A Walk on the Web's Wild Side

STUDIENARBEIT

für die Prüfung zum

Bachelor of Science

des Studiengangs Informatik
Studienrichtung Angewandte Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Samuel Philipp
Daniel Brown
Jan-Eric Gaidusch

4. April 2017

Bearbeitungszeitraum	6 Monate
Matrikelnummern	9207236, 3788021, 8296876
Kurs	TINF14B2
Ausbildungsfirma	Fiducia & GAD IT AG
Gutachter der Studienakademie	Dr. Martin Johns

Abstract

Erklärung

(gemäß §5(3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 29.9.2015)

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

„A walk on the web’s wild side“

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 4. April 2017	
Ort, Datum	Samuel Philipp
Karlsruhe, den 4. April 2017	
Ort, Datum	Daniel Brown
Karlsruhe, den 4. April 2017	
Ort, Datum	Jan-Eric Gaidusch

Inhaltsverzeichnis

Abkürzungsverzeichnis **V**

Abbildungsverzeichnis **VI**

Tabellenverzeichnis **VII**

Listings **VIII**

1 Einleitung **1**
1.1 Einführung 1
1.2 Hintergrund 1
1.3 Team 1
1.4 Aufgabenstellung 1
1.5 webifier 2

2 Grundlagen **4**
2.1 Frontend Technologien und Frameworks 4
2.2 Backend Technologien und Frameworks 4
2.3 Technologien und Frameworks der Tests 5
2.4 Angriffstypen 6
2.4.1 Malware 6
2.4.2 Request Header Investigation 6
2.4.3 JavaScript Port Scanning 6
2.4.4 JavaScript IP Scanning 6
2.4.5 Clickjacking 6
2.4.6 Phishing 6

3 Konzept **7**
3.1 Gesamtkonzept 7
3.1.1 webifier Tests 7

3.1.2 webifier Tester . . . 7
3.1.3 webifier Platform . . . 7
3.1.4 webifier Mail . . . 7
3.1.5 webifier Data . . . 7
3.1.6 webifier Statistics . . . 7
3.2 Testarten . . . 7
3.2.1 Virenscan . . . 7
3.2.2 Vergleich in verschiedenen Browsern . . . 8
3.2.3 Test auf Port Scanning . . . 8
3.2.4 Test auf IP Scanning . . . 8
3.2.5 Link Checker . . . 8
3.2.6 Google Safe Browsing . . . 8
3.2.7 Überprüfung des Zertifikats . . . 8
3.2.8 Erkennung von Phishing . . . 9
3.2.9 Screenshot . . . 9

4 Umsetzung **10**
4.1 Gesamtanwendung . . . 10
4.1.1 webifier Tests . . . 10
4.1.2 webifier Tester . . . 10
4.1.3 webifier Platform . . . 10
4.1.4 webifier Mail . . . 10
4.1.5 webifier Data . . . 10
4.1.6 webifier Statistics . . . 10
4.2 Tests . . . 10
4.2.1 Virenscan . . . 10
4.2.2 Vergleich in verschiedenen Browsern . . . 11
4.2.3 Test auf Port Scanning . . . 11
4.2.4 Test auf IP Scanning . . . 11
4.2.5 Linkchecker . . . 11
4.2.6 Google Safe Browsing . . . 11
4.2.7 Überprüfung des Zertifikats . . . 11
4.2.8 Erkennung von Phishing . . . 11
4.2.9 Screenshot . . . 11

5 Analyse **12**

6 Ausblick **13**

7 Fazit **14**

7.1 Zusammenfassung 14

7.2 Bewertung der Ergebnisse 14

Literaturverzeichnis **X**

Abkürzungsverzeichnis

WWW World Wide Web

JVM Java Virtual Machine

Abbildungsverzeichnis

1 Secutitysquad - Logo 1
2 webifier - Logo 2

Tabellenverzeichnis

Listings

1 Einleitung

1.1 Einführung

TODO Samuel

1.2 Hintergrund

1.3 Team

Abbildung 1: Secutitysquad - Logo

1.4 Aufgabenstellung

Anbieter von zwielichtigen Web-Angeboten greifen ihre User mit diversen Client-seitigen Methoden an. Beispiele für solche Angriffe sind Malware Downloads, Phishing, JavaScript Intranet Angriffe, oder Browser Exploits.

Ziel der Arbeit ist eine systematische Untersuchung der Aktivitäten von semi-legalen Webseiten im World Wide Web (WWW). Das erwartete Ergebnis ist ein Prüfportal, auf dem jene Webseiten automatisiert analysiert werden und Ergebnisse präsentiert werden sollen.

Nach dem ersten Schaffen einer Übersicht von interessanten Zielen, wie z.B. One-Click-Hoster oder File-sharing Sites sollen ausgewählte Webseiten manuell untersucht werden. Außerdem sollen verschiedene Angriffsszenarien zur weiteren Prüfung ausgewählt werden. Der Untersuchungsprozes der Webseiten soll im Verlauf dieser Arbeit stückweise automatisiert und in den Rahmen einer Prüfanwendung gebracht werden.

Abschließend sollen eine Vielzahl von Webseiten mit der Anwendung getestet und die Ergebnisse ausgewertet und dokumentiert werden.

1.5 webifier

Abbildung 2: webifier - Logo

webifier ist eine Anwendung, mit der Webseiten auf deren Seriosität und mögliche clientseitige Angriffe auf den Nutzer geprüft werden können. Sie besteht aus mehreren eigenständigen Teilanwendungen. Im Zentrum steht der Tester, welcher die einzelnen Tests verwaltet, ausführt und anschließend die Ergebnisse auswertet. Jeder einzelne Test ist eine weitere isolierte Teilanwendung des Testers. So kann jeder Test unabhänig von allen anderen betrieben werden.

Die Platform ist eine Webanwendung welche den Endnutzern eine grafische Oberfläche zur Verfügung stellt, um Webseiten zu überprüfen. Im Hintergrund setzt die Plattform auf den Tester auf. webifier Mail ist ein Dienst mit dem Links aus E-Mails überprüft werden können. Anschließend erhält der Sender eine E-Mail mit den Resultaten zurück.

Eine weitere Teilanwendung von webifier ist das Data-Modul. Es stellt eine Schnittstelle für den Tester bereit, um alle Testergebisse sammeln zu können. Das Statisitik-Modul ist die letzte Teilanwendung von webifier. Es setzt auf das Data-Modul auf und stellt Funktionen zur Auswertung aller Testergebnisse bereit.

Um die Techniken und Algorithmen von webifier verstehen zu können sind einige Grundlagen erforderlich, welche nun im nächsten Kapitel genauer vorgsetellt werden.

2 Grundlagen

In diesem Kapitel werden die Grundlagen, welche für das weitere Verständnis der Arbeit und der gesamten Anwendung notwendig sind, näher beschrieben. Zunächst werden die verschiedenen Technologien und Frameworks, sowohl des Frontends, als auch des Backends dargestellt. Anschließend werden einige gängige Angriffstypen im WWW erläutert, welche webifier überprüft.

2.1 Frontend Technologien und Frameworks

- HTML
- CSS
- JavaScript
- jQuery
- Bootstrap

2.2 Backend Technologien und Frameworks

In diesem Abschnitt werden nun alle Technologien und Frameworks vorgestellt welche in den Backends der einzelnen Teilanwendungen zum Einsatz kamen.

Wohl am häufigsten kam die Programmiersprache Java zum Einsatz. Java ist eine universal einsetzbare, nebenläufige, klassenbarierte und objektorientierte Programmiersprache. Sie wurde möglichst einfach gestaltet um von vielen Entwicklern genutzt zu werden. In ihrer Syntax ähnelt sie den Programmiersprachen C und C++. Außerdem ist sie stark und statisch typisiert. Vorallem aber zeichnet sich Java durch seine plattformunabhängigkeit aus. Diese wird dadurch umgesetzt, dass Java-Quellcode in plattformunabhängigen Byte-Code kompiliert wird, welcher von einer Java Virtual Machine (JVM) ausgeführt wird. Java ist eine Hochsprache, die mit Hilfe des so genannten „Garbage Collectors“ eine automatische Speicherverwaltung bereitstellt.[1]

In einigen Teilprojekten wurde das auf Java basierende *Spring*-Framework verwendet.

- Spring
  TODO Samuel
- MongoDB
  TODO Samuel
- Gradle
- REST
- Docker
- R

2.3 Technologien und Frameworks der Tests

- Phantom JS
- Bro
- Python
- HTtrack
  TODO Samuel

[1] Vgl. Gosling u. a. (2014), S. 1

- Resemble JS
  TODO Samuel

2.4 Angriffstypen

2.4.1 Malware

TODO Samuel

2.4.2 Request Header Investigation

2.4.3 JavaScript Port Scanning

2.4.4 JavaScript IP Scanning

2.4.5 Clickjacking

2.4.6 Phishing

TODO Samuel

3 Konzept

3.1 Gesamtkonzept

3.1.1 webifier Tests

3.1.2 webifier Tester

TODO Samuel

3.1.3 webifier Platform

3.1.4 webifier Mail

3.1.5 webifier Data

TODO Samuel

3.1.6 webifier Statistics

3.2 Testarten

3.2.1 Virenscan

TODO Samuel

- Httrack (Umsetzung)

- Download aller Dateien der Webseite
- Scannen der Heruntergeladenen Dateien
  - Clamav (Umsetzung)
  - AVG (Umsetzung)
  - CAV (Umsetzung)

3.2.2 Vergleich in verschiedenen Browsern

3.2.3 Test auf Port Scanning

3.2.4 Test auf IP Scanning

3.2.5 Link Checker

- herausfiltern aller Links und nachgeladenen Ressourcen

3.2.6 Google Safe Browsing

3.2.7 Überprüfung des Zertifikats

TODO Samuel

- Auslesen der relevanten Informationen des Zertifikates der WEbseite
- Validierung des Zertifikates

3.2.8 Erkennung von Phishing

TODO Samuel

- Herausfiltern der Schlagwörter
- Finden möglicher Duplikate der Webseite
  - Erstes Schlagwort zu Top Level Domains
    - com
    - ru
    - net
    - org
    - de
  - Websuche nach den Schlagwörtern mittels Suchmaschinen
    - DuckDuckGo
    - Ixquick
    - Bing

3.2.9 Screenshot

4 Umsetzung

4.1 Gesamtanwendung

4.1.1 webifier Tests

4.1.2 webifier Tester

TODO Samuel

4.1.3 webifier Platform

4.1.4 webifier Mail

4.1.5 webifier Data

TODO Samuel

4.1.6 webifier Statistics

4.2 Tests

4.2.1 Virenscan

TODO Samuel

4.2.2 Vergleich in verschiedenen Browsern

4.2.3 Test auf Port Scanning

4.2.4 Test auf IP Scanning

4.2.5 Linkchecker

4.2.6 Google Safe Browsing

4.2.7 Überprüfung des Zertifikats

TODO Samuel

4.2.8 Erkennung von Phishing

TODO Samuel

4.2.9 Screenshot

5 Analyse

6 Ausblick

7 Fazit

7.1 Zusammenfassung

7.2 Bewertung der Ergebnisse

Literaturverzeichnis

Gosling, James u. a. (2014):
The Java Language Specification - Java SE 8 Edition, 5. Auflage, Addison-Wesley