Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Finn Christopher Detjen

Entwicklung eines Programms zur Überprüfung der Einhaltung rechtlicher und organisatorischer Vorgaben an Internet-Auftritte.

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Finn Christopher Detjen

Entwicklung eines Programms zur Überprüfung der Einhaltung rechtlicher und organisatorischer Vorgaben an Internet-Auftritte.

Bachelorarbeit eingereicht im Rahmen Bachelorprüfung

im Studiengang Wirtschaftsinformatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter : Prof. Dr. Martin Becke

Abgegeben am 20.06.2019

Finn Christopher Detjen

Thema der Bachelorarbeit

Entwicklung eines Programms zur Überprüfung der Einhaltung rechtlicher und organisatorischer Vorgaben an Internet-Auftritte.

Stichworte

Reputation, Docker, Compliance, Internet-Auftritte, IT-Sicherheit

Kurzzusammenfassung

In dieser Arbeit wird die rechtliche Situation in Deutschland betrachtet, welche Vorgaben für Internet-Auftritte existieren. Um die Einhaltung der Vorgaben zu überprüfen, wird ein Programm entwickelt, welches die Internet-Auftritte untersucht und aus den ermittelten Daten eine Benotung errechnet. Diese Benotung ist ein Index für die Umsetzung der Vorgaben.

Für die Ermittlung der Benotung werden nicht nur inhaltliche Vorgaben überprüft, sondern auch Aspekte, die die Sicherheit der Infrastruktur betreffen, analysiert.

Finn Christopher Detjen

Title of the paper

Development of a program to check compliance with legal and organizational requirements for Internet presences.

Keywords

Reputation, docker, compliance, internet presences, IT Security

Abstract

This thesis examines the legal situation in Germany, which requirements exist for Internet presences. In order to check the compliance with the requirements, a program is developed which examines the Internet presences and calculates a rating from the determined data. This grading is an index for the implementation of the specifications.

To determine the grading, not only content specifications are checked, but also aspects relating to the security of the infrastructure are analyzed.

Abbildungsverzeichnis

Inhaltsverzeichnis

1 Einleitung

„Von drückenden Pflichten kann uns nur die gewissenhafteste Ausübung befreien.“ [1]

Der Gesetzgeber stellt Regeln auf, denn ohne würde die Gesellschafft nicht funktionieren und Unternehmen könnten nicht in einem fairen Wettbewerb gegeneinander antreten. Daraus folgt die Selbstverständlichkeit die aufgestellten Regeln zu befolgen. In der Unternehmenssprache hat sich daraus der Begriff Compliance gebildet. Compliance bedeutet, dass sowohl rechtliche Bestimmungen eingehalten werden, als auch die unternehmensinternen Regeln. Zur Einhaltung dieser Regeln wird oftmals ein Compliance Management System in den Unternehmen eingesetzt, um die entsprechenden Prozesse zu steuern.

Compliance ist einer der Säulen um sich einen positiv behafteten Namen aufzubauen. Für Unternehmen ist es besonders wichtig, dass ihr Name bei den Stakeholdern mit positiven Eigenschaften versehen ist. Dieses Konstrukt der Reputation kann je nach Interessengruppen unterschiedliche Aspekte beinhalten. Für die interne Reputation bei den Mitarbeitern eines Unternehmens kann es zum Beispiel wichtig, dass ein positives Arbeitsklima herrscht und es Möglichkeiten der Fortbildung gibt.
Bei der externen Reputation, also aus der Sicht der Kunden, sind andere Eigenschaften von Bedeutung. Beispielsweise die persönliche Betreuung im Verkauf oder wie fortschrittlich und ideenreich die Produkte sind.

In einer Studie, die in Zusammenarbeit mit der Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt erstellt wurde, gab jedes dritte Unternehmen an, dass sie ein Compliance Management System eingeführt haben, um Reputationssicherung zu betreiben (vgl. Laute 2015).

[1] [Goethe 1829]

1.1 Motivation

Die rechtlichen Anforderungen an ein Unternehmen in der Informations- und Telekommunikationsbranche steigen stetig. Durch die Einführung der europäischen Datenschutzgrundverordnung (EU-DSGVO) kamen 2018 diverse Anforderungen hinzu.

Auch die stetig wachsende Zahl der Angriffe auf die IT-Infrastruktur von Unternehmen fordert diese immer mehr heraus. Dabei sind nicht nur neue Angriffstechniken von Interesse, sondern oftmals auch bekannte Sicherheitslücken, welche bisher aber nicht geschlossen wurden.

Beide Aspekte haben für ein Unternehmen immer mehr an Bedeutung gewonnen, denn die Kunden wollen sich mit Ihrem Produkt oder mit der Benutzung eines Dienstes sicher fühlen können.

1.1 Zielgruppe

Vorausgesetzt werden in dieser Arbeit fundamental wichtige Kenntnisse der Softwareentwicklung im Webbereich. Ebenso wird vorausgesetzt, dass grundlegende Kenntnisse im Umgang mit Linux und IT-Sicherheit vorhanden sind.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines Systems, welches beliebige Internet-Auftritte auf Vorhandensein von rechtlichen Pflichtangaben prüft. Ebenso wird das grundlegende Erscheinungsbild des Servers nach außen kontrolliert, ob mögliche Schwachstellen ersichtlich sind. Die gesammelten Ergebnisse sollen visualisiert werden und als Datensatz für die weitere Verarbeitung in externen Tools zur Verfügung gestellt werden.

1.3 Struktur

In dem hierauf folgendem Kapitel Grundlagen wird einerseits der Begriff der Reputation näher erläutert und dargestellt, welche Aspekte vor Allem im Bereich der IT von Bedeutung sind. Andererseits werden die grundsätzlichen gesetzlichen Anforderungen an einen Internet-Auftritt beschrieben und die technischen Grundlagen für diese Arbeit erläutert.
In dem dritten Kapitel wird der Aufbau des zu entwickelnden System und dessen Implementierung dokumentiert. Dabei sind die Architektur, die Sicherheitsaspekte und der Funktionsumfang von Bedeutung.
Im darauf folgenden vierten Kapitel werden Einschränkungen des Funktionsumfanges des Programmes beschrieben.
Kapitel Fünf soll einen Überblick über das entwickelte System und dessen Benutzung liefern.
Das letzte Kapitel ist schließlich dem Fazit über das erstellte System und Ausblick auf die Erweiterungs- und Verbesserungsmöglichkeiten gewidmet.

2 Grundlagen

Dieses Kapitel beschäftigt sich im ersten Unterkapitel mit der Reputation von Unternehmen und geht genauer auf die Reputation in der Informations- und Telekommunikationsbranche ein.
Darauf folgen die Anforderungen die der Gesetzgeber an einen Internet-Auftritt stellt. Dabei sind nicht nur allgemeine Anforderung im Fokus, sondern auch die, die sich durch datenschutzrechtliche Aspekte entstehen.
Der dritte Teil beschäftigt sich mit der Containervirtualisierungssoftware Docker.

2.1 Reputation

Bereits 2004 stellte Schwalbach fest, dass für den Begriff der Reputation keine allgemeingültige Definition in der wirtschaftswissenschaftlichen Literatur vorliegt. Aus den am häufigsten verwendeten Definitionen definierte er den Begriff wie folgt (vgl. Schwalbach 2004, S.1):

„Demnach verbindet man mit Reputation das Ansehen bzw. die Qualität einer Person, eines Produktes, einer Organisation oder allgemein einer Institution, wie sie von anderen wahrgenommen wird.“[2]

Mit dieser Definition grenzt er den Begriff auch gegenüber dem der Marke ab. Eine Marke definiert sich durch die Qualität des Produktes. Die Reputation eines Unternehmens wird positiv von der Marke beeinflusst, enthält aber noch weitere Faktoren. So kann sich ein hohes Umweltbewusstsein oder eine faire Bezahlung positiv auf die Reputation eines Unternehmens auswirken.

Um eine hohe Reputation zu erlangen bedarf es im Allgemeinen vier Eckpunkte, die erfüllt sein müssen. Dazu zählen die Glaubwürdigkeit, die Verlässlichkeit, die Vertrauenswürdigkeit und die Berechenbarkeit eines Unternehmens (vgl. Fombrun 1996, S.71 u. S.72). Um diese immateriellen Werte aufzubauen bedarf es viel Zeit. In einer sich schnell wandelnden Gesellschafft sind diese Werte besonders wichtig, um weiter zu bestehen. Unterstützend wirken kann an dieser Stelle auch ein positives Ergebnis bei Tests von unabhängigen Prüforganisationen.

[2][Schwalbach 2004, S. 1]

Der Wert der Reputation eines Unternehmens zählt zu den immateriellen Gütern, die das Unternehmen besitzt. Durch Reputationsmanagement wird unternehmensseitig versucht diesen Wert zu schützen und zu steigern.

„It takes 20 years to build a reputation and five minutes to ruin it."[3]

Diese Aussage von Warren Buffett zeigt einerseits, dass der Aufbau einer positiven Reputation einiges an Zeit in Anspruch nimmt. Andererseits zeigt es auch, dass es innerhalb kürzester Zeit möglich ist diesen Ruf wieder zu verlieren. Ein gutes Beispiel dafür ist die Verwicklung der Volkswagen AG in die Manipulation von Abgaswerten. Innerhalb weniger Monate verlor die Aktie des Konzerns im Jahre 2015 über 60 Prozent an Wert (vgl. Boerse 2019). Seitdem kämpft der Konzern gegen den Vertrauensverlust an. Durch das Bekanntwerden verlor das Unternehmen erheblich an Reputation in der Gesellschaft. Diesen Verlust wieder zu kompensieren wird wieder lange Zeit in Anspruch nehmen. Bis heute ist es der Volkswagen AG nicht gelungen diesen Reputationsverlust wieder einzuholen.

Auch in der Informations- und Telekommunikationsbranche ist eine hohe Reputation von Bedeutung. Neben den oben genannten Aspekten, sind vor Allem branchenspezifische Werte von Bedeutung. Dazu zählen unter anderem der datenschutzkonforme Umgang mit Kundendaten oder die sicherheitstechnischen Maßnahmen zum Schutz der Daten und der Infrastruktur. Unternehmen, die mit Vorsicht private Daten behandeln und sich dem Schutz jener verschrieben haben, haben oftmals ein positives Erscheinungsbild in der Öffentlichkeit.

Neben diesen Maßnahmen eines Unternehmens ist auch die Bewertung eines Unternehmens auf Suchmaschinen und in Onlineshops durch Kunden von Bedeutung für die Reputation eines Unternehmens von Bedeutung. Oftmals haben negative Bewertungen direkt Einfluss auf die Reputation eines Unternehmens. Einhergehend sind damit auch Umsatzverluste. Aber auch aus negativen Rezensionen kann ein positiver Eindruck hinterlassen werden, wenn auf die Kritik eingegangen wird und Fehler öffentlich eingestanden werden. Damit kann auch das öffentliche Erscheinungsbild des Unternehmens positiv beeinflusst werden.

Sicherheitsvorfälle, welche die IT-Infrastruktur betreffen, werden von nur rund 30 Prozent der Unternehmen an die zuständigen staatlichen Stellen gemeldet. In einer Studie von 2017, durchgeführt von Euroforum, gaben 41 Prozent von den betroffenen Unternehmen an den Vorfall nicht gemeldet zu haben, da sie durch die Veröffentlichung des Vorfalls einen Imageschaden befürchten (vgl. Haake 2017). Diese Imageschäden erzeugen eine negative öffentliche Wahrnehmung des Unternehmens und stehen deshalb in einem direkten Zusammenhang mit der Reputation des Unternehmens.

[3] [Schifrin 2015]

2.2 Gesetzliche Anforderungen an einen Internet-Auftritt

In Deutschland gilt für die Veröffentlichung im World Wide Web die Anbieterkennzeichnung. Diese ist auch umgangssprachlich als Impressumspflicht bekannt. Geregelt ist die Anbieterkennzeichnung im Telemediengesetz. Es trat zum März 2007 in Kraft und wurde zum letzten Mal 2017 geändert. Die grundlegenden inhaltlichen Vorschriften sind dabei nahezu unverändert geblieben. Besondere Bedeutung für einen Internet-Auftritt haben dabei der fünfte und dreizehnte Paragraph des Gesetzes. Zusätzlich zum Telemediengesetz befinden sich auch noch in der Datenschutzgrundverordnung im Artikel dreizehn Pflichtangaben.

§5 Telemediengesetz– Allgemeine Informationspflichten

In diesem Paragraphen des Telemediengesetztes ist die Anbieterkennzeichnung beschrieben. Diese gilt für Auftritte, die dem geschäftsmäßigen Zweck dienen. Laut einem Urteil des OLG Hamburg von 2007 gelten diese Pflichten auch für private Internet-Auftritte, wenn auf ihnen Werbung für sich selber oder für Dritte dargestellt wird (vgl. OLG HH 2007).

Die Informationen müssen, laut Absatz eins, *„[...]leicht erkennbar, unmittelbar erreichbar und ständig verfügbar[...]“* sein. Dies bedeutet, dass von jedem Punkt des Auftrittes auf diese Informationen der Zugriff möglich sein muss und dass die Informationsangaben als solche ersichtlich sein müssen. Im Allgemeinen hat sich dafür der Begriff Impressum respektive Legal Notices im Englischen eingebürgert.

Welche Angaben erforderlich sind, sind dabei in den Nummern eins bis sieben des ersten Absatzes beschrieben:

Nummer eins beinhaltet die Informationen, die angegeben werden müssen, zum Namen und der Anschrift. Wenn es sich um eine juristische Person handelt, dann müssen zusätzliche Informationen zu der Rechtsform und den vertretungsberechtigten Personen angegeben werden.

Nummer zwei behandelt die Angaben zu den Kontaktmöglichkeiten des Auftritt Betreibers. Vorgeschrieben ist eine Angabe, die *„[...]schnelle elektronische Kontaktaufnahme und unmittelbare Kommunikation mit ihnen ermöglichen[...]“.* Dies kann eine Telefonnummer sein. Verpflichtend ist allerdings die Angabe einer E-Mailadresse.

Nummer drei verlangt eine Angabe der zuständigen Aufsichtsbehörde, wenn der angebotene oder erbrachte Dienst mit einer Tätigkeit verbunden ist, die eine behördliche Zulassung bedarf.

In Nummer vier sind die Angaben zu dem entsprechenden Register beschrieben, in dem der Anbieter eingetragen ist. Ebenso ist die ihm zugeteilte Registernummer anzugeben.

Wenn es sich bei dem Anbieter um eine Einzelperson handelt, dann sind laut Nummer fünf noch Angaben zu der Kammer, der gesetzlichen Berufsbezeichnung, dem Staat, der ihm die Berufsbezeichnung verliehen hat, sowie der berufsrechtlichen Regelung und deren Zugänglichkeit zu machen.

Wenn der Auftrittsbetreiber eine Umsatzsteueridentifikationsnummer oder eine Wirtschafts-Identifikationsnummer besitzt, dann ist diese nach Nummer sechs anzugeben.

In der letzten Nummer sieben ist beschrieben, dass eine Angabe erfolgen muss, wenn es sich bei dem Betreiber um eine Gesellschaft handelt, welche sich gerade in dem Prozess der Abwicklung oder der Liquidation befindet.

§13 Telemediengesetz– Pflichten des Dienstanbieters
Dieser Paragraph des Telemediengesetzes befasst sich mit der Informationspflicht über die Speicherung und dem Umgang mit personenbezogenen Daten.

Im ersten Absatz wird dem Dienstanbieter vorgeschrieben, dass der Nutzer zu Nutzungsbeginn des Dienstes darüber informiert werden muss, in welchem Maße und zu welchem Zweck personenbezogene Daten erhoben, verwendet und Verarbeitet werden. Ebenfalls muss der Nutzer informiert werden, wenn die erhobenen Daten außerhalb der Europäischen Union verarbeitet oder vorgehalten werden. Diese Informationsweitergabe muss in einer, für den Nutzer, verständlichen Form vorliegen. Diese Informationen müssen dem Nutzer jederzeit zugänglich gemacht werden.

Der zweite Absatz beschreibt den Vorgang der Einwilligung. Diese kann elektronisch vollzogen werden, wenn der Benutzer diese bewusst und eindeutig macht. Diese Einwilligung muss protokolliert werden. Dabei muss auch der Inhalt der Einwilligung dem Nutzer jederzeit zur Verfügung stehen. Auch regelt dieser Absatz, dass der Nutzer die Einwilligung jederzeit widerrufen können muss. Auf dieses Widerrufsrecht muss der Nutzer aktiv, laut Absatz drei hingewiesen werden.

Absatz vier beschreibt, dass der Nutzer jederzeit die Nutzung des Dienstes beenden können muss. Ebenfalls wird vorgeschrieben, dass die erhobenen Daten nach der Beendigung beziehungsweise nach Ende von gesetzlichen Sperrfristen gelöscht werden müssen.

Im fünften Absatz wird der Betreiber des Internet-Auftrittes dazu verpflichtet den Nutzer darauf hinzuweisen, wenn er ihn an einen weiteren Dienstanbieter weiterleitet.

Sind kostenpflichtige Angebote in dem Dienst enthalten, dann muss dem Nutzer, wenn zumutbar, eine anonyme Zahlungsart oder eine Zahlung unter einem Pseudonym angeboten werden. Dies regelt der sechste Absatz.

Absatz sieben regelt, dass der Anbieter des Dienstes alle ihm zumutbaren Maßnahmen trifft, um die erhobenen personenbezogenen Daten vor Störungen der Integrität und Diebstahl zu schützen. Dabei ist jeweils der aktuelle Stand der Technik zu berücksichtigen.

Im letzten Absatz acht weist der Gesetzgeber darauf hin, dass der Nutzer das Recht gegenüber den Dienstanbieter hat auf Herausgabe der von ihm gesammelten personenbezogenen Daten.

Artikel 13 DSGVO - Informationspflicht bei Erhebung von personenbezogenen Daten bei der betroffenen Person
Dieser Artikel der Datenschutzgrundverordnung regelt neben dem §13 des Telemediengesetzes, welche Informationen dem Nutzer eines Dienstes bereitgestellt werden müssen, wenn der Dienst personenbezogene Daten erhebt.

Laut Absatz eins des Artikels müssen dem Nutzer folgende Daten, wenn zutreffend, zur Verfügung gestellt werden, wenn personenbezogene Daten erhoben werden:

- Name und Kontaktdaten des Verantwortlichen
- Kontaktdaten des Datenschutzbeauftragten
- Zwecke der Erhebung und Rechtsgrundlage für die Verarbeitung
- die berechtigten Interessen für die Verarbeitung der Daten
- Empfänger der personenbezogenen Daten
- Grund für die Weitergabe der erhobenen Daten an Dritte

Absatz zwei gibt dem Nutzer das Recht, dass er Informationen über die Dauer der Speicherung seiner personenbezogenen Daten erhält. Ebenfalls müssen die personenbezogenen Daten dem Nutzer auf Verlangen zur Verfügung gestellt werden. Zudem erhält der Nutzer das Löschrecht über die erhobenen Daten. Zusätzlich dazu muss der Nutzer über *„das Bestehen eines Beschwerderechts bei einer Aufsichtsbehörde“* informiert werden.

Wenn die personenbezogenen Daten zu einem anderem Zweck verarbeitet werden, als wofür sie erhoben wurden, dass muss der Verantwortliche dies dem Nutzer mitteilen.

Weitere Gesetze
In den Artikel 15 und 17 der Datenschutzgrundverordnung erhält der Nutzer weitere Rechte auf Hinsicht der von ihm erhobenen personenbezogenen Daten.
Artikel 15 schreibt ihm dabei das Recht zu, dass er ein Recht zur Auskunft über die Daten, die von ihm erhoben wurden, hat.
Artikel 17 gibt ihm das Recht, dass auf seinem Verlangen hin, seine erhobenen personenbezogenen Daten gelöscht werden.
Auf diese Gesetze wird nicht weiter eingegangen, da deren Inhalte bereits in anderen Gesetzen weiter oben in diesem Unterkapitel behandelt wurden.

2.3 Docker

Mit Docker Containern wird von Docker das Prinzip der Container aus der Logistik- und Transportbranche aufgegriffen. Die Container bieten ein genormtes Maß und deren Handhabung nach außen ist bei jedem identisch. Dabei ist der Inhalt und dessen Form für die Verwendung nicht von belangen.
Ebenso verhalten sich die Docker Container. Sie sind für sich alleinstehende Systeme, das heißt sie beinhalten ein eigenes Betriebssystem und betreiben beziehungsweise stellen Anwendungen bereit. Nach außen bieten die Docker Container eine einheitliche Schnittstelle um jene anzusprechen und zu verwalten.

Docker baut auf Techniken auf, welche im Kernel von Linux vorhanden sind. Der Linuxkernel stellt Schnittstellen zu der Hardware bereit und ist der zentrale Bestandteil von Linux basierten Betriebssystemen. Auf diese Schnittstellen greift das jeweilige Betriebssystem selber zu. Diese Funktionen des Kernels macht sich Docker zu nutzen und lässt abgeschottete Prozesse direkt auf dem Host laufen. Damit spart sich Docker eine Virtualisierungssoftware ein und wird somit leichtgewichtiger.

Wie in Abbildung 1 zu sehen, wurde dies bis Docker-Version 0.9 mittels externer Bibliotheken namens *libvirt, lxc* und *systemd-nspawn* gelöst. Seit Version 0.9 nutzt Docker eine eigene, integrierte Bibliothek namens *libcontainer.* Die Nutzung der eigenen Bibliothek löst Abhängigkeiten zu denen, die im Userland des Hostsystems vorliegenden, auf (vgl. Hykes 2014).

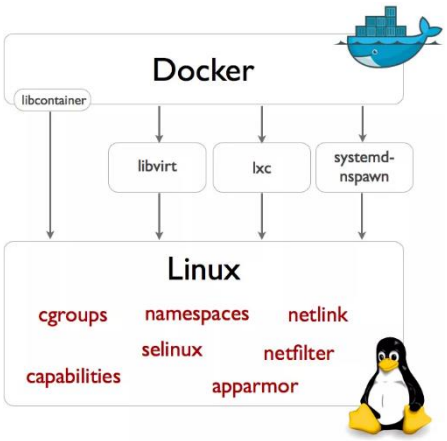

Abbildung 1: Docker Execution Driver[4]

[4] https://i0.wp.com/blog.docker.com/wp-content/uploads/2014/03/docker-execdriver-diagram.png?w=1024&ssl=1

Gesteuert wird Docker von dem Docker Daemon. Dieser muss auf dem Hostsystem als Prozess von dem Benutzer *root* gestartet werden, damit alle Funktionen bereitgestellt werden können. Docker teilt sich in drei elementare Komponenten auf. Das ist die Docker File, das Docker Image und der Docker Container selber.

Docker File
Die Docker File ist der Einstiegspunkt zur Erstellung von Docker Containern. In ihr sind alle Informationen in Form einer Liste enthalten. Diese Liste wird sequenziell abgearbeitet. Sie beinhaltet das jeweilige Betriebssystem auf dem das Docker Image aufbaut. Zudem enthält es Punkte, welche die zu installierende Software beschreiben. Zusätzlich dazu können weitere Befehle hinterlegt werden, welche beim Start des Docker Containers ausgeführt werden oder jene, die die grundsätzliche Konfiguration des Containers beeinflussen. (vgl. Docker Inc. File 2019)

Docker Image
Aus der Docker File wird ein Docker Image gebaut. Dieses kann mit einem Snapshot des Systems verglichen werden. Intern ist das Docker Image in Schichten aufgebaut. Jedoch ist es auch möglich, dass eine Schicht ein eigenständiges Image ist. Oftmals ist dies bei dem Betriebssystem der Fall. Dieses bildet die Basisschicht und kann bei der Erstellung eines weiteren Images geladen werden. Wenn ein weiteres Image erstellt wird, welches die gleiche Basisschicht nutzt, muss lediglich auf jene verwiesen werden. Dies hat den Vorteil, dass Ressourcen eingespart werden können, da auf mehrere gleiche Schichten verzichtet wird. (vgl. Docker Inc. Image 2019)

Docker Container
Aus dem Docker Image wird eine laufende Instanz eines Docker Containers erstellt. Wenn ein Docker Container ausgeschaltet und wieder neu gestartet wird, dann wird er wieder aus dem Image neu erstellt. Sollen persistente Änderungen an einem Docker Container vorgenommen werden, so müssen jene mithilfe des *commit*-Befehls gespeichert werden. Dadurch wird eine neue Schicht auf das vorhandene Image gelegt und somit ein neues Image mit den Änderungen erstellt. Es ist auch möglich, dass aus einem Image mehrere, parallel laufende Container erstellt werden.

3 Anforderungen an das System

In diesem Kapitel werden die Anforderungen an das zu erstellende System beschrieben. Diese beinhalten neben den inhaltlichen Anforderungen auch solche, die die Infrastruktur, die verwendeten Sprachen, aber auch Sicherheitstechnische Aspekte betreffen.

3.1 Architektur

Die zu erstellende Applikation soll als Webanwendung zur Verfügung gestellt werden. Dies soll den Wartungs- und Installationsaufwand gering halten und eine Unabhängigkeit von dem verwendeten Betriebssystem erzeugen. Bei einer solchen Webanwendung nach dem Client-Server-Modell liegt der rechenlastige Teil der Anwendung auf der Serverseite, während der Client nur einen gewöhnlichen Webbrowser benötigt. Dieser ist auf jedem stationären und mobilen Endgerät standardmäßig vorhanden. Um die Anwendung zu benutzen ist lediglich eine Internetverbindung erforderlich.
Die dafür notwendige serverseitige Infrastruktur ist mittels Docker bereitzustellen. Dabei ist die Infrastruktur außerdem nach dem *Three-Tier-Prinzip zu* erstellen. Durch die Nutzung dieser Schichtenarchitektur wird die Kohäsion der einzelnen Schichten erhöht, jedoch gleichzeitig die Kopplung gering gehalten. Dies hat den Vorteil, dass sowohl der Wartungsaufwand des Systems geringer wird und gleichzeitig die Austauschbarkeit der einzelnen Komponenten erhöht wird, ohne dabei andere Schichten zu tangieren.

Der Webserver soll mittels *nginx* realisiert werden.
Der Applikationsserver sollte mittels *FastCGI* an den Webserver angebunden werden und mit *PHP-FPM* ausgestattet sein.
Als Datenbankserver soll ein *MariaDB*-Server bereitgestellt werden.

Aus den infrastrukturellen Anforderungen ergeben sich die einzusetzenden Sprachen.
Die Implementierung der serverseitigen Logik erfolgt mittels *php.* Als Datenbankabfragesprache kommt *MySQL* zum Einsatz.
Für die Erstellung des Front Ends kommt *HTML, CSS* und *JavaScript* zum Einsatz.

3.2 Sicherheitsaspekte

„Web applications bring with them new and significant security threats. Each application is different and may contain unique vulnerabilities.“[5]

Sicherheit in Webanwendungen ist kein finaler Zustand. Es ist ein iterativer Prozess, welcher immer an die gegebene Situation angepasst werden muss. Daraus ergibt sich, dass die Sicherheitsmechanismen als Prozess verstanden werden kann, welcher entwicklungsbegleitend ist.
Eines der größten Sicherheitsprobleme bei Webanwendungen ist, dass die Anwendung mit einer unbekannten Menge und Varianz an Benutzereingaben umgehen können muss. Deshalb muss der Server alle übermittelten Daten als potenziell schädlich einstufen, um einen nicht autorisierten Zugriff abzuwehren und die Integrität der Daten, und jene selbst, zu schützen.
Daraus folgt, dass bereits in der Designphase die Anwendung auf die bekannten Sicherheitslücken beziehungsweise Sicherheitsrisiken vorbereitet wird.

Es sind nur die in den Unterkapiteln folgenden Sicherheitsaspekte zu berücksichtigen, da diese die größte Gefahr für die Sicherheit der Applikation darstellen. Auf die Absicherung der Server und dem Netzwerk wird in dem Rahmen dieser Arbeit verzichtet, da diese nicht dem Entwicklungsumfang einer Webapplikation zuzuordnen sind, sondern generell bei der Bereitstellung von IT-Diensten beachtet werden.

3.2.1 SQL-Injection

SQL-Injection ist eine Unterart von Injection-Angriffe. Diese Art des Angriffes stand in der Top 10 Liste der Risiken für die Anwendungssicherheit der OWASP 2017 an erster Stelle (vgl. OWASP 2017, S. 7). Mittels SQL-Injection wird versucht Daten zu stehlen oder deren Integrität zu gefährden. Auch kann es dazu dienen Informationen über die Datenbank zu erhalten oder im extremen Fall sogar die Kontrolle über den Datenbankserver zu bekommen. Um diese Ziele zu erreichen, wird mittels Konkatenation versucht Eingaben des Benutzers mit der originalen Datenbankabfrage eigene Parameter anzuhängen und dadurch auszuführen.

Um diese Art des Angriffes zu verhindern, ist ein geeigneter Weg zu wählen, um die Daten, deren Integrität und die Sicherheit des Datenbankservers zu schützen.

[5] Stuttard Pinto 2011, S. 38

3.2.2 Cross-Site-Scripting

Unter Cross-Site-Scripting, auch kurz *XSS* genannt, wird die Manipulation von Webseiten durch Benutzereingaben verstanden. Auch dieser Angriffsvektor ist in der Top 10 Liste der OWASP 2017 enthalten. Er liegt auf Platz 7 (vgl. OWASP 2017, S. 7). Auch diese Sicherheitslücke setzt darauf, dass Benutzereingaben nicht oder nicht richtig geprüft werden. Dadurch wird der Schadcode an den Browser des Benutzers zurückgesendet und kommt dort zur Ausführung. Diese Sicherheitslücke wird gezielt genutzt, um Benutzerdaten zu erhalten oder generell um Schadcode auszuführen.

Generell lässt sich Cross-Site-Scripting in drei Kategorien einteilen.

Reflected Cross-Site-Scripting

Bei reflected Cross-Site-Scripting, auch *non-persistent* Cross-Site-Scripting genannt, wird die Eingabe des Benutzers vom Server direkt wieder zurückgegeben. Enthält die Eingabe Schadcode, wird dieser durch den Browser interpretiert und gegeben falls ausgeführt. Dabei macht sich der Angreifer zu Nutze, dass viele Webseiten *HTTP-GET*- oder *HTTP-POST*-Parameter nutzen, um die Anzeige dynamisch zu generieren.
Diese Angriffsmethode wird als nicht persistent bezeichnet, da der Schadcode nicht auf dem Server gespeichert wird. Wird die Seite ohne die manipulierten Parameter aufgerufen, so verhält sich die Webseite normal und führt keinen Schadcode aus.

Persistent Cross-Site-Scripting

Das persistent Cross-Site-Scripting wird auch als *stored* beziehungsweise *dauerhaftes* Cross-Site-Scripting bezeichnet. Die Grundmethode des Angriffes ist dieselbe, wie bei reflected Cross-Site-Scripting. Es unterscheidet sich jedoch darin, dass der Server den Schadcode speichert. Dadurch wird bei einem erneuten Aufruf der Webseite der Schadcode erneut ausgeliefert und durch den Browser des Benutzers interpretiert und ausgeführt. Dies ist solange möglich, bis eine geeignete Validierung der Benutzereingaben Erfolg und die Ausgabe jener durch ein geeignetes Verfahren kodiert wird.

DOM-based Cross-Site-Scripting

Diese Variante des Cross-Site-Scripting wird auch als *lokales* XSS bezeichnet. Der Unterschied zu den zuvor genannten Varianten des Cross-Site-Scripting besteht darin, dass bei diesem Angriff der Web- beziehungsweise Applikationsserver nicht beteiligt ist. Er erfolgt über die Manipulation der URL. Ihr werden weitere Argumente hinzugefügt oder Vorhandene manipuliert. Nutzt nun ein Skript die übergebenen Argumente für die Ausgabe, so kommt es zu der Ausführung des Schadcodes.
Betroffen sind bei dieser Variante des Cross-Site-Scripting somit auch statische HTML-Seiten, welche eine Skriptsprache, beispielsweise JavaScript, nutzen.

Um das Sicherheitsrisiko zu senken, ist eine Methode zu entwickeln, welche die aufgeführten Arten von Cross-Site-Scripting unterbindet. Dabei soll es aber weithin möglich sein HTML-Attribute, welche für die Formatierung von Texten zuständig sind, einzugeben.

3.2.3 Command-Injection

Da die Applikation auch Shell-Befehle ausführt, ist diese Angriffsart zu verhindern. Bei dieser Art des Angriffes wird durch den Angreifer mittels Benutzereingaben versucht seine eigenen Befehle auf dem Hostsystem auszuführen oder Bestehende zu manipulieren oder jene zu erweitern.
Mit Comand-Injection versucht der Angreifer die Integrität des Web- beziehungsweise Applikationsservers zu gefährden oder weiteren Schadcode auszuführen, um die Kontrolle über den diesen Server zu erlangen.
Da die Befehle der Shell mit entsprechenden Rechten des Hostsystems ausgeführt werden, ist diese Art des Angriffes besonders gefährlich, denn dadurch kann nicht nur der Applikation geschadet werden, sondern der gesamten Infrastruktur.

Die Comand-Injection gehört zu der Familie der Injection-Angriffe und ist deshalb auch dem ersten Platz der Liste der OWASP zuzuordnen (vgl. OWASP 2017, S. 7).

Durch die Gefährlichkeit und die Häufigkeit solcher Angriffe ist ein geeignetes Verfahren zu nutzen, um die Benutzereingaben zu überprüfen und zu validieren, bevor diese den Shell-Befehlen hinzugefügt werden.

3.3 Funktionsumfang

In den folgenden Unterkapiteln wird der Funktionsumfang des zu programmierenden Scans genauer erläutert.

3.3.1 Import von Ansprechpartnerdaten

Damit einem Scan ein Ansprechpartner zugeordnet werden kann, soll es möglich sein eine CSV-Datei mit Ansprechpartnerdaten hochzuladen. Als Trennzeichen wird ein Semikolon verwendet. Strings dürfen nicht in Anführungszeichen eingefasst werden. Diese CSV-Datei beinhaltet keine Überschriften. Die erste Zeile repräsentiert den ersten Ansprechpartner.
Leere Zeilen werden nicht beachtet.
Die Reihenfolge der Daten ist folgendermaßen:

1. Feld:
    - Datentyp: Integer
    - Wert: Eindeutige ID zur Identifikation des Ansprechpartnerdatensatzes.
2. Feld:
    - Datentyp: String
    - Wert: Nachname des Ansprechpartners
3. Feld:
    - Datentyp: String
    - Wert: Vorname des Ansprechpartners.
4. Feld:
    - Datentyp: String
    - Wert: Gültige E-Mailadresse des Ansprechpartners.
5. Feld:
    - Datentyp: String
    - Wert: Domain beziehungsweise Subdomain, für die der Ansprechpartner zuständig ist.
6. Feld:
    - Datentyp: String
    - Wert: Zuständiger Bereich des Ansprechpartners.
    - Mögliche Ausprägungen:
        - i für inhaltlich
        - t für technisch
        - u für unbekannt

3.3.2 Überprüfungsumfang

Um das Ziel des Scans zu bestimmen, soll es möglich sein sowohl eine (Sub-)Domain einzugeben, als auch eine IP-Adresse.
Dem Benutzer des Programms soll es möglich sein den Überprüfungsumfang selber festzulegen.

3.3.2.1. Port Scan

Mittels eines geeignet Portscanners soll es möglich sein die *well-known* Ports zu überprüfen. Dieser soll die auffindbaren Ports analysieren und so Informationen über deren Status, deren Protokoll und dem Dienst hinter dem Port sammeln.
Wenn ein SSH-Port gefunden wurde, dann soll es dem Benutzer möglich sein diesen hinsichtlich der Authentifizierungsmethoden zu überprüfen. Dabei ist von besonderer Bedeutung, ob der SSH-Dienst eine Authentifizierung mittels Passwort ermöglicht.

3.3.2.2. SSL Labs

SSL Labs ist ein kostenloser Dienst für die Überprüfung von SSL Servern und deren ausgelieferten Zertifikaten. Dieser Dienst bietet eine Schnittstelle an, um die Überprüfung auszuführen.
Die Schnittstelle von SSL Labs soll angesprochen und die gelieferten Ergebnisse ausgewertet werden.
Dabei ist die Schnittstelle so anzusprechen, dass die Ergebnisse nicht auf der Internetseite von SSL Labs veröffentlicht werden. Außerdem soll bei jeder Anfrage über die Schnittstelle die Überprüfung auf der Seite von SSL Labs neu gestartet werden und nicht ein bereits ermitteltes Ergebnis genutzt werden.

3.3.2.3. Überprüfung des Impressums

Der Internetauftritt soll auf das Vorhandensein eines Impressums überprüft werden. Dazu sollen gängige Dateinamen, Dateiendungen und Pfade geprüft werden. Wenn ein Impressum gefunden wurde, dann ist der Link zu Jenem zu speichern.
Zusätzlich soll versucht werden zu überprüfen, ob Angaben zum Verantwortlichen des Internetauftrittes gefunden werden können.

3.3.2.4. Überprüfung der Datenschutzseite

Neben dem Impressum soll auch das Vorhandensein einer Datenschutzseite geprüft werden. Dazu sollen, wie auch beim Impressum, gängige Dateinamen, Dateiendungen und Pfade geprüft werden. Wenn eine Datenschutzseite gefunden wurde, dann soll der Link zu der Seite gespeichert werden.
Wenn eine Datenschutzseite gefunden wurde, dann soll versucht werden die Seite auf die Angabe eines Datenschutzbeauftragten zu überprüfen.

3.3.3 Export der Ergebnisse

Es soll eine Funktion entwickelt werden, welche die Daten aller durchgeführten Scans in einer Datei bereitstellt. Das zu wählende Format ist JSON.

4 Einschränkungen des Programms

Bei der Durchführung des Scans kann es unter Umständen zu Komplikationen kommen. So können Firewalls durch ihre Funktionsweise verhindern, dass der Server beziehungsweise der Internetauftritt überprüft werden kann. Dies kann zu verfälschten Ergebnissen führen.

Ebenso könnten sogenannte Blocker dazu führen, dass der Scan durch das Sperren von IP-Adressen abbricht und somit nicht vollständig durchgeführt werden kann. Ebenso kann es durch den Einsatz von Blockern auf der zu untersuchenden Seite dazu kommen, dass nicht der komplette Portumfang geprüft werden kann. Außerdem ist es möglich, dass das Programm als Maschine erkannt wird und somit an der weiteren Untersuchung des Internetauftritts gehindert werden kann. Dies ist bei sogenannte Captcha-Abfragen der Fall.

Des Weiteren können Internetauftritte nicht überprüft werden, wenn eine Authentifizierung von Nöten ist, um auf den Inhalt zugreifen zu können.

Auch das Auslösen von Alarmmaßnahmen auf der Seite des untersuchten Systems kann die vollständige und richtige Ausführung des Scans behindern beziehungsweise beeinflussen. So können Intrusion Detection und Intrusion Prevention Systeme den Scan als bösartig einstufen. Ebenfalls ist es möglich, dass ein solches System die Ausführung des Scan dahingegen beeinflusst, dass es den Datenverkehr auf ein HoneyPot weiterleitet. Dadurch können falsche Ergebnisse zustande kommen.

Um solche Maßnahmen auf den zu untersuchenden System zu verhindern, ist der Besitzer jenes System im vorab zu informieren und eine Ausführung des Scans ist nicht ohne schriftliche Zustimmung des Besitzers erlaubt.
Diese schriftliche Einwilligung ist von Bedeutung, da der Scan als illegale Handlung zur Vorbereitung einer Straftat nach Paragraf 202a und 202b des Strafgesetzbuches betrachtet werden kann. Dieser Teil des Gesetzes befasst sich mit dem Ausspähen und Abfangen von Daten.
Der Scan kann rechtlich als eine Art des *Penetration Testings* verstanden werden. Dieses hat derzeit keine strafrechtlichen Folgen, da die Sicherheit der IT-Systeme im Vordergrund steht (vgl. Gaykan 2011, S. 162f.).

Das Programm ist so zu konstruieren, dass es zu keinen Zeitpunkt die Sicherheit oder die Erreichbarkeit des zu untersuchenden Systems gefährdet.
Zusätzlich ist die Ausführung von Angriffen auf das System nicht Inhalt des Programmes. Es dient lediglich zum Sammeln von Informationen über den Zustand des Internetauftrittes und der dazugehörigen IT-Systeme.

5 Funktionsweise des Systems

Dieses Kapitel befasst sich mit der Funktionsweise des Systems. Es beinhaltet den Aufbau der Infrastruktur und der Datenbank. Auch wird darauf eingegangen, wie die Anforderungen an die programmseitige Sicherheit des Systems umgesetzt wurden. In einem weiteren Unterkapitel werden der Aufbau des Scans und die Ermittlung der Benotung genauer beschrieben. Am Ende wird ein Blick auf die Darstellung der Scanergebnisse und deren Bereitstellung für externe Systeme geworfen.

5.1 Infrastruktur

Im Folgenden wird näher auf die im Laufe dieser Arbeit erstellten infrastrukturellen Komponenten eingegangen. Die Infrastruktur wurde auf einem Linux Betriebssystem der Distribution *Fedora*[6] in Version 29 entwickelt.

Für die Bereitstellungen der einzelnen Docker Images und den damit verbundenen Docker Containern wurde eine Docker-Compose-Datei entwickelt. Diese ist wie folgt aufgebaut:

[6] https://getfedora.org

```
version: "3.1"
services:

   mariadb:
     image: mariadb:10.4
     container_name: bachelorfcd-mariadb
     working_dir: /application
     volumes:
       - .:/application
       -                    ./phpdocker/mariadb/complianceChecker.sql:/docker-entrypoint-
initdb.d/complianceChecker.sql
     environment:
       - MYSQL_ROOT_PASSWORD=<PASSWORD>
       - MYSQL_DATABASE=complianceChecker
       - MYSQL_USER=webUser
       - MYSQL_PASSWORD=<PASSWORD>
     ports:
       - "1340:3306"

   webserver:
     build: phpdocker/nginx
     container_name: bachelorfcd-webserver
     working_dir: /application
     volumes:
        - ./src:/var/www:z
        - .:/application:z
        - ./phpdocker/nginx/nginx.conf:/etc/nginx/conf.d/default.conf:z
     ports:
      - "1337:80"
   php-fpm:
     build: phpdocker/php-fpm
     container_name: bachelorfcd-php-fpm
     working_dir: /application
     volumes:
       - ./src:/var/www:z
       - .:/application:z
       - ./phpdocker/php-fpm/php-ini-overrides.ini:/etc/php/7.3/fpm/conf.d/99-overrides.ini:z
```

Zu sehen ist, dass drei verschiedene Services bereitgestellt werden: Ein MariaDB-Container, ein nginx-Container und ein php-fpm-Container. Die Containernamen setzen sich aus zwei Teilen zusammen. Als ersten Teil haben sie einem gemeinsamen Präfix zusammen. Dieser lautet „bachelorfcd-“. Im Anschluss folgt die Bezeichnung des jeweiligen Services.

Der Datenbankcontainer hat den Namen „bachelorfcd-mariadb“ und wurde aus dem MariaDB-Image mit der Version 10.4 erstellt.
Nach dem Start des Datenbankcontainers wird eine SQL-Datei ausgeführt, damit die Datenbank mit den benötigten Tabellen befüllt wird.

Der Container, welcher den Webserver beinhaltet, basiert auf dem nginx:alpine-Image. Dieses wird in der entsprechenden Docker File definiert. Er trägt den Namen „bachelorfcd-webserver“.
Unter *volumnes* wird dem Webserver der Quellcode des Programmes bereitgestellt und im Docker Container auf den Pfad */var/www/* verlinkt. Dies hat den Vorteil, dass Änderungen am Quellcode auf dem Hostsystem direkt im Container zur Verfügung stehen, ohne dass dieser neugestartet beziehungsweise neu erstellt werden muss.
Zusätzlich wird dem Webservercontainer die nginx-Konfigurationsdatei zur Verfügung gestellt. Dieses ist besonders wichtig, da Änderungen an der Standardkonfiguration vorgenommen wurden. Diese Änderungen wurden vollzogen, da der Webserver bei der Ausführung des Scans länger auf eine Antwort des Applikationsservers warten muss, als standardmäßig konfiguriert.

Der Applikationsserver basiert auf dem aktuellsten phpdockerio/php73-fpm-Image und hat den Namen „bachelorfcd-php-fpm“. In diesem Container ist PHP in der Version 7.3 vorhanden. Wie auch beim Webservercontainer, wird diesem Docker Container der Quellcode mittels Verlinkung zur Verfügung gestellt.
Eine Besonderheit des Docker Containers des Applikationsservers ist die Docker File. In dieser werden weitere PHP-Module installiert, sowie der Portscanner *nmap.* Dieser wird für die Ausführung des Portscans benötigt.

Datenbankstruktur

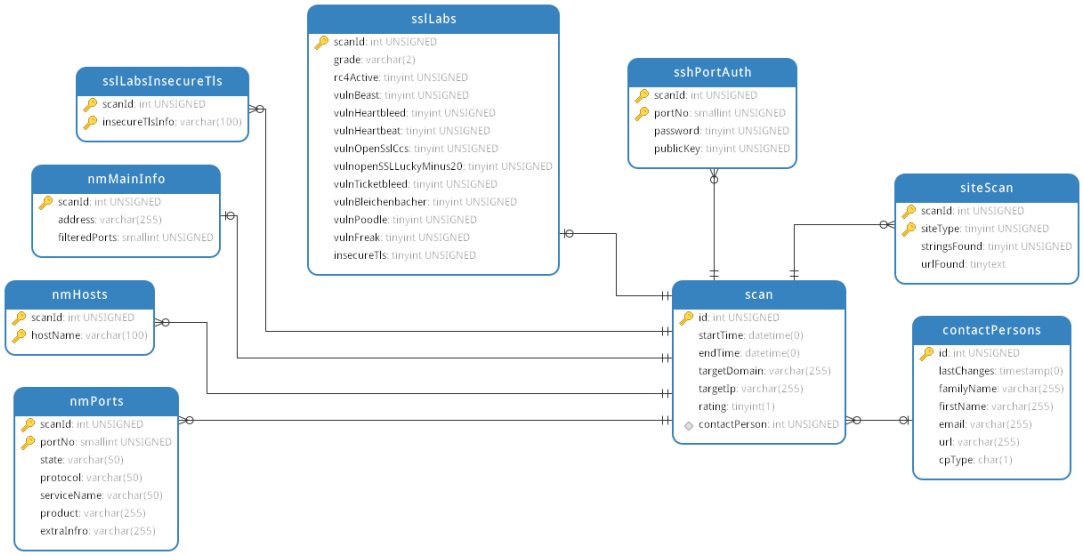

Abbildung 2: ER-Modell der Datenbank

Die Datenbank wurde in der Dritten Normalform erstellt.
Wie in Abbildung 2 zu sehen, ist der zentrale Bestandteil der Datenbank die Entität „scan“. Diese beinhaltet alle allgemeinen Information eines Scans und ist das führende System zur Erstellung der Fremdschlüsselbeziehungen. Jede andere Entität hält über die ID des Scans eine Beziehung zu der Entität „scan“. Eine Ausnahme bildet hierbei die Entität „contactPerson“. Jene Entität beinhaltet die Ansprechpartner Daten, die nicht bei der Ausführung eines Scans ermittelt werden. Sie werden genutzt, um dem Scan einen Ansprechpartner zuzuordnen.
Alle Entitäten, die logisch gesehen zusammengehören, haben einen gemeinsamen Präfix.

5.2 Sicherheit

Alle an den Applikationsserver gesendeten Daten oder Dateien durchlaufen eine Sicherheitsüberprüfung. Dies ist von Nöten, um die Sicherheitsanforderung an das System umzusetzen.

Dateien

Übermittelte Dateien werden auf zwei Arten überprüft.
Zur Überprüfung der Korrektheit einer Datei steht die Klasse *SecureFileUpload* zur Verfügung. Diese überprüft einerseits, ob die Dateiendung korrekt ist. Auf der anderen Seite überprüft sie den MIME-Type der Datei.
Für das entwickelte Programm wurden Dateien ermöglicht, welche die Endung .csv tragen und den MIME-Type text/csv haben. Entspricht eine Datei nicht den Anforderungen wird diese abgelehnt.

Neben dieser Überprüfung wird auch der Inhalt der Datei überprüft. Dafür kommt die Klasse *SecurityFilter* zum Einsatz. Diese bietet Methoden zur Überprüfung von Werten an.

Werteüberprüfung
Wie bereits oben beschrieben, bietet die Klasse *SecurityFilter* mehrere Methoden für die Überprüfung von Werten an.

Die Überprüfung von Integer Werten erfolgt durch ein Typecast. Durch den Aufruf der entsprechenden Methode wird der übergebene Wert in ein Integer Wert umgewandelt. Dies hat den Vorteil, dass einerseits eine höhere Typsicherheit vorliegt, andererseits kann nach dem Typecast kein Schadcode mehr im Wert vorliegen. Ein möglicher Integer Overflow kann an dieser Stelle vernachlässigt werden, da eine Plausibilitätsprüfung des Wertes an dieser Stelle nicht vollzogen wird.

Für die Überprüfung von übermittelten Domainnamen, E-Mailadressen oder IP-Adressen stehen weitere Methoden bereit. Diese validieren den Wert anhand der PHP-eigenen Funktion *filter_var* und dem entsprechenden Filterflag. Wenn ein Wert nicht der Vorgabe entspricht, wird dies als möglicher Angriff gewertet und die Programmlogik ersetzt den Wert durch *NULL*.

```php
static private function createDomDoc($inputText){
    libxml_use_internal_errors( use_errors: true);
    $dom = new DOMDocument( version: '1.0');
    self::fixAmps( &html: $inputText);
    $dom->loadHTML(mb_convert_encoding( str: "<div>".$inputText."</div>",
                to_encoding: 'HTML-ENTITIES',
                from_encoding: 'UTF-8'),
            options: LIBXML_HTML_NOIMPLIED | LIBXML_HTML_NODEFDTD);
    $secLevel = (count(libxml_get_errors()) > 0) ? self::SECURITY_LEVEL['high'] : null;
    return array(
        'dom' => $dom,
        'secLevel' => $secLevel
    );
}
```

Abbildung 3: Erstellung des DOMDocument-Objektes

Die Überprüfung von sonstigen Texten erfolgt mittels der entwickelten Methode *validateString.* Diese filtert den übergebenen Text und entfernt mögliche Cross-Site-Scripting-Versuche. Die Filterung findet in PHP statt. Die folgend genannten Klassen beziehungsweise Objekte und Methoden sind standardmäßig in PHP enthalten.
Für diese Filterung wird der übergebene Text in ein *DOMDocument* geladen. Wenn bei diesem Vorgang ein interner Fehler auftritt, dann wird das als Angriff gewertet. Die Überprüfungsroutine läuft zwar normal weiter, jedoch werden am Ende alle HTML-Tags entfernt und der verbliebene Text mittels *htmlentities* kodiert. Dieser Schritt wird in der obigen Abbildung dargestellt.

Das erstellte *DOMDocument*-Objekt wird anschließend einem *DOMXPath*-Objekt übergeben und mittels einer *XPath*-Query wird eine Liste aller enthaltenen Nodes erstellt. Diese Nodeliste wird im weiteren Verlauf gefiltert. Mittels einer White List ist festgelegt, welche HTML-Tags zugelassen sind. Alle anderen HTML-Tags werden aus der Liste entfernt.
Die übrig gebliebenen Nodes werden danach auf ihre Attribute überprüft. Auch hier kommt eine White List zum Einsatz. Alle nicht erlaubten Attribute werden entfernt.
Am Schluss werden die Werte der Attribute überprüft. Mittels einer Black List werden auch die letzten möglichen Cross-Site-Scripting-Vektoren entfernt.
Die gefilterte Nodeliste wird wieder in ein String umgewandelt und dieser wird zurückgegeben.

Shell Argumente
Auch wenn alle Argumente, die einem Shell-Befehl übergeben werden, bereits durch die obigen Funktionen überprüft wurden, werden sei durch die PHP-Funktion *escapeshellarg* kodiert. Mögliche Angriffe werden dadurch zusätzlich verhindert.

Datenbank
Die Datenbank ist auf der Programmseite durch die Verwendung von *Prepared Statements* geschützt. Für den Zugriff auf die Datenbank wird die PHP-eigene Klasse *PDO* verwendet. Diese bietet bei der Konsequenten Anwendung von *Prepared Statements* nicht nur Schutz vor möglicher SQL-Injection, sondern abstrahiert auch den Zugriff auf die Datenbank. Bei einem Wechsel der Datenbank auf eine andere SQL-basierte Datenbank sind die möglichen Anpassungen geringer.

5.3 Umsetzung des Funktionsumfanges

In diesem Unterkapitel wird erläutert, wie der geforderte Funktionsumfang umgesetzt wurde.

Ansprechpartnerdaten
Über den Menüpunkt „Einstellungen“ gelangt der Benutzer auf die Einstellungsseite für Ansprechpartner. Hier kann der Benutzer entscheiden, ob er sich die Liste der bereits importierten Ansprechpartner anschauen möchte oder ob er eine erste oder weitere Ansprechpartnerdatei zum Importieren hochladen möchte.

Überprüfungsumfang

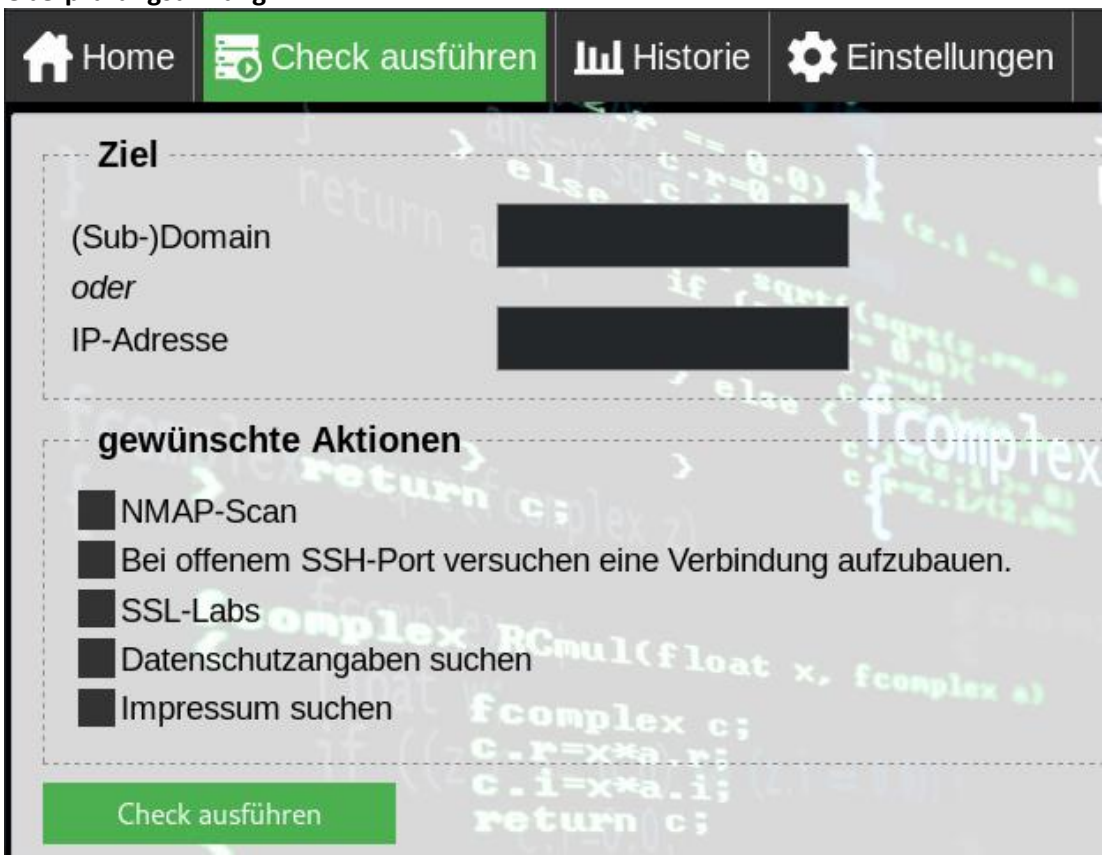

Abbildung 4: Startmaske eines Scans

Wie in Abbildung 4 zu sehen, kann der Benutzer zum Starten eines Scans eine (Sub-)Domain oder eine IP-Adresse eingeben. Ebenso kann er den Umfang der Überprüfung anhand der „gewünschten Aktionen“ bestimmen.
Für den Überprüfungsumfang sind zwei Abhängigkeiten vorhanden. Soll versucht werden mit einem SSH-Port eine Verbindung aufzubauen, so ist auch ein Port Scan (NMAP-Scan) erforderlich. Die andere Abhängigkeit betrifft die Anfrage über die SSL-Labs-API. Um diese zu starten ist der Benutzer gezwungen eine Domain einzugeben. Diese ist ein Pflichtfeld in der Schnittstellenbeschreibung.

Für die Durchführung des Portscans wurde sich für den Portscanner *nmap* entschieden. Dieser ist für viele Linux Distributionen verfügbar und in vielen Paketverwaltungssystemen vorhanden. Überprüft werden die Ports 0 bis 1023.

Zusätzlich können auch weitere Prüfskripte mit *nmap* über die *Nmap Scripting Engine (NSE)*[7] ausgeführt werden. Ein solches Skript wird für den SSH-Verbindungsaufbau verwendet. Dafür kommt das Skript *ssh-auth-methods* zum Einsatz.
Um die Ergebnisse der Scans mit *nmap* zu verwenden wird eine XML-Datei der Ergebnisse erzeugt. Diese wird im Anschluss durch die Programmroutine eingelesen und verarbeitet.

Für die Entwicklung der Schnittstelle zu SSL-Labs wurde die öffentliche Schnittstellenbeschreibung für die API v3 in Version v.1.35.x verwendet[8].
Folgende Werte werden über die Schnittstelle abgefragt:
- SSL Labs Einstufung (Grade)
- Ist RC4 aktiv?
- Ist ein Beast-Angriff möglich?
- Ist ein Heartbleed-Angriff möglich?
- Ist ein openSSL CCS-Angriff möglich?
- Ist ein openSSL Lucky Minus 20-Angriff möglich?
- Ist ein Ticketbleed-Angriff möglich?
- Ist ein Bleichenbacher-Angriff möglich?
- Ist ein Poodle-Angriff möglich?
- Ist ein Freak-Angriff möglich?
- Werden unsichere SSL/TLS-Versionen verwendet?

Die Ausführungsdauer des Scans erhöht sich immens, wenn die SSL-Labs Schnittstelle angesprochen wird. Dies liegt an der hohen Ausführungsdauer auf der Seite der Schnittstelle. Die Überprüfung einer Domain dauert mehrere Minuten. Die Überprüfungsroutine startet den SSL-Labs-Scan und ruht dann für 20 Sekunden. Nach Ablauf dieser Zeit fragt es die Schnittstelle erneut an und wertet die Anfrage aus. Wenn die Überprüfung seitens SSL-Labs noch nicht abgeschlossen ist, wird erneut gewartet. Dies wird solange wiederholt, bis ein vollständiges Ergebnis vorliegt. Das vollständige Ergebnis wird ausgewertet und abgespeichert.

[7] https://nmap.org/man/de/man-nse.html
[8] https://github.com/ssllabs/ssllabs-scan/blob/master/ssllabs-api-docs-v3.md

```php
public function startScan(){
    $curl = curl_init();
    curl_setopt($curl, option: CURLOPT_RETURNTRANSFER, value: TRUE);
    curl_setopt($curl, option: CURLOPT_TIMEOUT, value: 30 );
    curl_setopt($curl, option: CURLOPT_FOLLOWLOCATION, value: true);
    curl_setopt($curl, option: CURLOPT_POSTREDIR, value: 3);
    foreach($this->subDirs as $subDir){
        foreach($this->sites as $site){
            foreach($this->endings as $ending){
                if(($subDir == '/' && $site == '') || ($site == '' && $ending != '')){
                    continue;
                }
                curl_setopt($curl, option: CURLOPT_URL, value: $this->target.$subDir.$site.$ending);
                $ret = curl_exec($curl);
                if(curl_getinfo($curl, opt: CURLINFO_HTTP_CODE) == 200){
                    $this->pageFound = $this->target.$subDir.$site.$ending;
                    $this->searchResult($ret);
                    return TRUE;
                }
            }
        }
    }
    curl_close($curl);
    return FALSE;
}
```

Abbildung 5: : Routine zum Aufsuchen der Impressums-/Datenschutzseite

Die lange Ausführungsdauer dieses Teils der Überprüfungsroutine hatte zur Folge, dass die Konfiguration des Webservers angepasst werden musste.

Zur Überprüfung des Vorhandenseins einer Impressums- oder Datenschutzseite wird die gleiche Routine genutzt. Diese ist in Abbildung 5 zu sehen. Für die individuellen Verzeichnisse und Dateinamen der jeweiligen Überprüfung stehen Methoden bereit. Diese erweitern die in der Routine vorhandenen.
Die Suche der jeweiligen Seite wird mittels eines *cURL-Requests* durchgeführt, da dieser auch Weiterleitungen des untersuchten Servers unterstützt. Erlaubt sind bei diesem Scan drei Weiterleitungen. Wenn der HTTP-Status-Code 200 geliefert wird, dann wurde die gesuchte Seite gefunden. Dieser Pfad beziehungsweise der vollständige Link wird abgespeichert. Das Ergebnis des *cURL-Requests* ist der repräsentiert das ausgelieferte HTML-Gerüst. Dieses Gerüst wird im Anschluss in der Methode *searchResult* nach dem möglichen vorhandenen Verantwortlichen (bei der Suche nach dem Impressum) oder dem Datenschutzbeauftragten (bei der Suche nach der Datenschutzseite) durchsucht. Dies erfolgt durch einen Textvergleich.

5.4 Darstellung und Bereitstellung der Ergebnisse

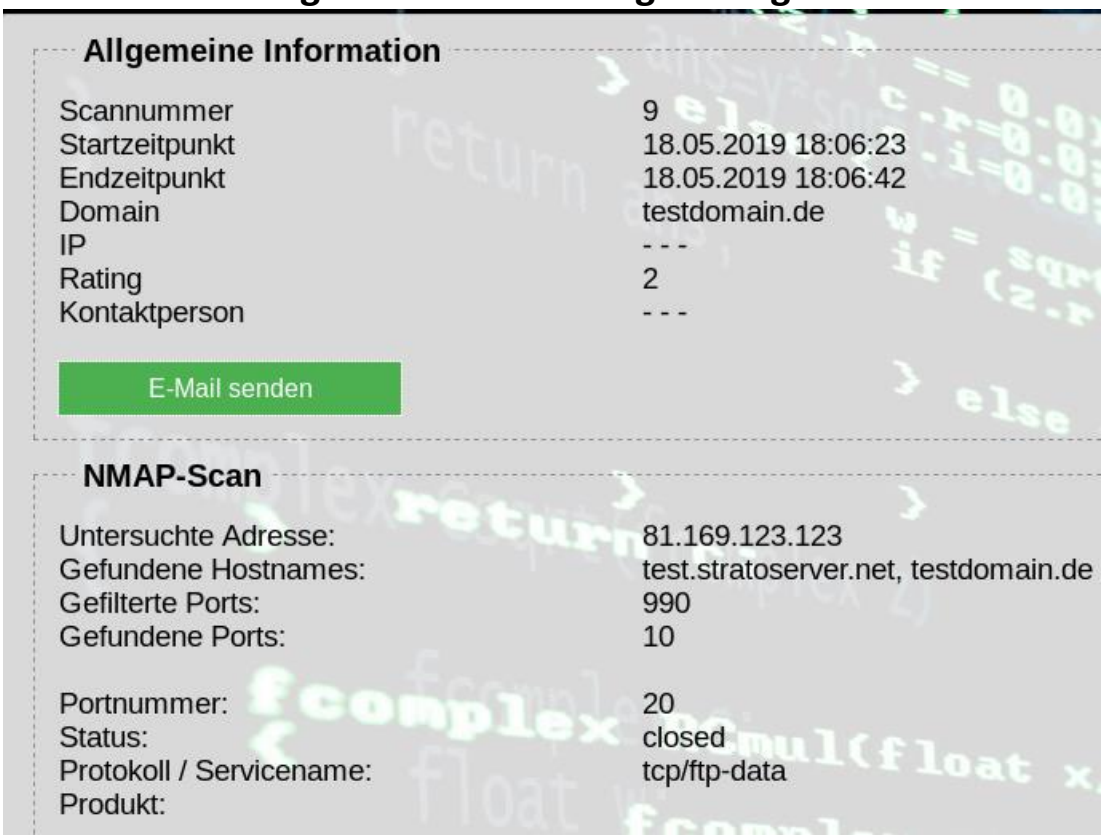

Abbildung 6: Ergebnisse eines Scans im Detail

Wenn der Scan erfolgreich abgeschlossen wurde, dann wird der Benutzer an eine Seite weitergeleitet, welche die Ergebnisse im Detail anzeigt. Diese Detailseite zeigt Abbildung 6. Unter der Gruppe „Allgemeine Informationen“ werden Informationen, wie der Startzeitpunkt und der Endzeitpunkt des Scans, die untersuchte Domain beziehungsweise IP-Adresse, das Rating des Scans, sowie die Kontaktperson, wenn vorhanden, angezeigt.
Im weiteren Verlauf der Seite werden die jeweiligen Details der einzelnen Scankomponenten angezeigt. In Abbildung 6 sind dies die Details des NMAP-Scans.

Des Weiteren ist es dem Benutzer möglich aus dem System heraus eine E-Mail zu versenden. In Abbildung 6 ist dies über den „E-Mail senden“-Button ersichtlich.
Wenn der Benutzer diese Funktion nutzt, dann wird ein Betriebssystemfenster zum Versenden von E-Mails geöffnet. Diesem neuen Fenster werden alle wichtigen Daten des Scanergebnisses übergeben. Dazu zählen unter anderem die allgemeinen Informationen, aber auch die Auffälligkeiten der jeweiligen Einzelscans. So werden beispielsweise die gefundenen Schwachstellen des SSL-Labs-Scans dargestellt. Auch zählen dazu die Daten aus

der Suche nach dem beziehungsweise der Durchsuchung des Impressum oder der Datenschutzseite.
Das interne Rating wird aus den einzelnen Scanergebnissen ermittelt.
Ein Rating der Stufe „1“ kann erreicht werden, wenn keine Auffälligkeiten gefunden werden konnten. Dies ist das beste Ergebnis.
Das Rating der Stufe „2“ wird erreicht, wenn der Scan keinen Verantwortlichen im Impressum oder keinen Datenschutzbeauftragten auf der Datenschutzseite finden konnte. Ebenso wird diese Stufe erreicht, wenn SSL-Labs eine Benotung von „B+“ oder „B“ vergibt.
Stufe „3“ des Ratings wird ermittelt, wenn SSL-Labs eine Benotung von „B-“ vergibt. Ebenfalls kann diese Stufe erreicht werden, wenn keine Datenschutzseite oder kein Impressum gefunden werden konnte.
Das schlechteste Rating der Stufe „4“ kann erreicht werden, wenn Sicherheitslücken von SSL-Labs gefunden werden oder deren Rating eine Benotung von „C“ oder schlechter ergibt. Zu dieser Stufe des Ratings kann auch führen, wenn bei der Untersuchung des SSH-Ports die Passwortauthentifizierungsmöglichkeit gefunden wurde.

Die Einstufung des Scans in ein Rating ist nicht ausschließlich zu Klassifizierung des Scans gedacht, sondern auch für die Übersicht aller bisherigen Scans. Wie in Abbildung 7 zu sehen, ist jeder Stufe eine Farbe zugeordnet. Dadurch kann der Benutzer direkt erkennen, welche Scans im Detail betrachtet werden sollten oder bei welchen Scan der Webseitenbetreiber über mögliche Sicherheitslücken oder Fehlkonfigurationen informiert werden sollte.

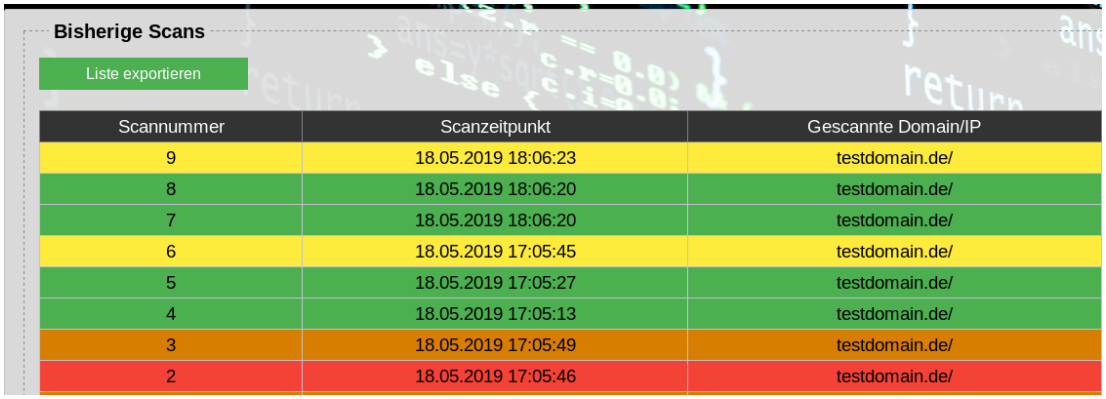

Abbildung 7: Ausschnitt der Übersicht der bisherigen Scans

Über den Button „Liste exportieren“ wird dem Benutzer eine JSON-Datei bereitgestellt, welche alle Scans, sowie deren Details, enthält.

6 Fazit und Ausblick

Im Verlauf dieser Bachelorarbeit wurde erfolgreich ein webbasiertes Programm entwickelt, welches sich für den produktiven Einsatz eignet. Die im dritten Kapitel definierten Anforderungen an die Infrastruktur, der Sicherheit und der Überprüfungsroutine wurden umgesetzt und um die E-Mail-Funktion erweitert. Zusätzlich sind weitere Ideen zur Erweiterung und des Scans aufgekommen. Diese werden in den folgenden Unterkapiteln behandelt.

Die Nutzung des Scans kann auch ohne die Oberfläche erfolgen. Für die Ausführung eines Scans ist lediglich ein POST-Request auf das Ausführungsskript notwendig. Auch die Ergebnisse lassen sich per GET-Request als JSON ermittelt lassen.

Bei der Durchführung einer Überprüfung fiel die lange Ausführungsdauer negativ auf, wenn die SSL-Labs-API angesprochen wird. Um diese zu minieren, wäre es denkbar, dass dieser Teil der Überprüfungsroutine umgebaut wird. So könnte weiterhin die Anstoßung der Überprüfung von SSL-Labs innerhalb des Scans durchgeführt werden. Die Abfrage des Ergebnisses könnte in einem separaten Skript erfolgen und durch einen Cronjob ausgeführt werden. Dazu sind auch Änderungen an der Datenbankstruktur und der Ermittlung des Ratings notwendig. Auch könnte es die Gesamtlänge des Scans erhöhen. Dies ist abhängig von der Ausführungshäufigkeit des Cronjobs.

Eine weitere Möglichkeit zur Verbesserung des Scans ist die Nutzung eines anderen Portscanners. Ob ein anderer Portscanner die Ergebnisse verbessert, gilt es zu Validieren. Der Umbauaufwand des Quellcodes und der Datenbank hält sich in Grenzen, da intern eine solche Möglichkeit bereits vorgesehen ist, denn die interne Datenübergabe erfolgt schon über Datenobjekte.

Des Weiteren wäre ein Einsatz eines PHP-Crawlers mit JavaScript Unterstützung denkbar. Dies könnte die Routine zum Auffinden der Impressums- beziehungsweise Datenschutzseite verbessern.

6.1 Erweiterung der Überprüfungsroutine

Eine Erweiterungsmöglichkeit der Überprüfungsroutine ist, dass der Port Scan um die Erkennung des Betriebssystems des zu untersuchenden Servers erweitert wird. Dies würde die Last auf dem untersuchten System erhöhen und damit dessen Erreichbarkeit möglicherweise Einschränken.
Zudem könnte der Port Scan um ein Abgleich mit einem öffentlichen CVE-Verzeichnis erweitert werden, um Sicherheitslücken Betriebssystemversion und der Dienste eines Ports zu erkennen. Dadurch würde das Scanergebnis noch detaillierter ausfallen und zu einer Besserung des Ergebnisses führen.

Generell könnte die Überprüfungsroutine um eine Funktion erweitert werden, welche einen Reputationsdienst anspricht und die mögliche Gefährdungslage des Internetauftrittes und der dazugehörigen Infrastruktur zu ermitteln.

Wenn die fehlende Übereinstimmung von SSL-Zertifikat und Domain ignoriert werden kann, so ist es denkbar die SSL-Labs-API mittels einer Domain anzusprechen, welche aus einer IP-Adresse über das *Reverse-DNS*-Verfahren ermittelt wurde. Dies birgt aber die Gefahr, dass das Ergebnis ungenauer wird, als es jetzt der Fall ist.

Eine letzte aufgekommene Erweiterungsmöglichkeit, wäre die Einbindung eines Schwachstellenscanner. Beispielsweise *OpenVAS*[9]*.* Ein solcher Schwachstellenscanner sollte mittels einer Schnittstelle angebunden werden, damit dieser austauschbar bleibt, um zu validieren, welcher sich für diese Erweiterung am besten eignet.
Die Gefahr bei einem solchen Schwachstellenscanner ist, dass dieser eine hohe Belastung für das Netzwerk und das zu untersuchende System bedeutet. Deshalb sollte diese Erweiterung nicht auf ein Produktivsystem angewendet werden.

6.2 Spezifische Exporte der Ergebnisse

Eine spezielle Art der Erweiterung wäre die Bereitstellung benutzerspezifischer Exporte der Scanergebnisse. So könnte der Benutzer die gewünschten Daten der Scans selber definieren und so an die Folgesysteme anpassen. Auch eine Eingrenzung der zu exportierenden Scans wäre über die Start oder Endzeit des Scans denkbar.

Eine weitere Möglichkeit zur Erweiterung der Scans wäre es, dass ein anderes Datenformat im Export angeboten wird. Beispielsweise das CSV-Format.

[9] http://openvas.org/

7 Quellenverzeichnis

Boerse 2019
Verlauf Volkswagen Vz Aktie. boerse.de
Online verfügbar:
https://www.boerse.de/chart-tool/Volkswagen-Vz-Aktie/DE0007664039
Abruf: 2019-03-01

Docker Inc. File 2019
DOCKER INC: *Docker reference.* 2019
Online verfügbar:
https://docs.docker.com/engine/reference/builder/#dockerignore-file
Abruf: 2019-04-19

Docker Inc. Image 2019
DOCKER INC: *About images, containers, and storage drivers.* 2019
Online verfügbar:
https://docs.docker.com/v17.09/engine/userguide/storagedriver/imagesandcontainers/
Abruf: 2019-04-19

Fobrun 1996
FOBRUN, Charles J.: *Reputation: Realizing Value from the Corporate Image.* Watertown, Massachusetts, USA: Harvard Business Review Press, 1996. - ISBN: 9780875846330

Goethe 1829
GOETHE, Johann Wolfgang: *Wilhelm Meisters Wanderjahre,* 1821; vollst. Fassung 1829. 1. Buch, 7. Kap.

Gaykan 2011
GAYKAN, Sandro: *Cyberwar: Das Internet als Kriegsschauplatz.* München: OpenSourcePress, 2011. - ISBN: 978-3-941841-23-9

Haake 2017
HAAKE, Henning: *So reagieren deutsche Unternehmen auf IT-Sicherheitsvorfälle – Angst um das Image.* Handelsblatt.com, 2017.
Online verfügbar:
https://veranstaltungen.handelsblatt.com/cybersecurity/it-sicherheitsvorfaelle-2017/
Abruf: 2019-03-01

Hykes 2014
HYKES, Solomon: *Docker 0.9: introducing execution drivers and libcontainer.* 2014
Online verfügbar:
https://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/
Abruf: 2019-04-22

Laute 2015
LAUTE, Hartwig: *Compliance Readiness in deutschen Unternehmen 2015: Status Quo und Handlungsempfehlungen*
Online verfügbar:
https://www.compliance-manager.net/sites/default/files/150303_recommind_-_compliance_studie_whitepaper_0.pdf
Abruf: 2019-03-29

OLG HH 2007
HANSEATISCHE OLG: *Verstoß gegen Impressumspflichten. Beschluss vom 03.04.2007 - 3 W 64/07.* Hamburg, 2007.
Online verfügbar:
https://openjur.de/u/30590.html
Abruf: 2019-03-10

OWASP 2017
OWASP: *OWASP Top 10 Risiken für die Anwendungssicherheit –2017.* 2017
Online verfügbar:
https://www.owasp.org/images/9/90/OWASP_Top_10-2017_de_V1.0.pdf
Abruf: 2019-04-25

Schifrin 2015
SCHIFRIN, Matt: *Not For Investors Only: Top 10 Nuggets Of Buffett Wisdom For Life Success.* Forbes.com, 2015
Online verfügbar:
https://www.forbes.com/sites/schifrin/2015/10/04/not-for-investors-only-top-10-nuggets-of-buffett-wisdom-for-life-success/#362093783971
Abruf: 2019-02-20

Schwalbach 2004
SCHWALBACH, Joachim: *Reputation.* Forschungsbericht, Berlin 2004
Online verfügbar:
https://web.archive.org/web/20130521004541/http://www2.wiwi.hu-berlin.de/institute/im/publikdl/2004-2.pdf
Abruf: 2019-02-14

Stuttard Pinto 2011
STUTTARD, Dafydd; Pinto, Marcus: *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws.* Indianapolis, Indiana, USA: John Wiley & Sons, Inc., Zweite Auflage 2011. - ISBN: 978-1-118-02647-2

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____ _____