112 footer p a {
113     text-decoration: none;
114 }
115
116 a {
117     text-decoration: underline;
118     color: #333;
119 }
120
121 a:hover {
122     color: #55c801;
123     transition: 0.3s;
124 }
125
126 .row::after {
127     content: "";
128     clear: both;
129     display: block;
130 }
131
132 .col-1 {
133     width: 8.33%;
134 }
135 .col-2 {
136     width: 16.66%;
137 }
138 .col-3 {
139     width: 25%;
140 }
141 .col-4 {
142     width: 33.33%;
143 }
144 .col-5 {
145     width: 41.66%;
146 }
```

```
147 .col-6 {
148     width: 50%;
149 }
150 .col-7 {
151     width: 58.33%;
152 }
153 .col-8 {
154     width: 66.66%;
155 }
156 .col-9 {
157     width: 75%;
158 }
159 .col-10 {
160     width: 83.33%;
161 }
162 .col-11 {
163     width: 91.66%;
164 }
165 .col-12 {
166     width: 100%;
167 }
168
169 [class*="col-"] {
170     float: left;
171     padding: 0 0.8em 0 0.8em;
172     font-weight: 300;
173 }
174
175 .well {
176     background-color: #fafafa;
177     border: 1px #ccc solid;
178     padding: 0 0.9em 0.1em 0.9em;
179     border-radius: 0.3em;
180     margin: 0 0 1em 0;
181 }
182
183 hr {
184     border: 0;
185     height: 0;
186     border-top: 1px solid rgba(0, 0, 0, 0.1);
187     border-bottom: 1px solid rgba(255, 255, 255, 0.3);
188 }
189
190 .jumbotron {
191     background-image: url("jumbotron-bg.png");
192     background-size: 100% 100%;
193     background-color: #eee;
194     height: 20em;
195     width: 100%;
196     padding: 1.7em 2.5em 2em 2.5em;
197     margin-bottom: 3em;
198     border-radius: 0.3em;
199 }
200
201 .jumbotron h1 {
202     margin-top: 0;
```

```
203     font-size: 38pt;
204 }
205
206 .jumbotron p {
207     font-size: 17pt;
208 }
209
210 /**
211  * MEDIA QUERIES
212  *
213  * 1. Tablet portrait and landscape orientation
214  * 2. Smartphone portrait and landscape orientation
215  * 3. Smaller devices
216  */
217
218 @media screen and (min-width: 600px) and (max-width: 1024px) {
219
220     .container {
221         margin: 0 10% 0 10%;
222     }
223
224     .jumbotron {
225         background-size: 150% 100%;
226     }
227
228     .jumbotron h1 {
229         font-size: 31pt;
230     }
231     .jumbotron p {
232         font-size: 15pt;
233     }
234
235     [class*="col-"] {
236         width: 100%;
237         padding: 0;
238     }
239
240 }
241
242 @media screen and (min-width: 320px) and (max-width: 600px) {
243
244     .container {
245         margin: 0 2% 0 2%;
246     }
247
248     .jumbotron {
249         background-size: 200% 100%;
250         padding: 1em 1em 1em 1em;
251     }
252
253     .jumbotron h1 {
254         font-size: 27pt;
255     }
256     .jumbotron p {
257         font-size: 14pt;
258     }
```

```
259
260     [class*="col-"] {
261         width: 100%;
262         padding: 0;
263     }
264
265     footer p {
266         margin: 0.65em 1em 0 1em;
267     }
268
269     .quicklinks a {
270         font-size: 11pt;
271     }
272
273     th {
274         font-size: 10pt;
275     }
276
277 }
278
279 @media screen and (max-width: 320px) {
280
281     .container {
282         margin: 0 2% 0 2%;
283     }
284
285     .jumbotron {
286         background-size: 200% 100%;
287     }
288
289     .jumbotron h1 {
290         font-size: 23pt;
291     }
292     .jumbotron p {
293         font-size: 12pt;
294     }
295
296     [class*="col-"] {
297         width: 100%;
298         padding: 0;
299     }
300
301     footer p {
302         margin: 1em 1em 0 1em;
303         font-size: 7pt;
304     }
305
306     .quicklinks a {
307         font-size: 11pt;
308     }
309
310     th {
311         font-size: 10pt;
312     }
313
314 }
```

Anhang C

„Index“ PHP Seitenaufbau (index.php)

Im folgenden Anhang wird der Seitenaufbau der PHP-Dateien „header.php“, „index.php“ sowie „footer.php“ der Reihe nach in Codeform dargestellt, welche zu Beginn jeweils wichtige Header- und ApiGen-Kommentare aufweisen, die genauestens beachtet werden sollten. Diese Codeauszüge dienen im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 3 „Der Umstieg von HTML auf PHP“ als Musterbeispiele.

Datei: header.php

```
1  <?php
2
3  /*!
4   * header.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  /**
10   * HEADER FILE
11   * @param void
12   * @var string $out
13   * @return $out
14   */
15
16  $out = '<!DOCTYPE html>
17      <!--
18          Thesis Project v4.2.0
19          Copyright (C) 2016 Simon Marik <simon.marik@me.com>
20          Licensed under the GNU General Public License v3.0
21      -->
22      <html lang="de">
23          <head>
24              <meta http-equiv="content-type" content="text/html;
25                  charset=utf-8">
26              <meta name="author" content="Simon Marik">
27              <meta name="viewport" content="width=device-width,
28                  initial-scale=1.0, user-scalable=no">
29              <title>Thesis Projekt</title>
30              <link rel="stylesheet" type="text/css" href="layout.css">
```

```

29         </head>
30         <body>
31             <nav>
32                 <ul>
33                     <li><a href="index.html">Thesis Projekt</a></li>
34                     <li><a href="about.html">Über</a></li>
35                 </ul>
36             </nav>';
37     echo $out;
38 ?>

```

Datei: index.php

```

1  <?php
2
3  /*!
4   * index.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  /**
10   * INDEX FILE
11   * @param void
12   * @uses function require_once()
13   * @var string $title
14   * @var string $out
15   * @return $out
16   */
17
18 session_start();
19 $title = 'INDEX';
20
21 define('ROOT', './');
22 require_once(ROOT.'core/header.php');
23 $out = '<main>
24     <div class="container">
25         <div class="row">
26             <div class="col-12">
27                 <div class="jumbotron">
28                     <h1>Hallo miteinander!</h1>
29                     <p>Lorem ipsum</p>
30                     <p>Lorem ipsum</p>
31                 </div>
32             </div>
33         </div>
34         <div class="row">
35             <div class="col-6">
36                 <div class="well">
37                     <h2>Quicklinks</h2>
38                     <hr>
39                     <ul>
40                         <li><p><a href="about.html">Über</a></p></li>
41                         <li><p><a

```

```

42         </ul>
43     </div>
44 </div>
45 <div class="col-6">
46     <div class="well">
47         <h2>Login</h2>
48         <hr>
49         <form class="form-signin"
            action="'.ROOT.'core/scr/scr.signin.php"
            method="post" autocomplete="off">
50             <input type="text" name="inputUsername"
                id="inputUsername"><br>
51             <input type="password" name="inputPassword"
                id="inputPassword">
52             <input type="checkbox" name="remember"
                value="remember-me"> Merken<br>
53             <button class="btn" id="submit"
                type="submit">Anmelden</button>
54             <button class="btn" id="reset"
                type="reset">Reset</button>
55         </form>
56     </div>
57 </div>
58 </div>
59 </div>
60 </main>';
61 echo $out;
62 require_once(ROOT.'core/footer.php');
63 ?>

```

Datei: footer.php

```

1 <?php
2
3 /*!
4  * footer.php v4.2.0 (for Thesis Project)
5  * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6  * Licensed under the GNU General Public License v3.0
7  */
8
9 /**
10  * FOOTER FILE
11  * @param void
12  * @var string $out
13  * @return $out
14  */
15
16 $out = '<footer>
17     <p>Thesis Projekt v4.2.0 &copy;2016 <a
        href="mailto:simon.marik@me.com">Simon Marik</a></p>
18     </footer>
19 </body>
20 </html>';
21 echo $out;
22 ?>

```


Anhang D

„Login“ Seitenaufbau (dashbd.php)

Im folgenden Anhang wird der Seitenaufbau der PHP-Datei „dashbd.html“ in Codeform dargestellt, welcher zu Beginn wichtige Header- und ApiGen-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 3 „Der Umstieg von HTML auf PHP“ als Musterbeispiel.

```
1  <?php
2
3  /*!
4   * dashbd.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  /**
10   * DASHBOARD FILE
11   * @param void
12   * @uses function require_once()
13   * @uses function session_start()
14   * @var string $out
15   * @return $out
16   */
17
18  session_start();
19  $title = 'DASHBOARD';
20
21  define('ROOT', '../..');
22  require_once(ROOT.'core/scr/scr.auth.php');
23  require_once(ROOT.'core/header.php');
24  $out = '<main>
25      <div class="container">
26          <div class="row">
27              <div class="col-12">
28                  <h1 class="h1-dashdb">Dashboard - "'. $username. '"</h1>
29              </div>
30          </div>
31          <div class="row">
32              <div class="col-12">
33                  <div class="well">
```

```
34         <h3>Willkommen im Dashboard</h3>
35         <hr>
36         <p>Sie sind nun angemeldet!</p>
37     </div>
38 </div>
39 </div>
40 </main>';
41 echo $out;
42 require_once(ROOT.'core/footer.php');
43 ?>
```

Anhang E

„Users“ Datenbank (thesis.sql)

Im folgenden Anhang wird die Struktur der Datenbanktabelle „thesis.users“ in Codeform dargestellt, welche zu Beginn wichtige Header-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 4 „PHP Anmeldeskripts mit Base64 kodierten Cookies“ als Musterbeispiel.

```
1  /*!  
2   * thesis.sql v4.2.0 (for Thesis Project)  
3   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>  
4   * Licensed under the GNU General Public License v3.0  
5   */  
6  
7  CREATE TABLE 'thesis_users' (  
8      'id' int(5) unsigned NOT NULL,  
9      'username' varchar(500) NOT NULL,  
10     'password' varchar(500) NOT NULL,  
11     'groups' varchar(100) NOT NULL  
12 )  
13 ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;  
14 ALTER TABLE 'thesis_users' ADD PRIMARY KEY ('id');
```

Die Datenbanktabelle „thesis.users“ muss durch Lehrende für das Unterrichtsszenario 4 vorab erstellt und zur Verfügung gestellt werden. Innerhalb von phpMyAdmin ergibt sich dabei die folgende Tabellenstruktur (Tab. E.0.1):

id:	username:	password:	groups:
1	admin	\$2y\$10\$SeR8vJ7wUPbHE4WFnbGB2eCCMYnrzn0BNIWAzUmuyeVkYe6SmiT.a	admins
2	test-user01	\$2y\$10\$S1BlvNUYXVz.8YbPsZ.FruHX/NjLx3QEUVy3.ayezyvuRibdvWY.y	users
3	test-user02	\$2y\$10\$p4t9YuL8dAB5DqeUpiB2.w7MHAKcqLIQaqlQMYmps2vWcsENh7m	users
4	test-user03	\$2y\$10\$JTinPbCdp.V6XCj1BrUXi.2wHUMKCBvzZfvrDp85sfzGxIx2RAILa	users
5	test-admin01	\$2y\$10\$VZIPN.UU77us04IXWzuAx.CCIGbMWJNIVWLZTVvU/Q4EfaxnrRjrm	admins
6	test-admin02	\$2y\$10\$etvNT6iFhshfkg9JhRMnROUvB4huxXcv6/Ql6zpy67PS0TlhJVJYy	admins

Tabelle E.0.1: Auszug aus der Datenbanktabelle „thesis.users“

Anhang F

„Connection“ Skript (scr.connect.php)

Im folgenden Anhang wird die Funktionsweise des PHP-Skripts „scr.connect.php“ in Codeform dargestellt, welche zu Beginn wichtige Header- und ApiGen-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 4 „PHP Anmeldeskripts mit Base64 kodierten Cookies“ als Musterbeispiel.

```
1  <?php
2
3  /*!
4   * scr.connect.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  /**
10   * CONNECT TO DATABASE
11   * @param void
12   * @uses function mysqli_connect_errno()
13   * @uses function mysqli_connect_error()
14   * @var string $db_host
15   * @var string $db_username
16   * @var string $db_password
17   * @var string $db_name
18   * @return void
19   */
20
21  $db_host = 'localhost';
22  $db_username = 'root';
23  $db_password = 'root';
24  $db_name = 'thesis';
25
26  $dbconnect = new mysqli($db_host, $db_username, $db_password, $db_name);
27  mysqli_set_charset($dbconnect, 'utf8');
28
29  if (mysqli_connect_errno())
30  {
31      echo 'Failed to connect to MySQL: ' . mysqli_connect_error();
32  }
33  ?>
```


Anhang G

„Sign in“ Skript (scr.signin.php)

Im folgenden Anhang wird die Funktionsweise des PHP-Skripts „scr.signin.php“ in Codeform dargestellt, welche zu Beginn wichtige Header- und ApiGen-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 4 „PHP Anmeldeskripts mit Base64 kodierten Cookies“ als Musterbeispiel.

```
1  <?php
2
3  /*!
4   * scr.signin.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  session_start();
10 define('ROOT', '../..');
11 require_once(ROOT.'core/scr/scr.connect.php');
12
13 /**
14  * USER SIGN IN
15  * @param string $username
16  * @param string $password
17  * @param string $remember
18  * @uses _DB usercheck
19  * @uses _DB dbact
20  * @uses _SESSION thesis-session
21  * @uses _SESSION user
22  * @uses _SESSION group
23  * @uses _SESSION start
24  * @uses _SESSION expire
25  * @uses _COOKIE thesis-cookie
26  * @uses _COOKIE thesis-token
27  * @var int $id
28  * @var string $group
29  * @var string $time
30  * @var string $date
31  * @var string $activityen
32  * @var string $activityde
33  * @var string $cookiedata
```

```

34  * @var      string      $cookietoken
35  * @var      string      $cookiehash
36  * @return void
37  */
38
39  $username = $_POST['inputUsername'];
40  $password = sha1($_POST['inputPassword']);
41  $remember = $_POST['remember'];
42
43  $usercheck = 'SELECT * FROM thesis_users WHERE username=\'\'.' . $username . '\';';
44  $dbconnect->real_escape_string($usercheck);
45  $query = mysqli_query($dbconnect, $usercheck);
46  $fetch = mysqli_fetch_object($query);
47
48  $group = $fetch->groups;
49  $id = $fetch->id;
50
51  if($fetch->username == $username && $fetch->password != '')
52  {
53      if($fetch->username == $username && password_verify($password,
54          $fetch->password))
55      {
56          $_SESSION['thesis-session'] = true;
57          $_SESSION['user'] = $id;
58          $_SESSION['group'] = $group;
59          $time = date('H:i:s');
60          $date = date('Y-m-d');
61
62          if($remember == 'remember-me')
63          {
64              $cookiedata = base64_encode('\'\'.' . $id . '\'-\'.' . $group . '\');
65              $cookietoken = sha1('\'\'.' . $id . '\'-\'.' . $group . '\'.substr($fetch->password, 7,
66                  5) . '\'-\'.' . substr($fetch->username, 0, 5) . '\');
67              $cookiehash = password_hash($cookietoken, PASSWORD_BCRYPT);
68              setcookie('thesis-cookie', $cookiedata, time() + (86400 * 5000), "/");
69              setcookie('thesis-token', $cookiehash, time() + (86400 * 5000), "/");
70          }
71          else
72          {
73              $_SESSION['start'] = time();
74              $_SESSION['expire'] = $_SESSION['start'] + (30 * 60);
75          }
76          header('Location: \' . $ROOT . \'core/dashbd.php\');
77      }
78      else
79      {
80          echo '<script> alert("Permission denied! Username and/or password
81              invalid."); window.location.href=\'\' . $ROOT . \'core/index.php\';</script>';
82      }
83  }
84  else
85  {
86      echo '<script> alert("Permission denied! User does not exist.");
87          window.location.href=\'\' . $ROOT . \'core/index.php\';</script>';
88  }
89  }
90  }
91  }
92  }
93  }
94  }
95  }
96  }
97  }
98  }
99  }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Anhang H

„Authentication“ Skript (scr.auth.php)

Im folgenden Anhang wird die Funktionsweise des PHP-Skripts „scr.auth.php“ in Codeform dargestellt, welche zu Beginn wichtige Header- und ApiGen-Kommentare aufweist, die genauestens beachtet werden sollten. Dieser Codeauszug dient im Zuge des Unterrichtskonzeptes für das Unterrichtsszenario 5 „Authentifizierung anhand des Salted-Hash-Verfahrens“ als Musterbeispiel.

```
1  <?php
2
3  /*!
4   * scr.auth.php v4.2.0 (for Thesis Project)
5   * Copyright (C) 2016 Simon Marik <simon.marik@me.com>
6   * Licensed under the GNU General Public License v3.0
7   */
8
9  session_start();
10 define('ROOT', '..../..');
11 require_once(ROOT.'core/scr/scr.connect.php');
12
13 /**
14  * USER AUTHENTICATION
15  * @param int $idnow
16  * @uses _SESSION thesis-session
17  * @uses _SESSION group
18  * @uses _SESSION start
19  * @uses _SESSION expire
20  * @uses _COOKIE thesis-cookie
21  * @uses _COOKIE thesis-token
22  * @uses _DB usercheck
23  * @uses _DB dbact
24  * @var arr $pieces
25  * @var string $cookiedec
26  * @var string $time
27  * @var string $date
28  * @var string $username
29  * @var string $group
30  * @var string $now
31  * @var string $activityen
32  * @var string $activityde
33  * @var string $cookietoken
```

```

34  * @return void
35  */
36
37  $idnow = $_SESSION['user'];
38  $time = date('H:i:s');
39  $date = date('Y-m-d');
40
41  if(!isset($_COOKIE['thesis-cookie']))
42  {
43      if (!isset($_SESSION['thesis-session']) || !$_SESSION['thesis-session'])
44      {
45          echo '<script> alert("Permission denied! Please sign in first.");
46              window.location.href="'.ROOT.'core/index.php"</script>';
47      }
48      else
49      {
50          $usercheck = 'SELECT username, groups FROM thesis_users WHERE
51              id=\'\'.'.$idnow.'\'';
52          $dbconnect->real_escape_string($usercheck);
53          $query = mysqli_query($dbconnect, $usercheck);
54          $fetch = mysqli_fetch_object($query);
55
56          $username = $fetch->username;
57          $group = $fetch->groups;
58          $now = time();
59
60          if ($now > $_SESSION['expire'])
61          {
62              session_unset();
63              session_destroy();
64              echo '<script> alert("Session has expired! Please sign in again.");
65                  window.location.href="'.ROOT.'core/index.php"</script>';
66          }
67          else
68          {
69              $_SESSION['group'] = $group;
70              $_SESSION['start'] = time();
71              $_SESSION['expire'] = $_SESSION['start'] + (30 * 60);
72          }
73      }
74  }
75
76  if(isset($_COOKIE['thesis-cookie']))
77  {
78      $cookiedec = base64_decode($_COOKIE["thesis-cookie"]);
79      $pieces = explode('-', $cookiedec);
80      $id = $pieces[0];
81      $group = $pieces[1];
82
83      $usercheck = 'SELECT username, password, groups FROM thesis_users WHERE
84          id=\'\'.'.$id.'\'';
85      $dbconnect->real_escape_string($usercheck);
86      $query = mysqli_query($dbconnect, $usercheck);
87      $fetch = mysqli_fetch_object($query);
88
89      $cookietoken = sha1('\'\'.'.$id.'-\'\'.'.$group.'-\'\'.'.substr($fetch->password, 7,
90          5).'-\'\'.'.substr($fetch->username, 0, 5).'\'');

```

```
85     $username = $fetch->username;
86
87     if(password_verify($cookietoken, $_COOKIE['thesis-token']))
88     {
89         $_SESSION['thesis-session'] = true;
90         $_SESSION['user'] = $id;
91         $_SESSION['group'] = $group;
92     }
93     else
94     {
95         $_SESSION['thesis-session'] = false;
96         setcookie('thesis-cookie', '', time() - (86400 * 5000), "/");
97         setcookie('thesis-token', '', time() - (86400 * 5000), "/");
98         session_unset();
99         session_destroy();
100         echo '<script> alert("Userdata changed! Please sign in again.");
            window.location.href="'.ROOT.'core/index.php"</script>';
101     }
102 }
103 ?>
```