

IT-Sicherheit in Webprojekten

Diplomarbeit

Alph Raue
30.09.2015

IT-Sicherheit in Webprojekten

Inhalt

1. Einführung	4
2. Grundlegende Möglichkeiten und Verfahren zur Absicherung privater Informationswebseiten und kleinerer Foren.....	7
2.1. Möglichkeiten des Webhosters	7
2.2. Formulare	9
2.3. Datenbanken.....	10
2.4. Verschlüsselungen	12
2.5. Software	16
3. Zusammenfassung.....	22
4. Verzeichnisse	23
4.1. Literaturverzeichnis.....	23
4.2. Abbildungsverzeichnis	24
5. Eigenständigkeitserklärung	25
Statistik	26
Arbeitstabelle	26

1. Einführung

Während meines Studiums über Webdesign & -development wurde immer wieder auf das Thema der Sicherheit bei Webanwendungen hingewiesen, diese Thematik aber, außer im Bereich der Formularerstellung, nicht weiter vertieft. Ein Grund, sich dieser Problematik im Rahmen einer Facharbeit zu nähern und das Thema kurz zu beleuchten.

Es stellt sich zunächst einmal die Frage, warum IT-Sicherheit¹ auch auf privaten Informationswebseiten (Webvisitenkarten) und kleineren Foren (z.B. in speziellen Nischenthemen), mit denen ich mich in dieser Arbeit bevorzugt befassen werde, notwendig ist?

Da mit dem Begriff IT-Sicherheit ein großes Gebiet, mit zum Teil unterschiedlichen Begriffen, in Zusammenhang gebracht wird, muss erst einmal kurz umrissen werden, um was es sich dabei handelt.

Ein IT-System ist „...ein geschlossenes oder offenes, dynamisches technisches System mit der Fähigkeit zur Speicherung und Verarbeitung von Informationen“ (Eckert, 2006, p. 2)

Diese Systeme stehen größtenteils nicht alleine. Sie sind in kleineren oder größeren Gruppen mit einander verbunden. Sei es in Heimnetzwerken, in einem Intranet oder (umfassend) im Internet. In dem Moment, wo ein Status erreicht wurde, bei dem mehrere IT-Systeme miteinander kommunizieren, kommen die Elemente und Maßnahmen IT-Sicherheit zum Tragen.

Es kommen drei verschiedene Formen von Sicherheitsformen zur Anwendung:

1. **Funktionssicherheit**
2. **Informationssicherheit**
3. **Datensicherheit**

Eine vierte Form ist der **Datenschutz** als die Kontrolle der Weitergabe und dem Schutz von persönlichen Daten. Auf dieses Thema wird später, im Rahmen der Behandlung des Themas Datenbanken, noch einmal eingegangen.

„Unter **Funktionssicherheit** (engl. *safety*) eines Systems verstehen wir die Eigenschaft, dass die realisierte Ist-Funktionalität der Komponenten mit der spezifizierten Soll-Funktionalität übereinstimmt.“ (Eckert, 2006, p. 4)

„Die **Informationssicherheit** (engl. *security*) ist die Eigenschaft eines funktionssicheren Systems, nur solche Systemzustände anzunehmen, unautorisierten Informationsveränderung oder – gewinnung führen.“ (Eckert, 2006, p. 5)

„Die **Datensicherheit** (engl. *protection*) ist die Eigenschaft eines funktionssicheren Systems, nur solche Systemzustände anzunehmen, die zu keinem unautorisierten Zugriff auf Systemressourcen und insbesondere auf Daten führen.“ (Eckert, 2006, p. 5)

Die Wahrung dieser Sicherheitsformen innerhalb eines IT-Systems wird gemeinhin als IT-Sicherheit bezeichnet.

¹ Informationstechnisches System

Hauptaugenmerk des uns hier interessierenden Gebietes der IT-Sicherheit (oder auch Websicherheit) liegt auf der Feststellung von Problemen bei der Verwendung von Computersystemen, die durch den unbefugten Zugriff von außen entstehen können. Ziel ist es, diese Zugriffe zu verhindern oder weitestmöglich einzuschränken. Der Begriff des weitestmöglichen Einschränkens wurde hier ganz bewusst gewählt, da eine hundertprozentige Sicherheit nie gewährt werden kann. Laut einer Umfrage der Wirtschaftsprüfungsgesellschaft *Pricewaterhouse Coopers* (PwC) aus dem Jahr 2014² wurde festgestellt:

„Das zentrale Ergebnis: Im Jahr 2013 ist die Gesamtzahl der Angriffe auf die IT- Sicherheit von Unternehmen im Vergleich zum Vorjahr um 48 Prozent auf 42,8 Millionen angestiegen. Dies entspricht 117.330 Angriffen pro Tag. Seit 2009 ist die Zahl damit sogar um 66 Prozent angestiegen.“ (www.pwc.de, 2014)

Und obwohl sich hier um eine steigende Tendenz handelt, wird weiter festgestellt:

„Trotz der zunehmenden Anzahl an Sicherheitsvorfällen sinken die Ausgaben für IT-Sicherheit.“ (www.pwc.de, 2014)

Das sind die Ergebnisse einer Umfrage im professionellen Anwendungsbereich von Unternehmen! Es zeigt sich, dass Websicherheit einen hohen Stellenwert bei der Konzeption von Webseiten einnehmen sollte. Wenn die Anzahl und die Intensität von Cyberangriffen im Bereich der Unternehmenskommunikation ansteigen, dann gilt das erst recht bei der Konzeption eher privater Webseiten oder Lösungen für Kleinunternehmen.

Sei es nun als Betreiber einer kleinen privaten Homepage mit reinem Informationscharakter, als Kleinunternehmer mit Webvisitenkarte oder als Betreiber eines Forums mit populärem Charakter (z.B. Gartenverein oder Sammler von Feuerwehrhelmen³). In jedem Fall müssen die Erkenntnisse im Bereich der IT-Sicherheit möglichst schon bei der Konzeption der Webseite, beim Gestalten von Formularen und Datenbanken , sowie einem eventuellen Verwaltens von Dateien mit einfließen oder aber später implementiert werden.

Der Grund dafür liegt in der Tatsache, dass genau solche Webkonzeptionen, vor allem wenn sie über eine Datenbanklösung verfügen, als Einfallstor für Attacken genutzt werden.

Das russische Softwareunternehmen *Kaspersky*, bekannt durch seine Antivirensoftware, ermittelte im Rahmen einer 2014 durchgeführten Umfrage (www.computerwoche.de, 2014), die Spitzenreiter der Cybergefahren:

1. **Spam.** Hierbei handelt es sich um ungewollt zugesendete Mails (meistens Werbung), deren Anhang (Bilder, PDF oder Textdateien) Schadprogramme enthalten kann.
2. **Malware.** Sammelbezeichnung für Programme, die in böswilliger Absicht geschrieben und verbreitet werden, meist ohne Einwilligung der Benutzer in Rechner oder Computersysteme eindringen und Irritationen, Störungen oder Schäden verursachen können. (Autorenteam, 2013, p. 561)

² Zum Zeitpunkt der Quellensuche für diese Arbeit lagen keine zuverlässigen und kompetenten aktuelleren Studien vor.

³ Ja, die gibt es: <http://www.derfeuerwehrhelm.de/>

3. **Phishing.** Bezeichnet das durch Malware ermöglichte Abgreifen von Nutzerdaten und deren späteren Missbrauch.

Durch Spam soll ein Nutzer dazu gebracht werden, Anhänge von E-Mail Nachrichten zu öffnen. Ist das gelungen, wird unter Umständen Malware installiert, welche wiederum genutzt werden kann, Phishingangriffe zu starten. Alle diese teilweise aufeinander aufbauenden Bedrohungen nutzen größtenteils unsichere PCs und Server von obengenannten Betreibern.

Obwohl diese Gefahren allgemein bekannt sind und obwohl auch die Wirkung von Cybergefahren fast täglich in den Medien publiziert und debattiert werden, wird die Thematik der IT-Sicherheit sowohl in Unternehmen, als auch in dem hier beleuchteten Sektor häufig vernachlässigt.

Grund dafür ist teilweise Bequemlichkeit. Aber auch Unkenntnis über Verbreitungswege und –möglichkeiten, sowie die Unterschätzung der daraus resultierenden Gefahren.

„Für viele Nutzer wird Internetsicherheit das nervende Popup-Fenster des Virenschanners und dieses doofe Windows-Update bleiben, das immer wieder ungefragt Arbeit oder Vergnügen blockiert.“ (www.sueddeutsche.de, 2014)

Tatsache ist aber: Wenn ein Programm in Form von Malware den Weg auf das System gefunden hat, dann besteht, ohne geeignete Gegenmaßnahmen, die Gefahr, dass auch weitere Systemkomponenten davon betroffen werden! Der Angreifer ist nun potentiell in der Lage, Passwörter auszulesen und somit Zugriff auf einen Webaccount, die dort liegenden Datenbanken und Dateien zu erhalten. Der betroffene Rechner wird nun selber zur Gefahr.

Was wird nun gegen diese Möglichkeiten von Angriffen getan? Was kann man als Betreiber einer privaten Homepage oder eines Kleinforums, möglichst schon im Vorfeld, tun, um Angreifern die Möglichkeiten zu nehmen Zugriff zu erhalten und somit die Verbreitung von Spam und Malware zu stoppen und die persönliche Daten vor Missbrauch zu schützen?

2. Grundlegende Möglichkeiten und Verfahren zur Absicherung privater Informationswebseiten und kleinerer Foren

2.1. Möglichkeiten des Webhosters

2.1.1. Webhoster

Den Webhostern, also den professionellen Anbietern von Webspace, sind die im vorherigen Abschnitt angesprochenen Problematiken bekannt und sie bieten von vornherein die richtigen Werkzeuge an, um dem Nutzer Schutzmaßnahmen für seine Projekte bieten zu können, welche bei Bedarf auch erweitert werden können. Die Erweiterungen erfolgen kostenpflichtig.

Der Webhoster kümmert sich um alle Belange, die den Betrieb der Hardware, der Software und die Speicherung der Daten betreffen. Es wird angestrebt, dass keinerlei Datenverlust durch ausgefallene Hardware entsteht. Auch die grundlegenden softwareseitigen Sicherheitsmaßnahmen sind bei allen Hostern fest konfiguriert und können in den meisten Einstellungen nicht modifiziert werden. Das mag ärgerlich erscheinen, verhindert aber auch eine Manipulation von eventuellen Angreifern. Unter gewissen Voraussetzungen könnte, z.B. bei falscher oder nachlässiger Konfiguration, der Benutzer sich selber schaden oder gar aussperren. Auch das wird durch diese Einschränkungen verhindert.

Zum Stand September 2015 gab es bei allen Hostern unter den getesteten oberen Zehn⁴ (www.hosting-review.com, 2015) folgende vorkonfigurierten Sicherheitsmaßnahmen:

- **Vorkonfigurierte PHP.ini:** Mit der PHP.ini liegt dem Anwender eine Konfigurationsdatei vor, mit der er das Laufzeitverhalten seiner PHP Anwendungen bestimmen kann. Dazu gehören Pfadfestlegungen, Einstellungen, welche die maximale Laufzeit von Scripten festlegen und die Behandlung von Laufzeitfehlern.
- **SiteGuard⁵:** Mit *SideGuard* liefern die Hoster eine Möglichkeit, Schreibzugriffe auf den Webspace zu protokollieren. Es kann explizit ausgewählt werden, welche Ordner überwacht werden sollen. In der Basiseinstellung werden alle Ordner überwacht, was aber unter Umständen nicht sinnhaltig sein kann. Z.B. beim Betrieb eines Kleinforums, eines Wordpress Blogs oder eines Typo CMS⁶. Dort kommt es, durch Einträge von Seiten der Besuche, durch Einkäufe und Kommentare, häufiger zu Schreibzugriffen. Diese Ordner sollten von der Überwachung ausgeschlossen werden. Über Zugriffe wird der Betreiber in wählbaren Intervallen per E-Mail informiert.
- **ServerSide AntiVirus⁵:** Dabei handelt es sich um eine serverseitige Lösung zum Schutz vor Viren in Nachrichtenordnern. Es soll sichergestellt werden, dass evtl. eingeschleuste Viren nicht weiter verbreitet werden können, in dem sie schon vor der Weiterleitung von Nachrichten in Quarantäne verschoben oder sofort gelöscht werden. Die verwendete Suchheuristik wird vom Webhoster festgelegt. Es besteht keine Möglichkeit

⁴ Dazu gehörten auch die bekannten deutschen Hoster all-incl.com (Platz 10), 1&1 (Platz 3) sowie STRATO (Platz 2). Da der Verfasser dieser Arbeit schon bei diesen Hostern über Space verfügte, werden diese 3 als Referenz herangezogen. Es wird davon ausgegangen, dass die anderen Hoster über gleiche bzw. ähnlich geartete und auch ähnlich konfigurierbare Angebote verfügen.

⁵ Die folgenden Angebote werden beim Webhoster STRATO so bezeichnet. Unter anderem Anbieter kann auch der Name abweichen!

⁶ CMS: Content Management System.

der Modifikation oder der Überwachung des Aktualisierungsstandes. Es ist zu empfehlen, dass diesbezüglich immer für eine weitere clientseitige AntiVirus Lösung gesorgt wird!

- **ServerSide AntiSpam⁵**: Hierbei handelt es sich um einen serverseitig arbeitenden Spamfilter. Dieser ist ausschließlich auf die zum Webspace gehörigen E-Mail Postfächer beschränkt. Die Filter werden, wie bei *ServerSide AntiVirus*, vom Webhoster gepflegt und konfiguriert. Webhoster Strato erklärt die vom Unternehmen genutzte Filtertechnologie folgendermaßen: „*Um Spam- und Phishing-E-Mails zu erkennen, analysiert der Filter den Inhalt aller E-Mails. Dabei berücksichtigt er alle Bedeutungseinheiten, die sich aus mehreren (bis zu fünf) Wörtern ergeben, die unmittelbar aufeinander folgen oder nahe beieinander stehen. Anhand der enthaltenen Kombination von Formulierungen werde Spam-E-Mails zuverlässig erkannt, auch wenn sie zum ersten Mal verschickt werden. Der Filter wird auf einer Basis von mehreren Millionen Spam-E-Mails regelmäßig neu trainiert. Der Filter berücksichtigt, ob Sender und Empfänger einer Nachricht sich kennen und gegenseitig E-Mails schicken. Das Risiko, dass der Filter eine wichtige Nachricht irrtümlich als Spam markiert, wird so auf nahezu null reduziert.*“ (www.strato.de, 2015)⁷

Es muss klargestellt werden, dass die angebotenen serverseitigen Lösungen für gängige Sicherheitsprobleme nicht ausreichend sind, um Schaden und/oder Angriffe vollständig abzuwehren. Um höchstmöglichen Schutz zu gewährleisten, ist immer ein hoher Eigenanteil an Sicherheitslösungen zu geben.

2.1.2. Freehoster

Freehoster bieten grundsätzlich erst einmal kostenlosen Webspace an. In manchen Fällen mag dass eine gute Idee sein, doch sollte man sich, bevor man sich auf den Betrieb einer Webseite oder eines Forums über einen Freehoster einlässt, über gewisse Punkte Gedanken machen!

Freehoster sind zumeist Betreiber kleiner Serverfarmen, die sich zum größten Teil durch Werbeeinnahmen finanzieren. Da es dabei um eine recht unzuverlässige Einnahmequelle handelt und die entsprechenden Mittel fehlen, kann es vorkommen, dass die verwendete Hardware nicht mehr dem neusten Stand entspricht. Bevor man also Daten von Usern auf solchen Servern speichert, sollte man sich bewusst sein, dass ein kompletter Datenverlust möglich ist. Der kann auch eintreten, wenn der Freehoster so in die finanzielle Schieflage gerät, dass er den Betrieb einstellen muss. Auch in diesem Fall sind die Daten unwiederbringlich verloren.⁸

Abgesehen davon, dass es nicht schön ist, wenn die eigene Webseite nur so von Werbung strotzt (die Finanzierung läuft nun mal über diese Konzept), weiß man auch nicht, was die Werbung ansonsten noch so mit sich bringt. Im Zeitalter von Supercookies und Tracking⁹ ist das ein nicht zu vernachlässigender Punkt! Schnell kommt der Punkt, wo dann im kleinen, mühsam aufgebauten Forum die User wegbleiben.

⁷ Strato geht auf die verwendeten Technologien und Vorgehensweisen nur auf den Hilfeseiten angemeldeter Nutzer ein. Auf der offiziellen Seite (<https://www.strato.de/hosting/#sicherheit>) bekommt man nur einen allgemeinen Einblick.

⁸ Siehe auch im Abschnitt Zusammenfassung

⁹ Webseitenverfolgung zum Zweck der Erfassung des Nutzerverhaltens

2.2. Formulare

Schlecht durchdachte und nachlässig programmierte Formulare stellen und stellen immer ein beliebtes Einfallstor für manipulative Zugriffe, Datendiebstahl und Einbrüche in Datenbanken dar.

Das was ein Formular an Funktionalität beinhaltet, ist gleichzeitig der Grund, warum es, im schlechtesten Fall, eine Gefahr darstellt: es übergibt Variablen an eine Datenbank. Und diese Variablen lassen sich für den Transport von Fehlinformationen an diese Datenbank nutzen.

Eine derart manipulierte Datenbank stellt mannigfaltige Möglichkeiten bereit um Daten zu versenden (z.B. Spam), Daten abzufragen und ohne Wissen der Dateninhaber zu missbrauchen oder die Datenbank wird einfach genutzt, um illegal in Besitz genommene Daten „zwischenzulagern“⁸.

Wie wir im nächsten Abschnitt sehen werden, können wir eine Datenbank so absichern, dass ein direkter Zugriff erschwert wird. Mit einem Formular ist es jedoch möglich, in eine Datenbank einzudringen und dort Aktionen zu starten, ohne die erforderlichen Zugangsdaten zu besitzen. Diese Form des Angriffs nennt man *SQL-Injection*.

Um zu verstehen, worum es sich dabei handelt, muss man zuerst wissen, wie ein Formular Daten erhält, behandelt und an eine Datenbank überträgt.

Ein Formular besteht aus einem HTML Code in der Form (stark vereinfacht und sehr kurz):

Beispiel:

```
<form method="post" action="index.php">  
E-Mail: <input type="text" name="email" id="email"/><br>  
<input class="button" type="submit" name="submit" id="submit" value="Senden" />  
</form>
```

Erzeugt:

A screenshot of a web form. It features a text input field with the label 'E-Mail:' to its left. Below the input field is a button labeled 'Senden'. A mouse cursor is pointing at the 'Senden' button. The entire form is enclosed in a light gray border.

A. Raue © 2015

E-Mail Abfrage (Abbildung 1)

Erzeugt wird ein Eingabefeld vom Typ Text mit dem Namen „email“. Der Parameter „action“ sagt dem Formular, wo sich das verarbeitende Script befindet, welches beim Klick auf „Senden“ abgearbeitet werden soll. Die Versandmethode ist „post“.

In die Datenbank gelangt der im Feld „E-Mail“ eingetragene Wert nun über den Weg: Auslesen, Zuordnen, Datenbank aufrufen, Wert übergeben, Eintragen.

Nach diesem Prinzip sind auch die Abfragen aufgebaut, wenn sich ein registrierter User z.B. in seinem Forum anmelden will (Abbildung 2):

Das Bild zeigt ein Login-Formular mit einem roten gestrichelten Rahmen. Es besteht aus zwei Eingabefeldern: 'Benutzername' und 'Passwort'. Darunter befinden sich zwei Buttons: 'Anmelden' und 'Zurücksetzen'. Ein Mauszeiger ist über dem 'Zurücksetzen'-Button positioniert.

A. Raue © 2015

Abfragemaske (Abbildung 2)

Wenn ein Angreifer nun statt eines Benutzernamens oder Passwortes an dieser Stelle geschickt gewählte MySQL Statements eingibt und diese absendet, so kann es, wenn die PHP Abfragen des Formulars nachlässig programmiert wurden, zu der oben erwähnten SQL-Injection kommen. Dies kann im Extremfall dazu führen, dass Datensätze verfälscht, gelöscht oder sogar die komplette Datenbank zerstört wird!

2.3. Datenbanken

Da bei den, im Rahmen dieser Betrachtung behandelten Benutzergruppe, verwendeten Datenbanken MySQL eine herausragende Rolle spielt und diese Datenbank auch von allen einbezogenen Hostern vorwiegend angeboten wird, wird dieses Datenbanksystem als Referenz für die anderen gängigen Systeme¹⁰ genommen.

MySQL ist eine 1994 vorgestellter und ursprünglich von einem schwedischen Entwicklerteam (MySQL AB) programmierter, relationaler Datenbankserver¹¹. Momentan wird das System von *Oracle* betreut. Es handelt sich um eine OpenSource Software. MySQL ist Bestandteil der Entwicklungsumgebung XAMPP, auf die im Abschnitt Software noch einmal näher eingegangen werden soll.

Was zeichnet MySQL nun aus? MySQL ist relativ einfach zu erlernen. Mit nur geringen Kenntnissen in der Thematik relationale Datenbanken und einigen SQL Anweisungen (so genannten Statements) kann auch ein Einsteiger in kurzer Zeit eine Datenbank anlegen, mit Daten füllen und Datensätze abfragen.

Beispiel:

```
CREATE DATABASE automarken;
```

¹⁰ Z.B. Oracle Database, PostgreSQL, MongoDB

¹¹ Ein auf, in Verhältnis stehenden, Tabellen beruhendes System der Datensammlung.

```
CREATE TABLE marken (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
bezeichnung VARCHAR(25), ps (INTEGER(3), max_kmh (INTEGER(3));
```

In diesem Beispiel wird zuerst eine Datenbank namens „automarken“ erzeugt (kreiert). In dieser Datenbank wiederum wird eine Tabelle mit der Bezeichnung „marken“ erstellt, welche vier Spalten enthält: „id“, „bezeichnung“, „ps“ und „max_kmh“. Die *id* wird benötigt, um das Element in der Tabelle über seinen Index anzusprechen. Es ist vom Typ Ganzzahl (integer), darf nicht leer sein (not null), es ist der Primärschlüssel (index) und dieser Index wird bei jedem Tabelleneintrag automatisch hochgezählt (autoincrement).

Diese Art der Datenbank- bzw. Tabellenerstellung kann über verschiedene Wege erfolgen. Möglich wäre z.B. die Erstellung über die Kommandozeile (cmd (Windows und MacOS), bash (Linux). Über diesen Weg allerdings ohne grafische Oberfläche. Mit grafischer Oberfläche (GUI¹²) kann über PhpMyAdmin (einer PHP basierenden Browseranwendung) gearbeitet werden. Die wohl gebräuchlichste Form der Erstellung im Bereich der Webentwicklung stellt aber der Datenbankaufbau über PHP Kommandos dar. Über diese erfolgt auch der Aufruf der Datenbank.

Und hier setzt auch schon ein primäres Sicherheitskonzept von MySQL an: der Zugriff auf die Datenbank und damit auf ihren Inhalt, soll mit einem Benutzernamen und einem Passwort gesichert werden. Bei der Einrichtung von MySQL wird der Benutzer deshalb dazu aufgefordert, diese Daten anzulegen. Rein theoretisch wäre es möglich, die Datenbank auch ohne diese Angaben zu betreiben, aber genau das wäre es, was allen bereits besprochenen Sicherheitskonzepten (in diesem Fall der Datensicherheit) widersprechen würde.

Der Anwender kann durch das Setzen von Zugangsparametern, sowie die Art des Zugriffs, die Sicherheit einer Datenbank immens erhöhen. Er steuert schon im Vorfeld, wer wann und in welcher Form Zugriff auf welche Teile der Daten haben darf. Hier spielt das bereits in der Einführung erwähnte Thema des Datenschutzes eine Rolle. Als Betreiber einer Datenbank hat man die Pflicht für die Sicherheit der verwalteten Daten zu sorgen. Jeglicher unbefugter Zugriff muss ausgeschlossen werden können.

Um diese Konzept konsequent durchsetzen zu können, wurde dem Entwickler ein Werkzeug in die Hand gegeben, welches die Art des Zugriffs sicherer macht: der Zugriff über PDO¹³. PDO unterscheidet sich von den noch vor wenigen Jahren üblichen und auch heute noch (leider) angewendeten Zugriffsmethoden dadurch, dass der Zugriff über objektorientierte Module erfolgt.

Bei der prozeduralen Programmierung wird eine Verbindung in der im Beispiel aufgezeigten Form aufgebaut. Durch die offene Gestaltung kann ein Angreifer auf die sensiblen Daten „Servername“, „Benutzername“ und „Passwort“ Zugriff erlangen:

Beispiel:

```
$datenbank = mysql_connect(„servername“, „benutzername“, „passwort“);  
mysql_select_db(„automarken“);
```

Über PDO werden nun diese Daten in einem Objekt gekapselt. Dadurch ist ein Zugriff von außen nur sehr schwer bis unmöglich gemacht:

¹² Graphic User Interface: Grafische Benutzeroberfläche

¹³ PHP Data Objects

Beispiel:

```
try{
$datenbank = new PDO(„mysql:dbname=automarken; host=servername“,
„benutzername“, „passwort“);
} catch (PDOException $e) {
echo „Fehler: “ .htmlspecialchars($e->getMessage());
exit();
}
```

Durch eine durchdachte Gestaltung einer Datenbank in Verbindung mit modernen Zugriffsmethoden, ist es also möglich, ohne großen Aufwand für die erforderliche Sicherheit zu sorgen. Ein weitere Schritt, um die Sicherheitskonzepte durchzusetzen.

2.4. Verschlüsselungen

Das Verschlüsseln von Nachrichten gehört zu einer der ältesten Anliegen, um Daten vor dem Zugriff von Unbefugten zu schützen. Aus diesem Anliegen, bzw. der Notwendigkeit der Verschlüsselung, entwickelte sich früh¹⁴ eine wissenschaftliche Methode: die Kryptographie. „Unter Kryptographie versteht man die Lehre von den Methoden zur Ver- und Entschlüsselung von Nachrichten zum Zweck der Geheimhaltung von Informationen gegenüber Dritten (Angreifern).“ (Eckert, 2006, p. 281)

Die Kryptografie versuchte, meist mit mechanischen oder mathematischen Verfahren, eine höchst mögliche Sicherheit zu erlangen.

Nicht unerwähnt bleiben soll die Kryptoanalyse. „Die Kryptoanalyse ist die Wissenschaft von den Methoden zur Entschlüsselung von Nachrichten, ohne Zugriff auf den verwendeten Schlüssel zu haben.“ (Eckert, 2006, p. 281)

Kryptographie und Kryptoanalyse bilden zusammen das Gebiet der Kryptologie.

Als Erweiterung der Kryptologie können die Techniken der Steganografie betrachtet werden. Diese beschäftigen sich damit, eine Nachricht gar nicht erst sichtbar zu machen, anstatt sie zu verschlüsseln. „Unter Steganografie (griech. *stegano*: geheim, *graphein*: schreiben) fasst man Methoden zusammen, die darauf abzielen, bereits die Existenz einer Nachricht zu verbergen (engl. *conceal*) und nicht nur den Informationsgehalt einer Nachricht zu verschleiern.“ (Eckert, 2006, p. 281)

Bevor nun näher auf allgemeine Fragen und spezielle Verfahren eingegangen werden soll, muss klargestellt werden, dass es für die alle Teile der Kryptografie Standards gibt. Diese legen fest, welche Anforderungen eine Verschlüsselung erfüllen muss, um bestimmten Aufgaben gewachsen zu sein. Ebenso gilt zu Beachten, dass es Bemühungen gibt, kryptografische Technologien auf gewissen gebieten einzuschränken oder sogar zu verbieten. Natürlich gibt es auch gegenläufige Bemühungen. Die so genannte Kryptoregulierung gehört zwar nur indirekt zur hier behandelten Thematik, soll aber doch erwähnt werden.

¹⁴ Erste kryptographische Ansätze sind seit 3000 v. Chr. nachgewiesen

Es ergeben sich zwei Lager: Regulierungsbefürworter und Regulierungsgegner.

„Das Hauptaugenmerk der Regulierungsbefürworter betrifft die Notwendigkeit, dass staatliche Stellen zum Zweck der Verbrechensbekämpfung Zugriff auf die Klartexte verschlüsselter Daten erhalten müssen.“ (Eckert, 2006, p. 348)

Dem gegenüber steht aber:

„Hauptargumente der Regulierungsgegner betreffen zum einen die Unsicherheit heutige Netztechnologie, die den Einsatz kryptografischer Systeme als einen unverzichtbaren Bestandteil für den Betrieb vernetzter Rechner erforderlich macht.“ (Eckert, 2006, p. 348)

Diese Problematik hat auch juristische Auswirkungen. So steht in vielen Ländern oder Gemeinschaften seit Jahren ein Verschlüsselungsverbot zur Debatte. So äußerte eine EU Kommission im April 2015 Bedenken über die Nutzung von Verschlüsselungswerkzeugen. (www.netzpolitik.org, 2015) Da es sich dabei um Bedenken gegen die Verschlüsselung von E-Mails handelt, soll diese Problematik hier nicht weiter interessieren.

In Deutschland zumindest gibt es in Fragen der Verschlüsselung keine Regulierung, soweit es sich nicht um Exporte in Länder außerhalb der EU handelt. Die Anwendung der folgenden Methoden von Verschlüsselungen ist also unbedenklich.

Es gibt zahlreiche Verschlüsselungsverfahren mit unterschiedlichsten Methoden und Anwendungsgebieten. Zu den für diese Arbeit relevanten Verschlüsselungen gehören die sogenannten Hashfunktionen. (engl. *hash*: durcheinander), welche über mehr oder weniger komplexe mathematische Verfahren einen so genannten digitalen Fingerabdruck erstellen. Allen im Verlauf vorgestellten Hashfunktionen ist gemein, dass sie, egal wie kurz oder lang der Ausgangswert (z.B. ein Passwort) ist, immer einen Wert statischer Länge ausgeben. Aus dem Ausgabewert kann nur unter großen Aufwendungen wieder auf den Ausgangswert geschlussfolgert werden. Wichtig ist, dass ein gleicher Ausgangswert bei gleicher Hashfunktion immer den gleichen digitalen Fingerabdruck erzeugt. Dadurch ist es möglich, Diesen als Referenzwert zum Nachweis der Integrität eines versendeten Objekts (z.B. einer Bilddatei) heranzuziehen. In der hier zur Debatte stehenden Praxis werden nun die Hashfunktionen genutzt, um statt sensibler Daten (z.B. Passwörter) die Hashwerte in einer Datenbank abzulegen und somit keine Rückschlüsse auf Inhalt oder Größe der Originaldaten gezogen werden können.

Stellt ein Benutzer ein Passwort ein, wird dieses durch die Hashfunktion verschlüsselt. Der Schlüsselwert wird, an Stelle des Passwortes in der Datenbank gespeichert. Meldet sich der Benutzer nun erneut an, wird er zur Eingabe seines Passwortes aufgefordert. Dieses wird wieder mit der Hashfunktion verschlüsselt. Anschließend werden beide Hashwerte (der erneut generierte und der aus der Datenbank) mit einander verglichen. Ergibt sich eine Abweichung, dann war das eingegebene Passwort falsch und wird abgelehnt.

Ein Beispiel, wie die Erzeugung des Hashwertes vor sich geht (umgesetzt mit PHP):

Beispiel:

```
<?php
$password = "password";
$hashwert = md5($password);
print $hashwert;
?>
```

In diesem Beispiel erzeugen wir eine Variable `$password` und geben ihr den Wert „password“. Es sollte klar sein, dass dieser Wert im Normalfall auf einer Webseite aus einem Formular ausgelesen wird.

Die Variable wird nun an die PHP Funktion „`md5()`“ übergeben, welche den entsprechenden Hashwert errechnet und an die Variable „`$hashwert`“ übergibt. Dieser Wert wird nun ausgegeben, bzw. in eine Datenbank geschrieben. Als Ausgabe erhält man: „e22a63fb76874c99488435f26b117e37“.

Sollte sich der Benutzer nun verschrieben haben („Passwort“ statt „password“), ergibt sich ein vom Datenbankeintrag abweichender Wert, nämlich: „3e45af4ca27ea2b03fc6183af40ea112“. Der Benutzer konnte nicht authentifiziert werden.

Wie man sieht, hat die Änderung nur eines Teilwertes eklatante Auswirkungen auf das Endergebnis!

Die verschiedenen, hier gleich kurz betrachteten, Hashfunktionen funktionieren alle nach unterschiedlichen mathematischen Berechnungsmustern. Da diese aber unterschiedlich komplex sind, eignen sich nicht alle Hashfunktionen für jede benötigte Verschlüsselung. Es gibt Unterschiede in der Stärke der Verschlüsselung, die unmittelbare Auswirkungen auf die Sicherheit haben.

Die für diese Betrachtung relevanten, weil am häufigsten genutzten Hashfunktionen, sollen jetzt kurz vorgestellt werden

2.4.1. Message-Digest-Algorithm 5 (MD5)

Bei MD5 handelt es sich, wie auch bei SHA-1 und BCrypt, um eine so genannte dedizierte Hashfunktion (von lat. *dedicare*: widmen, zuordnen).

Hierbei wird der ursprüngliche Datenblock in 64 Verarbeitungsschritten zu einem 32 stelligen 32-Bit Wort verarbeitet. Dabei werden 128-Bit Hashwerte verarbeitet.

Wie im obigen Beispiel schon erläutert, ist die Länge der Ausgangswertes unerheblich: der Berechnungsalgorithmus führt immer zu einem 32 stelligen Ausgabe wert:

Beispiel:

```
„password“ ergibt „e22a63fb76874c99488435f26b117e37“
„Passwort“ ergibt „3e45af4ca27ea2b03fc6183af40ea112“ und
„Wir benutzen immer nur sichere Passwörter!“ ergibt „b2125a069a566315e0b23446eccaf347“.
```

Der MD5 gilt, auf Grund der zum Einsatz kommenden Berechnungsalgorithmen, als nicht so sicher wie der...

2.4.2. Secure-Hash-Algorithm (SHA)

Dieser arbeitet ähnlich wie der eben besprochene MD5 Algorithmus, verwendet aber 160-Bit Hashwerte und rechnet mit 80 Verarbeitungsschritten. Dabei werden 40 stellige 32-Bit Worte erzeugt.

„Der SHA-1 ist eine Hashfunktion, die aufgrund ihres großen Hashwertes und der durch Analysen hinreichend gut überprüften Qualität ihrer Kompressionsfunktion alle Eigenschaften einer starken Hashfunktion erfüllt und damit zur Durchführung von Integritätskontrollen in heutigen und zukünftigen Systemen gut geeignet ist.“ (Eckert, 2006, p. 363)

Wir bleiben, um ein Beispiel zu geben, bei den schon oben verwendeten Werten und nutzen dazu die PHP Funktion „`sha1()`“:

Beispiel:

„`password`“ ergibt „2e2b6533a81bc15430cf65de46dc097eeb5ba70c“
„`Password`“ ergibt „0719708d1cc814839bd818fdc27d446652f03383“ und
„*Wir benutzen immer nur sichere Passwörter!*“ ergibt
„4eb9076b28e558cfd3cdce66659ec871b48d7afe“.

2.4.3. BCrypt

BCrypt folgt einer etwas anderen Strategie. Während die schon genannten Algorithmen darauf ausgelegt wurden, die zugrunde liegenden Berechnungen möglichst schnell und ressourcenschonend durchzuführen, wird bei BCrypt sogar Wert auf lange und ressourcenbelastende Methoden gesetzt. Der Grund dafür: Je länger eine Berechnung zur Erzeugung des Hashwertes benötigt, desto schwieriger hat es auch ein potentieller Angreifer, den Schlüssel zu knacken.

Es werden insgesamt 521 Verarbeitungsschritte ausgeführt, in die ein Kostenfaktor (hier in Bezug auf die Rechenzeit) einbezogen wird. Je höher der Kostenfaktor, desto besser die Verschlüsselung, die aber auch mit höheren Kostenfaktor mehr Rechenzeit in Anspruch nimmt.

Am Ende steht ein 59 stelliger Hashwert, der aus Zahlen, Buchstaben (Groß- und Kleinschreibung) und einigen Sonderzeichen besteht und immer mit dem Dollarzeichen „\$“ beginnt.

Auch hier wieder das, schon öfter strapazierte Beispiel. Diesmal wird die PHP Funktion „`password_hash`“ mit dem Parameter „`PASSWORD_DEFAULT`“ verwendet. Der Parameter weist die Funktion an, den BCrypt Algorithmus zu Erzeugung des Hashwertes zu verwenden.

Code: `$hashwert = password_hash($password, PASSWORD_DEFAULT);`

Ein anderer Parameter ist „`PASSWORD_BCRYPT`“, Dieser benutzt, anders als zu erwarten, den so genannten Blowfish Algorithmus, auf dem BCrypt beruht, zur Berechnung. Das PHP Entwicklerteam folgt hier wohl einer eigenen Logik...

Beispiel:

„`password`“ ergibt
„\$2y\$10\$FuoVOUX73vd.pLhxK1PW7.Dd/K1vMAhNaVW3XAX7UXDeTxm09EbjO“
„`Password`“ ergibt

„\$2y\$10\$dGnDX9TYUW645rq/2.89d.1QSmKc0.EXP3kNTkN0D9zMsV9tmv9s6“ und
„Wir benutzen immer nur sichere Passwörter!“ ergibt
„\$2y\$10\$dH7RjrLu6kZjwupNufeeMfeY1lJWiQu8e24NPpdZmWwrKc.TRO“.

2.5. Software

Zur Sicherung von Webprojekten kann auch komplexe Software eingesetzt werden, die teilweise schon seit Jahrzehnten gepflegt wird. Hervorstechend ist hier XAMPP, dessen für diese Arbeit relevantesten Komponenten jetzt kurz angesprochen werden.

2.5.1. Apache

Apache ist ein Webserver, der als OpenSource Software, wesentlicher Bestandteil des XAMPP¹⁵ Projektes ist. Apache kann und wird als eigenständiges System auf Webservern verwendet. Im Rahmen von XAMPP wird Apache genutzt, um z.B. auf einem heimischen PC-System eine Serverumgebung zu simulieren. Auf Grund seiner inneren Struktur ist Apache fast vollständig Plattformunabhängig.

Apache verarbeitet Anfragen von Webbrowsern und liefert die Ergebnisse aus. Während der Verarbeitung der Anfragen (z.B. von PHP Seiten) kommen nun auch noch andere Funktionen ins Spiel, welche unter dem Strich nicht nur die Ergebnisse liefern, sondern auch einige Sicherheitsaspekte einfließen lassen und diese abarbeiten. Welche sind das?

Der Webserver versucht Scriptfehler in Anfragen abzufangen, diese weitestgehend zu reparieren oder zu blocken. Der Client bekommt eine Rückmeldung, aus der, je nach Einstellung, direkt korrigierend auf den Fehler geschlussfolgert und eingewirkt werden kann.

Eine weitere Aufgabe der Webserver ist es, sicher zu stellen, dass nur befugte Zugriffe auf die Verzeichnisstruktur erfolgen können. Ein wesentlicher Bestandteil dieses Verzeichnisschutzes ist die Anlage einer *.htaccess* Datei in den zu schützenden Ordnern.

Auf einem Server unterliegen alle Verzeichnisse und die darin liegenden Daten einem genau ausgearbeiteten System aus Zugriffsrechten. Für jeden Benutzer, der Zugriff auf den Server hat, muss explizit festgelegt werden, auf welche Verzeichnisse er zugreifen darf und auf welche nicht. Auch der Webserver selber unterliegt teilweise solchen Beschränkungen.

Unterschieden wird bei den Zugriffsrechten zwischen verschiedenen Benutzern und Benutzergruppen. Dabei wird zwischen administrativen und nicht-administrativen Benutzern unterschieden. Administrative Benutzer können hierbei globale Änderungen am System (Hard- und Softwareseitig) vornehmen und selber Benutzer zulassen und einer Benutzergruppe zuordnen.

Erfahrungsgemäß ist die größte Benutzergruppe die der nicht-administrativen Benutzer. Diese können, im Rahmen des ihnen zur Verfügung stehenden Rahmens und der für sie eingeräumten Freigaben, Aufgaben erledigen. Dazu gehören auch die Ausführung von Anwendungen und das Speichern von Daten in freigegebenen Ordnern oder Teilen einer Datenbank.

¹⁵ Das Projekt XAMPP ist unter <https://www.apachefriends.org> zu finden. Es beinhaltet den Apache Webserver, sowie die aktuellen Versionen von MySQL, PHP und Perl. Das gesamte Projekt ist ein OpenSource Paket.

Umgesetzt wird diese Freigabenregelung durch die bereits erwähnte *.htaccess* Datei. Unmittelbar mit *.htaccess* arbeitet die Datei *.htpasswd* zusammen. Beide Dateien werden in der Basisdatei des zu verwaltenden Verzeichnisses erstellt. Zur Bearbeitung ist ein Kommandozeilenbefehl oder ein Texteditor notwendig. Hier ein Beispiel für beide Dateien in einer windowsbasierenden Verzeichnisstruktur:

Beispiel:

.htpasswd:

```
c:\xampp\apache\htpasswd -c c:\xampp\htdocs\projekt_1\htpasswd [benutzer] [passwort]
```

.htaccess:

```
AuthUserFile: c:\xampp\htdocs\projekt_1\htpasswd
```

```
AuthName: „Protected Projekt 1“
```

```
AuthType Basic
```

```
<Limit GET POST>
```

```
require valid user
```

```
</LIMIT>
```

In diesem Beispiel wird durch die *.htpasswd* die im XAMPP Unterverzeichnis /apache liegende *htpasswd.exe* mit dem Parameter *-c* (erstellt den in der nächsten Zeile definierten Passwortpfad) aufgerufen und die Befehle abgearbeitet.

Die *.htaccess* beinhaltet als Erstes den Pfad des zu schützenden Verzeichnisses. Dann folgt der Name, welcher beim Zugriff auf dieses Verzeichnis angezeigt wird, wenn nach den Benutzerdaten gefragt wird. Der Authentifikationstyp ist Basic, d.h. der Verzeichniszugriff ist auf definierte Benutzer (*.htaccess*) beschränkt.

Die *<LIMIT>* Direktive legt die Methode der Zugriffskontrolle fest. In diesem Fall auf die http Methoden GET und POST¹⁶.

Durch konsequentes und durchdachtes Festlegen von Benutzern und Benutzergruppen und deren Zuordnung im Rahmen des Verzeichnisschutzes, wird die Sicherheit der Dateien auf dem Server weiter vertieft.

Des Weiteren übernimmt der Webserver die Aufgabe, die Kommunikation zwischen Server und Client zu verschlüsseln. Das ist notwendig, um zu vermeiden, dass die Datenübertragung im „Klartext“ erfolgt. Dadurch wird ein manipulativer Eingriff auf den Datenstrom und somit die Verfälschung von Informationen erschwert.

Die verschlüsselte Datenübertragung erfolgt über den SSL¹⁷. Dass die Übertragung verschlüsselt erfolgt, erkennt man, wenn in der Adressleiste des Browsers vor der eigentlichen Webadresse der Protokolltyp *https* angezeigt wird.

¹⁶ Das hier gezeigte Beispiel ist nicht besonders sicher! Besser wäre es, die *<LIMITExcept>* Direktive zu verwenden, da damit nicht die zugelassenen, sondern die ausgeschlossenen Methoden angeführt werden.

¹⁷ Secure Socket Layer(Sicherheitstransportschicht)

Beispiel:

<https://onedrive.live.com/getting-started>

Das ist z.B. bei LogIn Vorgängen beim Webhoster oder beim Onlinebanking der Fall.

Wie schon im Abschnitt „Verschlüsselungen“ erwähnt, wird für den Austausch von verschlüsselten Daten auf beiden Seiten (Server und Client) ein Schlüssel benötigt, der die Entschlüsselung ermöglicht. Eine Form dieses Schlüssels ist die Ausfertigung eines Zertifikates¹⁸. Diese Zertifikate (Serverzertifikate) kann man sich teilweise selber erstellen. Die Echtheit solcher selbsterstellter Zertifikate kann aber nicht automatisch überprüft werden. Somit ist eine echte Authentifizierung mittels eines solchen Zertifikates nicht möglich.

Für eine offiziell zugelassene Authentifizierung benötigt man ein Serverzertifikat einer Zertifizierungsstelle (CA) oder ein gleichwertiges Zertifikat mit Glaubwürdigkeit. Die Ausstellung solcher Serverzertifikate von offizieller Stelle ist kostenpflichtig, was bei Betreibern von kleinen Webseiten ein Kostenfaktor sein kann. Zwar werden zeitlich begrenzte Testzertifikate¹⁹ angeboten, was aber einer längerfristigen Lösung im Wege steht. Um diese Hürde zu umgehen, wurde die quelloffene Zertifizierungssoftware OpenSSL veröffentlicht, welche Teil von Apache, und somit auch von XAMPP ist.

2.5.2. OpenSSL

Mit OpenSSL hat man ein Werkzeug in der Hand, welches es ermöglicht Serverzertifikate für eigene Zwecke zu erstellen und diese auf die angeschlossenen Geräte nach Bedarf zu verteilen.

Ein selbst erstelltes Zertifikat steht in Sicherheitsaspekten in nichts nach und ist für die meisten Zwecke der Betreiber kleiner Webseiten oder Foren ausreichend. Am Rande bemerkt, hat es sogar Vorteile, Zertifikate selber zu erstellen und zu Verwalten, denn auch Zertifikate offizieller Stellen sind nicht immer vertrauenswürdig:

„Der staatliche Zertifikatsherausgeber von Indien hat offenbar falsche SSL-Zertifikate für Google-Dienste erstellt. [Google berichtet in seinem Online Security Blog](#), dass Zertifikate für mehrere Google-Dienste im Umlauf sind, die vor der Zertifizierungsstelle (Certificate Authority, CA) des [National Informatics Centre \(NIC\)](#) ausgestellt wurden. Dies ist nicht im Auftrag von Google geschehen und hat durchaus ernste Folgen: Ruft etwa ein Google-Nutzer seine Mails ab, kann der verschlüsselte Datenaustausch mit den Google-Servern auf dem Transportweg entschlüsselt, mitgelesen und neu verschlüsselt werden; ohne dass der Nutzer davon etwas mitbekommt.“ (www.heise.de, 2014)

Wie erstellt man nun ein Serverzertifikat (CA)²⁰ ?

¹⁸ Zertifikat (von lat. *certus*: sicher) (www.wissen.de, 2015)

¹⁹ Z.B. von www.sslmarket.de oder www.symantec.com. Die meisten Webhoste bieten ebenfalls günstige SSL Zertifikate an. Diese müssen monatlich verlängert werden.

²⁰ CertificateAuthority

Dazu bietet Apache die Möglichkeit über ein Kommandozeilenwerkzeug (openssl) in wenigen Schritten die erforderliche Datei zu kreieren. Unter Windows und Mac nutzt man dazu *cmd*²¹, unter Linux *bash*. Hier ein Beispiel:

Beispiel:

```
# Pfad zur Konfigurationsdatei bekannt machen:
set OPENSSL_CONF=c:\xampp\apache\conf\openssl.cnf

# Generierung des privaten Schlüssels.
# Wobei -des3 die Generierung eines Passwortes bedeutet.
# Anschließend der Ausgabename und die Verschlüsselungstiefe in Bit:
C:\xampp\apache\bin\openssl.exe genrsa -des3 privatekey.key 2048

# Erstellung des öffentlichen Schlüssels.
# Wobei -key den Namen des privaten Schlüssels bedeutet.
# Nach -out wird der Ausgabename des öffentlichen Schlüssels benannt:
C:\xampp\apache\bin\openssl.exe req -new -key privatekey.key -out publickey.csr

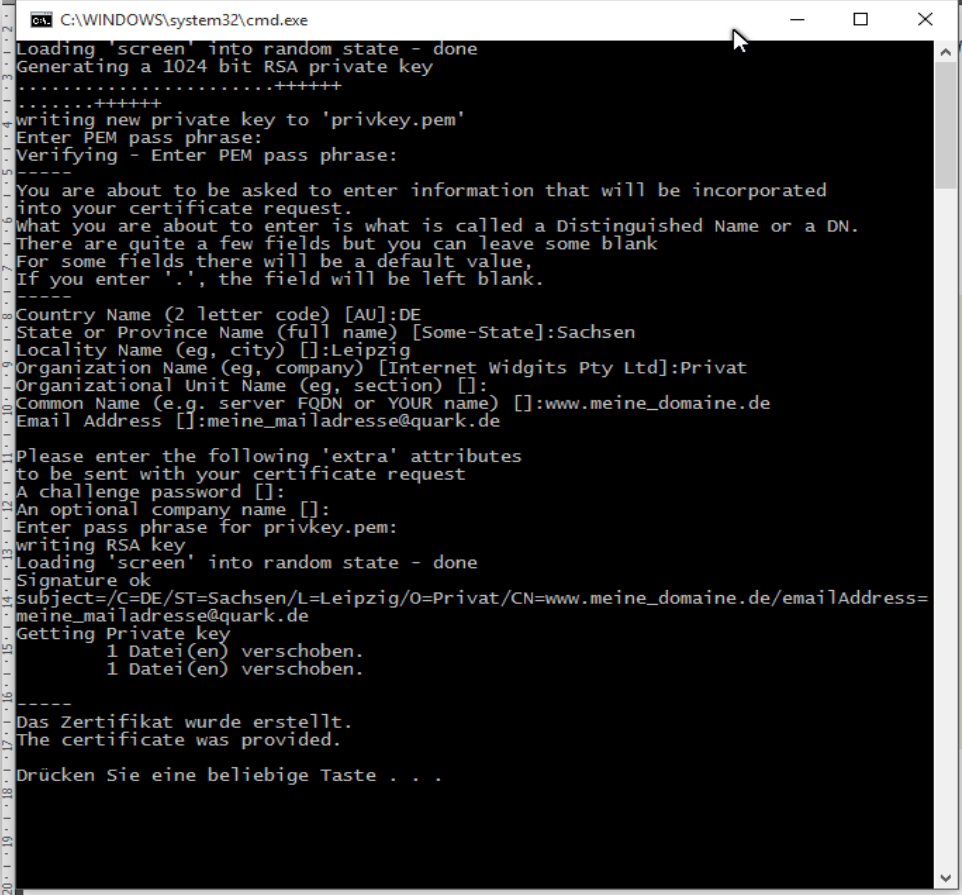
# Jetzt erfolgt (sofern falls richtig eingetippt) die Abfrage der Schlüsseldaten:
# Land (DE), Provinz (Sachsen), Stadt (Leipzig), Organisation (Privat),
# Common Name (www.meine_domaine.de), E-Mail (kann man, muss man aber nicht),
# Passwort (frei lassen) und Kurzname der Gesellschaft (frei lassen).
```

Seit Windows 10 kann man ohne Probleme auch im Pfad „C:\XAMPP“ die vorgefertigte Batch „makecert.bat“ anklicken und bekommt, nach Eingabe eines Passwortes für den privaten Schlüssel und Abfrage der oben genannten Eckdaten, sein fertiges Zertifikat im Ausgabepfad²² geliefert (siehe Abbildung 2)

Zu beachten ist, dass eingegebene Passwörter nicht sichtbar angenommen werden. Auch nicht mit Platzhaltern (Sterne etc.)!

²¹ Seit Windows 7 kann man zu solchen Aufgaben auch die Windows Power Shell ISE verwenden! Diese ist deutlich leistungsfähiger, als das einfache cmd. Sie ermöglicht, wie bash unter Linux, das Speichern von Abläufen.

²² C:\xampp\apache\conf\ssl.csr



```
C:\WINDOWS\system32\cmd.exe
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Sachsen
Locality Name (eg, city) []:Leipzig
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Privat
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:www.meine_domaine.de
Email Address []:meine_mailadresse@quark.de

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Enter pass phrase for privkey.pem:
writing RSA key
Loading 'screen' into random state - done
Signature ok
subject=/C=DE/ST=Sachsen/L=Leipzig/O=Privat/CN=www.meine_domaine.de/emailAddress=
meine_mailadresse@quark.de
Getting Private key
1 Datei(en) verschoben.
1 Datei(en) verschoben.
-----
Das Zertifikat wurde erstellt.
The certificate was provided.

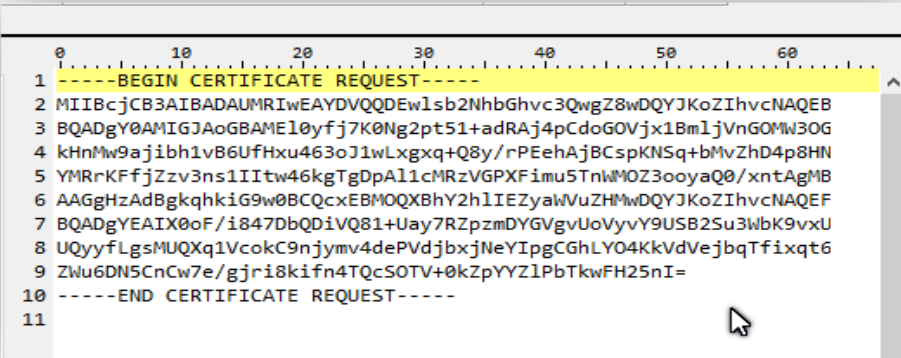
Drücken Sie eine beliebige Taste . . .
```

A. Raue © 2015

Das

Eingabefenster der makecert.bat (Abbildung 3)

Das Ergebnis sieht nun so aus (Abbildung 3):



```
0 10 20 30 40 50 60
1 -----BEGIN CERTIFICATE REQUEST-----
2 MIIBcCB3AIBADAUMRIwEAYDVQQDEw1sb2NhbgHvc3QwgZ8wDQYJKoZIhvcNAQEB
3 BQADgY0AMIGJAoGBAMEl0yfj7K0Ng2pt51+adRAj4pCdoGOVjx18m1jVnGOMW3OG
4 kHnMw9ajibh1vB6UfHxu463oJ1wLxgXq+Q8y/rPEehAjBCspKNSq+bMvZhd4p8HN
5 YMRrKFfjZzv3ns1IItw46kgTgDpAl1cMRzVGPXFimu5TnWMOZ3ooyaQ0/xntAgMB
6 AAGGhzAdBgkqhkiG9w0BQCxEMBMOQXBhY2h1IEZyaWVvZHMwDQYJKoZIhvcNAQEF
7 BQADgYEAIIX0oF/i847DbQDiVQ81+Uay7RZpzmDYGvUoVyvY9USB2Su3WbK9vXu
8 UQyyfLgsmUQXq1VcokC9njymv4dePvdjbxjNeYIpgCGhLY04KkVdVejbqTfixqt6
9 ZWu6DN5CnCw7e/gjri8kifn4TQcSOTV+0kZpYYZ1PbTkWFH25nI=
10 -----END CERTIFICATE REQUEST-----
11
```

A. Raue © 2015

Der generierte öffentliche Schlüssel (Abbildung 4)

Nun muss dieses Zertifikat auf dem Server übertragen werden und in eine Zertifikatsdatenbank eingetragen werden, um dort seinen Dienst zu tun. Die Erstellung und Wartung einer solchen Datenbank ist ein komplexer Vorgang. Die Beschreibung der Erstellung und der Pflege würde den Rahmen dieser Arbeit sprengen. Deshalb verweise ich auf eine längere, gut geschriebene Abhandlung²³ zu dieser Thematik.

²³ <http://root.bbs-duew.de/ca/OpenSSL-Windows-Zertifikat-Erstellung.htm> (Stand Mai 2010)

3. Zusammenfassung

Wir haben gesehen, dass es nicht sehr leicht ist, auch im kleinen Bereich, eine Webseite sicher zu erstellen. Viele Dinge sind zu beachten, spielen doch ein Vielzahl an Faktoren eine Rolle. Nicht immer sind es Angriffe von außen, die eine Gefahr darstellen. Manchmal ist es nur eine Unachtsamkeit und schon mogelt sich ein Fehler ins Programm, schmuggelt sich an Fehler Routinen vorbei, verursacht kleine aber störende Mängel und schaukelt sich letztendlich zum GAU hoch.

Aber wir haben auch gesehen: bei der richtig eingesetzten Verwendung der nunmehr bekannten Möglichkeiten, müsste es erreichbar sein, auch ein kleines Webprojekt sicher zu gestalten. Entscheidend dabei ist, dass man die Sicherheit des Projektes nicht hinten an stellt, sondern sich der Verantwortung bewusst ist, die man dabei trägt.

Fakt ist, dass eine hundertprozentige Sicherheit wohl kaum jemals erreicht werden kann. Das liegt schon in der Sache an sich: jeder Schritt zur Verbesserung der Sicherheit, zieht automatisch einen Schritt nach sich, welcher diese Sicherheit wieder umgehen will. Es herrscht also ein ständiger Wettbewerb zwischen den Lagern. Ständig werden neue Methoden entwickelt, Sicherheitssysteme zu hintergehen. Und in Gegenzug werden Sicherheitsmethoden immer stärker forciert. Das kann seltsame Ausmaße annehmen. Wenn zum Beispiel, wie im Abschnitt *Verschlüsselungen* schon erwähnt, zur angeblichen Steigerung der Sicherheit, Verschlüsselung bis hin zum Verbot unterdrückt werden soll. Dann wird meiner Meinung nach ein Schritt in die falsche Richtung getan. Wird doch dadurch eher ein Sicherheitskonzept beschnitten, denn ein Anderes gestärkt.

Ich selber habe seit 1996 verschiedenen private und geschäftliche Webseiten aufgebaut, gepflegt und wieder verworfen. Dabei habe ich so ziemlich alle Höhen & Tiefen durchlebt, die man auf diesem Gebiet erleben kann. Von „*Tut uns leid! Die Webseite ist nicht mehr erreichbar!*“²⁴ bis zur Abgabe einer Stellungnahme, weil mein Server als Zwischenlager für Torrent Dateien genutzt wurde, war alles dabei. Mit der Zeit kommt die Erfahrung die man in Punkto Sicherheit & Web unweigerlich macht. Und wenn man diese Erfahrung nun noch dauerhaft umsetzt, dann hat man einen kleinen Beitrag geliefert, um das WWW etwas sicherer zu machen.

„Schönen guten Tag, wir kommen vom Institut für Internet Sicherheit und analysieren Passwörter. Sind Sie so nett und schreiben Ihr Passwort auf? Natürlich bleiben Sie anonym, so dass keiner etwas damit anfangen kann. Sie können das Passwort einfach auf ein weißes Blatt Papier schreiben.“ (Schrödel, 2014, p. 303)

²⁴ Ich habe es zweimal erlebt, dass faktisch über Nacht mein kompletter Webspace im Datennirvana verschwand, weil der Freehoster Inkasso angemeldet hatte und einfach „dicht“ machte.

4. Verzeichnisse

4.1. Literaturverzeichnis

Autorenteam, 2013. *Fachlexikon Computer*. s.l.:Brockhaus Verlag.

Eckert, C., 2006. *IT-Sicherheit (Konzepte - Verfahren - Protokolle)*. 4. überarbeitete Auflage Hrsg. s.l.:Oldenbourg Verlag München Wien.

Schrödel, T., 2014. *Ich glaube, es hackt!*. 3. Auflage Hrsg. s.l.:Springer Spectrum.

www.computerwoche.de, 2014. *www.computerwoche.de*. [Online]

Available at: <http://www.computerwoche.de/a/angriffe-auf-die-eigene-it-die-beliebtesten-methoden-der-hacker-2014-und-2015,3098871>

[Zugriff am 23 09 2015].

www.heise.de, 2014. *www.heise.de/security*. [Online]

Available at: <http://www.heise.de/security/meldung/Indien-stellte-falsche-Google-Zertifikate-aus-2252544.html>

[Zugriff am 29 09 2015].

www.hosting-review.com, 2015. *www.Hosting-Review.com*. [Online]

Available at: http://lp.hosting-review.com/deutschland/?aff_sub=H-R-Deutschland-GR&aff_sub2=webhosting%20vergleich&aff_sub3=b&gclid={gclid}&gclid=Cj0KEQjw4ZOwBRDoxpjAvPXA15MBEiQAEek_3twb7Fd9LhkotodERcU5Bx8FDJfix-idnTdAemBv5wUaAleS8P8HAQ&ga1=H-R-Deutschland-GR&ga2=webho

[Zugriff am 21 September 2015].

www.netzpolitik.org, 2015. *www.netzpolitik.org*. [Online]

Available at: <https://netzpolitik.org/2015/eu-kommission-hat-weiterhin-bedenken-zu-verschluesselung-und-plant-gespraech-mit-internetdienstleistern/>

[Zugriff am 25 09 2015].

www.pwc.de, 2014. *www.pwc.de*. [Online]

Available at: <http://www.pwc.de/de/pressemitteilungen/2014/weltweite-pwc-umfrage-zur-it-sicherheit.html>

[Zugriff am 26 09 2015].

www.strato.de, 2015. *www.strato.de*. [Online]

Available at:

https://www.strato.de/apps/CustomerService?sessionID=e15c758c03d8ebfdecc93713c84927&node=kds_ServerSideAntiVirus

[Zugriff am 28 09 2015].

www.sueddeutsche.de, 2014. *www.sueddeutsche.de*. [Online]

Available at: <http://www.sueddeutsche.de/digital/sicherheit-im-internet-nerv-mich-nicht-1.1932962-2>

[Zugriff am 18 09 2015].

www.wissen.de, 2015. *www.wissen.de*. [Online]

Available at: <http://www.wissen.de/fremdwort/zertifizieren>

[Zugriff am 28 09 2015].

Alle Abbildungen: © by Alph Raue 2015

4.2. Abbildungsverzeichnis

E-Mail Abfrage (Abbildung 1).....	9
Abfragemaske (Abbildung 2).....	10
Das Eingabefenster der makecert.bat (Abbildung 3)	20
Der generierte öffentliche Schlüssel (Abbildung 4).....	20

5. Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit bzw. Leistung eigenständig, ohne fremde Hilfe und nur unter Verwendung der angegebenen Hilfsmittel und Quellen angefertigt habe. Alle sinngemäß und wörtlich übernommenen Textstellen aus der Literatur bzw. dem Internet habe ich als solche kenntlich gemacht.

A handwritten signature in black ink, appearing to read 'A. Raue', is written over a horizontal dashed line.

Alph Raue

Leipzig, 29.09.2015

Statistik

Worte gesamt		5797
Worte Text	ohne Fußnoten und Verzeichnisse.	5432
Seiten gesamt		26
Seiten Text	nur Facharbeit	19

Arbeitstabelle

Genaue Wortverteilung (Stand 29.09.2015)

Kapitel	Unterkp	Prozent (soll)	Worte	Prozent (ist)	Worte	Erlid.
Alle		100%	5000	100%	5432	
1.		15%	750	17%	942	192
2.		75%	3750	76%	4109	359
	2.1.	20%	1000	14%	735	-265
	2.2.	15%	750	6%	338	-412
	2.3.	10%	500	11%	605	105
	2.4.	20%	1000	23%	1239	239
	2.5.	10%	500	22%	1192	692
3.		10%	500	7%	381	119

Facharbeit begonnen am: 07.09.2015 (theoretischer Teil), 21.09.2015 (schriftlicher Teil) und beendet am 30.09.2015 01:45 Uhr.