

**Hochschule Worms**  
**Fachbereich Informatik**  
**Studiengang Angewandte Informatik B.Sc.**

**Penetration Testing kommerzieller Webanwendungen**

Dokumentation des Praxissemesters bei der Firma  
**WALLSEC**

Bruno Macedo da Silva  
676839  
inf3645@hs-worms.de  
Bebelstraße 22 Z18  
67549 Worms

Betreuer	Prof. Dr. Zdravko Bozakov Peter Todorov
Bearbeitungszeitraum:	Summer Semester 2022
Abgabedatum:	xxx. XXXXXX XXXX
Sperrvermerk:	Ja/Nein

# Inhaltsverzeichnis

<b>Glossar</b>	<b>3</b>
<b>Abkürzungen</b>	<b>7</b>
<b>1 Einleitung</b>	<b>8</b>
1.1 Wallsec . . . . .	8
<b>2 Anwendungsdomäne</b>	<b>10</b>
2.1 Theorie über Penetration Testing . . . . .	10
2.2 Phase und Methodologie eines Penetration Testing . . . . .	11
2.3 Penetration Testing in Webanwendungen . . . . .	11
<b>3 Durchführung der Aufgabe</b>	<b>13</b>
<b>4 Wöchentliche Zusammenfassung meines Praxissemesters</b>	<b>14</b>
<b>5 Ausführliche Beschreibung eines Penetration Testing innerhalb des Praxissemesters</b>	<b>15</b>
5.1 Sammlung von Informationen von dem Zielsystem . . . . .	15
5.2 Ausnutzung der Zielanwendung . . . . .	18
5.3 Kundebericht . . . . .	21
<b>6 Fazit</b>	<b>22</b>
<b>Literaturverzeichnis</b>	<b>23</b>

## Glossar

**Confidentiality, Integrity and Availability (CIA)** Beschreibt die drei wichtigsten Schutzziele der IT-Sicherheit: Vertraulichkeit, Integrität und Verfügbarkeit (Wendzel, 2018) .

**Common Vulnerability Scoring System (CVSS)** Internationale Standards für die Bewertung von Verwundbarkeiten von IT-Systemen. Es wurde im Jahr 2005 von dem National Infrastructure Advisory Councils entstanden und wird heute von dem Forum of Incident Response and Security Teams verwaltet (Security Insider, 2019) .

**National Institute of Standards and Technology (NIST)** US-Behörden für die Regelungen, Vereinheitlichung und Weiterentwicklung im Bereich Informationstechnologie zuständig (Hochschule Worms, 2018) .

**Open-source intelligence (OSINT)** Datenerhebung und Sammlung aus offenen Quellen, wie von Online-Repositories, Nachrichten, sozialen Netzwerken, wissenschaftlichen Texten unter anderen öffentlichen Quellen. In diesem Fall gibt es keine direkten Kontakte mit dem Ziel. Es kann passive Reconnaissance genannt werden (Yeboah-Ofori, 2018).

**Open Web Application Security Project®(OWASP®)** Non-Profit Organisation, die sich darauf fokussiert, die Sicherheit in dem Umgang mit Webanwendungen zu gewährleisten. Die Organisation verteilt Open-Source Informationen über sichere Entwicklung, Dokumentation, Best-Practices zu dem sicheren Umgang mit dem Internet und Bildung (Triaxiom Security, 2018).

**Rules of Engagement (ROE)** Bezieht sich auf ein Vertrag, der zwischen Kunden und Tester abgeschlossen wird, um den Umfang und die Rahmenbedingungen des Tests festzulegen. Dieses Dokument enthält unter anderen folgenden Informationen: Umgang mit sensitiven Daten, Notfallkontakten, Identifikation der zu testenden Ob-

jekten und Einschränkungen des Testobjektes (Triaxiom Security, 2018).

**Damn Vulnerable Web Application (DVWA)** ist eine mit Schwachstellen absichtlich entwickelte Webanwendung für Test- und Lernumgebung. Sie wird meist von Entwickler verwendet, um sich mit Schwachstellen und Best-Practices zu kennen. Dieses Plattform hat als Umfang die meisten bekannten Web-Angriffe (DVWA TEAM, 2016) .

**Hypertext Transfer Protocol (HTTP)** ein in 1989 entwickelter Model für die Übertragung von Dateien in dem Internet. Die neue Version, HTTP/2, sollte mehr Sicherheit und Leistung mit wenigen Datennutzung anbieten (Manzoor et al., 2017).

**Proof of Concept (PoF)** ist eine Demonstration, dass eine Methode funktioniert. In dem Sicherheitsbereich zeigt, dass eine Schwachstelle ausnutzbar ist. (Malwarebytes, 2022).

**Cross-Site Scripting (DVWA)** ist ein Angriff, wo bössartige Code in eine Webanwendung absichtlich hinzugefügt wird, um Anmeldedaten oder Sitzungen-Informationen zu fangen. In diesem Fall Ziel des Angriffes ist es, eine legitime Nutzung der Anwendung vorzutäuschen, um Informationen zu stehlen (wie Passwörter, persönliche oder finanzielle Daten) oder die Anwendung zu beschädigen (Mahmoud et al., 2017).

**Burp Suite** auch Burp genannt ist eine von der Firma PortSwigger in Java-Programmiersprache entwickelte Anwendung für die Durchführung von Sicherheitstests in Webanwendungen. Mit verschiedenen Funktionalitäten unterstützt die Anwendung während allen Phasen eines Penetration Testings von Reconnaissance bis zum Angriff (Junmei and YanChengkang, 2021).

**Cortex** Wie TheHive Project, Cortex ist auch eine Open Source Plattform für die Verwaltung und Weiterbearbeitung von Sicherheitsvorfällen. Es funktioniert wie eine

Analysis Engine, die Informationen sammelt und Antworten/Aktionen je nach Fälle durchführt. Es kann allein oder integriert mit TheHive funktionieren (Project, 2021).

**Cyberangriff** Angriffe, die über den Cyberspace stattfinden. Solche Angriffe zielen auf Unternehmen und deren Infrastrukturen, um sie zu zerstören, sie zu lähmen, sie zu kontrollieren oder die Integrität deren Daten zu stehlen oder zu dominieren (NIST, 2020).

**Cybersicherheit** Diese Domäne umfasst Kenntnisse und Methoden für den Schutz, für die Prävention, für die Wiederherstellung von elektronischen Kommunikationsmittel und deren Inhalt. Es konzentriert sich auf ihrer Verfügbarkeit, Integrität, Authentizität, Vertraulichkeit und Verbindlichkeit (NIST, 2020).

**Dirbuster, Gobuster, usw.** sind Anwendungen, die mit Brute Force, versuchen, Dateien und Verzeichnisse innerhalb Webanwendungen zu finden (KALI TOOLS, 2022).

**Javascript** ist ein Programmiersprache, die in Webanwendungen verwendet wird, um komplexe Strukturen wie Animationen, Bilder, Tun und Interaktionen zu implementieren (Mozilla Corporation, 2022).

**nmap** (Network Mapper) ist eine Open-Source Anwendung für die Netzwerkanalyse. Mit diesem Tool ist es möglich, Hosts, Diensten (und deren Versionen) schnell zu entdecken (Nmap.org, 2021).

**Pentester** Auch Ethical Hacker genannt ist ein Sicherheitsanalyst, der sich damit beschäftigt, Schwachstellen von IT-Systemen zu finden (ProSec, 2019).

**Port** ist eine Zahl, die ein Dienst oder eine Verbindung identifiziert. Es geht hier um eine logische Adressierung zur Identifizierung eines oder mehrere Prozessen (Tanenbaum and Wetherall, 2011).

**Schwachstelle** Schwäche eines Systems (Wendzel, 2018).

**Scout Suite** Audi-Tool für Sicherheitsüberprüfungen von Cloud-Umgebungen. Mit dem Tool werden Einstellungsinformationen gelesen und in einem lesbaren Dateiformat ausgegeben (nccgroup, 2022).

**Tenant** Einige Webanwendungen werden so konzipiert, dass verschiedene unabhängige Gruppe verwenden können. Als Beispiel kann eine Plattform für Online-Shop von verschiedenen Anbieter benutzt werden. Obwohl jeder Anbieter seine eigenen Namen, Marken, Produkte hat, benutzen beide nur eine Plattform. Jeder von diesem Anbieter nennen wir Tennants.

**TheHive Project** TheHive ist eine Open Source Plattform für die Verwaltung und Weiterbearbeitung von Sicherheitsvorfälle. Es integriert andere Plattform und Anwendungen, wie Cortex, um Informationen und Handlungen bereitzustellen, damit die Arbeit von Security Operation Center auf einer Plattform konzentriert bleibt (Project, 2021).

**Verwundbarkeit** Auch *vulnerability* genannt. Es beschreibt eine von Angreifer ausnutzbare Schwachstelle (Wendzel, 2018).

**Webanwendung** Internetseiten, die eine Interaktion ermöglichen. Diese Interaktion kann beispielsweise Login, Einkauf, Erstellung und Manipulation von Daten. Die meisten Webanwendungen sind mit einem Datenbank verbunden. Webseite sind seinerseits statische Seite, deren Inhalt nicht dynamisch aktualisiert sind (Essential Designs, 2019).

## **Abkürzungen**

**CIA** Confidentiality, Integrity and Availability.

**CVSS** Common Vulnerability Scoring System.

**DVWA** Damn Vulnerable Web Application .

**DVWA** Cross-Site Scripting.

**FPO** Fachspezifische Prüfungsordnung.

**HTTP** Hypertext Transfer Protocol.

**NIST** National Institute of Standards and Technology.

**OSINT** Open-source intelligence.

**OWASP®** Open Web Application Security Project®.

**PoF** Proof of Concept.

**ROE** Rules of Engagement.

# 1 Einleitung

Mein Praxissemester findet im Rahmen der Fachspezifische Prüfungsordnung (FPO) 2008 für den Studiengang Angewandte Informatik B.Sc. und dessen Modulhandbuch (Hochschule Worms, 2018) statt. Die Stellensuche orientierte sich an dem Schwerpunkt Networks & Security und spezifischer auf Cybersicherheit und Penetration Testing.

Da es sich um einen spezifischen Bereich handelt, war meine Suche nach einer geeigneten Firma, an die ich meine Bewerbungsunterlagen schickten, eingeschränkt. Die Firma Wallsec GmbH in Wiesloch-Walldorf hatte eine offene Studentstelle. Da diese Stelle genau meine Ziele entsprach, bewarb ich mich für die Position. Die Aufgabe in der Stellenbeschreibung beziehen sich hauptsächlich um Durchführung von Penetrationstests, um die Analyse von Source-Code und um Evaluation und Entwicklung von Sicherheitsprozessen. Als Voraussetzung verlangte Wallsec wenig Fachkenntnisse im Bereich Sicherheit und Penetration Testing, aber wollte große Interesse von den Kandidaten Lernbereitschaft (Wallsec Security, 2022).

Das gesamte Prozess von Bewerbung bis zum Einstieg dauerte ungefähr einen Monat. Am 15. Juli 2022 fing ich an, bei Wallsec als Praktikant im Vollzeit zu arbeiten. In diesem Bericht werden wir folgende Themen bearbeiten:

- Informationen über die Firma Wallsec
- Konzepte von Penetration Testing
- Aufgabebereich der Tätigkeit
- Ergebnis des Praxissemesters

## 1.1 Wallsec

Die Firma wurde im Jahr 2020 von Peter Todorov in Walldorf-Wiesloch, Baden Württemberg, gegründet. Laut der Beschreibung der Webseite fokussierte sie sich auf die



Planung, Bereitstellung und Risikoanalysen von Sicherheits-Infrastrukturen von Firmen verschiedenen Größen (Wallsec Security, 2022). Wallsec bietet folgenden Leistungen an:

- Penetrationstests
- Schwachstellenmanagement
- Richtlinien
- Automatisierung
- DevOps und CI/CD Pipeline Sicherheit
- Beratung im Bereich Cyberabwehr

## 2 Anwendungsdomäne

Oft gibt es Nachrichten über Firmen oder Regierungen, deren Geheimnisse im Netz von Angreifer veröffentlicht wurden oder deren Dienste wegen Cyberangriff unerreichbar sind. Da solche Situationen öfter als je vorkommen, ist das Interesse und die Forschung in dem Bereich Cybersicherheit in den letzten Jahren rasant gestiegen (Tanenbaum, 2009).

Diese Angriffe verletzen die drei wichtigsten Ziele der Cybersicherheit: die Vertraulichkeit, die Integrität und die Verfügbarkeit (aus dem Englisch Confidentiality, Integrity and Availability (CIA)). Diese Zielen werden in den Fachliteraturen folgend beschrieben: Schutz gegen unautorisierte Informationsgewinnung; Schutz gegen unautorisierte Datenmanipulation und Zugriffsgewährleistung für authentifizierte und autorisierte Subjekte (Wendzel, 2018). Ein Angriff zielt auf Schwachstelle oder auf Verwundbarkeit eines Systems. Diese wird dann zu einer Bedrohung, wenn es möglich ist, dieses System auszunutzen.

### 2.1 Theorie über Penetration Testing

Schwachstellenanalyse und Penetration Testing sind heutzutage oft verwendete Methode, um Verwundbarkeit zu finden und zu analysieren. Der National Institute of Standards and Technology (NIST) beschreibt das Erste als systematische Analyse eines Systems oder Produktes in Bezug auf ihrer Sicherheit, um dessen Schwachstelle zu finden; und das Zweite als Methode für die Verifizierung von binärischen Komponenten oder Anwendungen im Ganzen, um zu finden, ob dessen Verwundbarkeiten in Bezug auf ihre Daten oder Resources ausnutzbar sind (NIST, 2020). Die Schwachstellenanalyse umfasst auch eine Datenerhebung, die später in der Penetration Testing auf eine autorisierte Weise ausgenutzt wird (Goel and Mehtre, 2015). Das Ergebnis dieser zwei Prozessen werden später dem Beauftragten bekanntgemacht, damit Sicherheitsmaßnahmen umgesetzt werden können.

Der Begriff ist auch als “Ethikal Hacking”, als “Pentest” oder als “white hats (weiße Huts)”

bekannt. Es umfasst spezifische Analyse von Drohungen und von Verwundbarkeiten eines Produktes und es findet im Rahmen eines Vertrags oder ROE zwischen einem Kunde und der Firma oder Person statt, die für die Tests verantwortlich sind (Bishop, 2007).

## 2.2 Phase und Methodologie eines Penetration Testing

Ein Penetration Testing findet in einer systematischen Reihenfolge mit drei Hauptphasen statt: Vorbereitung, Implementation und Analysis (Shebli and Beheshti, 2018). Während der Vorbereitung werden der Umfang, die Ziele und die Dauer definiert. Bei der Implementation wird das System oder das Produkt in ihrem Aufbau erkannt, analysiert und ausgenutzt (*exploited*). In der letzten Phase werden die gefundenen Verwundbarkeiten sämtliche Lösungsvorschläge dem Beauftragten mitgeteilt. In manchen Fällen kann das getestete Objekt bezüglich seiner Sicherheit mithilfe der Common Vulnerability Scoring System (CVSS) bewertet werden. Dieses Punktsystem stellt eine international anerkannte Evaluation eines Objekts dar.

Es gibt drei bekannte Methodologie, wo ein Penetration Testing stattfindet: *white box*, *black box* oder *zero-knowledge* und *grey box*. In der ersten Methodologie bekommen die Tester ausführliche Informationen über das getestete Objekt, wie Quellcode, interne Logik und Struktur. In der zweiten bekommen die Tester nur Open-source intelligence (OSINT) Informationen. Die dritte Variante ist eine Mischung aus den ersten zwei, in diesem Fall bekommen die Tester beschränkte Informationen über das zu testende Objekt (Ehmer and Khan, 2012).

## 2.3 Penetration Testing in Webanwendungen

Webanwendungen bieten ihren Nutzer eine dynamische und interaktive Umgebung, ohne dass man Anwendungen in dem eigenen Rechner installieren muss. Im Vergleich zum Desktop-Anwendungen erlauben Webanwendungen, dass mehrere Nutzer die Anwendungen gleichzeitig benutzen können. Der Zugriff findet über verschiedene Plattformen statt,

wie Handys, Desktop, Tablet und Laptops. Sie bietet auch günstigere und schnellere Wartungsmaßnahmen an, da die Hardwarekonfiguration nicht ständig aktualisiert werden muss (TechoTarget, 2019).

Da die Häufigkeit der Transaktionen mit Webanwendungen ständig steigt, müssen Anbieter die Sicherheit dieser Anwendungen gewährleisten. Die Open Web Application Security Project® (OWASP®) kümmert sich darum, um die Sicherheit in Webanwendungen zu recherchieren. Die Publikationen von der Organisation werden weltweit von Sicherheitsfirmen, Entwickler und Pentester verwendet, um die Tests durchzuführen.

Jährlich veröffentlicht OWASP® eine Liste mit den zehn häufigsten Angriffen in Webanwendungen und sichere Maßnahmen, um sie zu vermeiden. Die Organisation bietet auch eine eigene *Security Testing Guide* an, die die Arbeit von Penetration Testing unterstützt, um die Schwachstellen von Anwendungen zu finden, zu überprüfen und zu härten.

Innerhalb meines Praxissemesters spielten die Publikationen von der Organisation eine wichtige Rolle, um spezifische Kenntnisse zu erwerben und um meine Arbeitsweise an den heutigen Anforderungen anzupassen.

### 3 Durchführung der Aufgabe

In diesem Kapitel beschreiben wir konkret, wie die Arbeit während meines Praxissemesters sich entwickelte. Die ersten zwei Wochen dienten als Einarbeitung und Einstieg. Nach dieser Phase bekam ich langsam und unter Betreuung mehr Verantwortung und mehr Freiheit, um die Arbeit durchzuführen. In der folgenden Tabellen wird der Ablauf systematisch und ohne Einzelheit beschrieben. In dem zweiten Teil dieses Kapitels geben wir eine ausführliche Beschreibung eines Projekts.

Jedes Projekt besitzt innerhalb von Wallsec einen festgelegten Aufbau. Dieser kann in den folgenden Punkten zusammengefasst werden:

1. *Kick-off Meeting* mit den Kunden, um grundsätzliche Informationen über die Anwendung zu bekommen
2. Definition des Testumfangs, wie Anmeldedaten, Rolle der Nutzer, Tennants und Einschränkungen
3. Durchführung von Tests nach einer vorgegebenen Checklist
4. Dokumentation der durchgeführten Testen, deren gefundene Schwachstellen und Vorschläge zur Härtung der Anwendung
5. Abschlussmeeting mit den Beauftragten, um die Schwachstellen und deren Ausnutzung zu präsentieren und zu demonstrieren

## 4 Wöchentliche Zusammenfassung meines Praxissemesters

Auflistung der Aufgabe	
Woche	Aufgabenbeschreibung
1 - 2	<p>Einarbeitung:</p> <ul style="list-style-type: none"> <li>• Installation von einer virtuellen Maschine für die Testumgebungen</li> <li>• Einführung in der Arbeitsablauf der Firma</li> <li>• Einführung, Installation und Einstellungen von Burp Suite</li> <li>• Einführung in einem laufenden Projekt, um über das Ablauf- und Dokumentationsverfahren zu lernen</li> <li>• Durchführung und Wiederholungen von einigen Tests, um mich an den gegebenen Tools zu gewöhnen</li> <li>• Teilnahmen an einer Abschlussmeeting des laufenden Projekts, um das Verfahren und den Ablauf des Kundenkontakt zu erkennen und später zu wiederholen</li> </ul>
3 - 4	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an einem Versicherungsanwendung mit dem obigen beschriebenen Schritte (3)
5 - 6	Weiterarbeitung an der Installation, an der Einstellungen und an der Nutzung der Tools TheHive Project und Cortex. Bereitstellung von Skripts zum Herunterladen von statistische Daten der Anwendungen und zur Automatisierung deren Nutzung.
7 - 8	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an einem Marketing-Webanwendung mit dem obigen beschriebenen Schritte (3)
9 - 10	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an Netzwerk-Umgebungen mit dem obigen beschriebenen Schritte (3). Die durchgeführten Tests konzentrieren sich auf die Sicherheit einer Netzwerk in einer Cloud-Umgebung. Für dieses Projekt spielen die Tools nmap und Scout Suite eine wichtige Rolle, da das Ziel war, Hosts, Dienste und dessen Einstellungen und Schwachstelle zu erkennen
11	Start, Durchführung und Abschluss eines Pentesting bei einem umfangreichen Webanwendung mit verschiedenen Tennants und Nutzungsrolle für die Verwaltung von Business-Processes.
11 - 14	Start, Durchführung und Abschluss eines Pentesting bei dessen Aufbau auf Hypertext Transfer Protocol (HTTP) und <b>Graphisql</b> basiert ist. Da diese beiden Technologie mir ganz neu waren, beschäftigte ich mich intensiv damit, sie in ihrem Aufbau und in ihren möglichen Schwachstellen auszuken- kennen.

## **5 Ausführliche Beschreibung eines Penetration Testing innerhalb des Praxissemesters**

Bevor der Durchführung jedes Tests ist Wallsec und ihre Mitarbeiter dazu verpflichtet, eine Vertraulichkeitserklärung zu unterschreiben. Keine Information weder über die Firma noch über die verwendeten Methode dürfen aus irgendwelcher Form veröffentlicht werden. Aus diesem Grund werden die hier demonstrierten Methode und Tests in der Test- und Lernumgebung Damn Vulnerable Web Application (DVWA) gezeigt. In den realen Tests verwenden wir ähnliche Methode, manchmal mit mehr oder weniger Details, um die Sicherheit der Anwendungen zu überprüfen.

Da dieses Bericht eingeschränkte Platz hat und das dieses Thema sehr umfangreich hier, demonstrieren wir in den nächsten zwei Unterkapitel einige Methode, die wir verwenden, um die Zielanwendung in ihrer Struktur zu kennen und auszunutzen.

### **5.1 Sammlung von Informationen von dem Zielsystem**

Obwohl jede Webanwendungen ihre eigene Eigenschaften und Ziele haben, besitzen fast alle eine ähnliche Struktur und Aufbau. Beim jeden Test fangen wir damit an, diese gemeinsame Struktur zu erkennen, indem wir nach öffentlichen Informationen suchen. Viele kritische Informationen, wie Username, Passwörter, Versionen, Systemen verbundenen IP-Adressen, lassen sich nach einer Online-Suche finden. Auch mit eingebauten Tools eines Betriebssystems können wir auf solche Informationen zugreifen. Dieses Verfahren nennen wir Banner Grabbing. Das folgende Bild zeigt ein Beispiel von einer einfachen Durchführung von Banner Grabbing:

```

bruno@DESKTOP: ~/git/Praxisbericht_Bachelorarbeit$ telnet www.google.de 80
Trying ...
Connected to www.google.de.
Escape character is '^]'.
GET / HTTP/1.1

HTTP/1.1 200 OK
Date: Sun, 18 Sep 2022 13:59:56 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: AEC; expires=Fri, 17-Mar-2023 13:59:57 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked

```

Abbildung 1: Banner Grabbing mithilfe von dem Tool telnet

Eine Webanwendung ist eine Gruppierung von verschiedene Verzeichnisse. Jedes Verzeichnis soll den Nutzer eine Information oder Interaktion anbieten. Manche sind aber nicht für Nutzer gedacht und dient zur Verwaltung von Einstellungen. Da solche Verzeichnisse nicht direkt aufrufbar sind benutzen wir Tester andere Methode, um zu finden, was die Entwickler im Hintergrund beibehalten wollte. Es gibt verschiedene Tools, die mithilfe von sogenannten *wordlists*, viele Anfrage an eine Anwendung schickt, um herauszufinden, was nicht direkt von dem Browser aufrufbar ist. Solche *wordlists* sind Textdateien, die häufige verwendete Wörter beinhalten, die für Webanwendungen, für Nutzernamen oder für Passwörter verwendet werden. Da viele Webanwendungen ähnliche Strukturen haben, ist auch meistens erwartet, dass gewöhnliche Wörter auch zu finden sind. Das nächste Beispiel zeigt uns, dieses Entdeckungsverfahren auf unserem Ziel, DVWA Tool. Hier benutzen wir das Tool Dirbuster, Gobuster, usw., um herauszufinden, welche Verzeichnisse in dieser Anwendung existieren. In diesem Fall werden verschiedene Anfrage geschickt, jede mit einem verschiedenen Wort, um zu sehen, welche positive Antworten liefern. Das folgende Bild zeigt die Durchführung und das Ergebnis des Scanverfahren mithilfe des Tools Dirbuster, Gobuster, usw.:



```
└─$ dirb http://localhost/dvwa/ /usr/share/dirb/wordlists/common.txt -w

DIRB v2.22
By The Dark Raver

START TIME: Sun Sep 18 10:33:37 2022
URL_BASE: http://localhost/dvwa/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

Scanning URL: http://localhost/dvwa/
+ http://localhost/dvwa/.git/HEAD (CODE:200|SIZE:23)
=> DIRECTORY: http://localhost/dvwa/config/
=> DIRECTORY: http://localhost/dvwa/database/
=> DIRECTORY: http://localhost/dvwa/docs/
=> DIRECTORY: http://localhost/dvwa/external/
+ http://localhost/dvwa/favicon.ico (CODE:200|SIZE:1406)
+ http://localhost/dvwa/index.php (CODE:302|SIZE:0)
+ http://localhost/dvwa/php.ini (CODE:200|SIZE:154)
+ http://localhost/dvwa/phpinfo.php (CODE:302|SIZE:0)
+ http://localhost/dvwa/robots.txt (CODE:200|SIZE:26)
=> DIRECTORY: http://localhost/dvwa/tests/
```

Abbildung 2: Brute force Scan für Verzeichniserdeckung

Der nächste Schritte wäre eine manuelle Beobachtung der entdeckten Material, um zu finden, ob irgendwelche sensitiven Information ausgelieferte wurde. Falls ja, wurden wir dann versuchen diese Schwachstelle, zu erkennen und auszunutzen.

Ein Netzwerk-Scan ist auch eine häufige verwendete Methode, um Dienste innerhalb des testenden Objekts zu finden. Dieser Scan schickt an das Ziel verschiedene Anfrage, damit wir das Reaktion des Zielsystems beobachten können. Während bei dem ersten Scan wir uns auf der Webanwendung fokussierten, bearbeiten wir hier eine Ebene, die nicht für normale Nutzung gezielt ist. Unser Fokus liegt auf dem Server, wo die Anwendung läuft. Dafür testen wir die sogenannte Port. Aus diesem Scan lassen sich meistens viele nützliche Informationen herausfinden, wie Betriebssystem, wo die Webanwendung läuft, Name und Versionen der existierenden Dienste. Mit dieser Information ist es möglich dann zielgerichtete Angriffe vorzubereiten, um Schwachstellen auszunutzen.

Das nächste Bild zeigt das Ergebnis der Durchführung von nmap gegen das Testziel *scanme.nmap.org*:

```

$ nmap -A scanme.nmap.org
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 10:57 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.15s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
|_ http-favicon: Nmap Project
|_ http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open  nping-echo   Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.41 seconds

```

Abbildung 3: Brute force Scan für Verzeichniserdeckung

Aus diesem Scan erfahren wir welche Betriebssystem und welche Version der Webanwendung benutzt werden. Auch wenn solche Versionen gegen Angriffe geschützt sind, ist es unsere Aufgabe dem Kunde zu informieren, dass sensitive Informationen für alle sichtbar sind. Eine böse absichtliche Nutzer könnte diese Information nutzen, um eine Schwachstellen für diese Anwendungen zu erfinden und dann auszunutzen.

## 5.2 Ausnutzung der Zielanwendung

Nachdem die vorherigen Scans durchgeführt wurden und öffentliche Serverseite Informationen gesammelt wurden, fangen wir mit den Tests auf der Webanwendung an. In diesem Fall ist es unser Ziel zu wissen, welche versteckte Daten oder unerlaubte Aktionen ein Angreifer durchführen kann, um die CIA der Anwendung zu verletzen. Für die folgenden Tests benutzen wir unter anderen auch das Tool Burp Suite.

Unser erster Test will überprüfen, ob es möglich ist, in ein Eingabefeld Daten einzutragen und das normale Verhalten der Anwendung zu ändern. Wir wollen eigenen Code hinzufügen und in dem Falle, dass es uns gelingt, das zu tun, können wir dann weitere Code hinzufügen, um Daten von Nutzer zu stehlen oder das normale Verhalten der Anwendung beschädigen. Wir prüfen hier, ob die Anwendung gegen Cross-Site Scripting (XSS) anfällig ist. Um diesen Test durchzuführen, fügen wir erwartete Daten hinzu, um das Verhalten der Anwendung zu beobachten. Nachdem wir das normale Verhalten erkannt

haben, versuchen wir eigenen Code hinzufügen und beobachten, ob wir die Anwendung ausnutzen können.

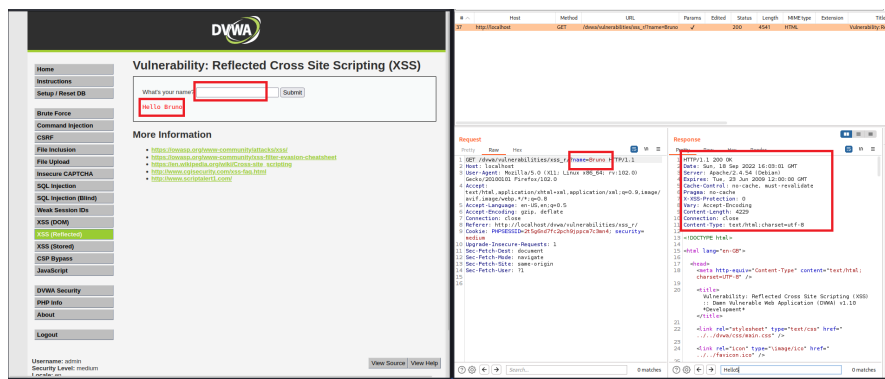


Abbildung 4: Beobachtung der Anwendung unter normale Nutzung

Das Bild 4 zeigt den ersten Test. Aus dieser Aufnahme der Anfrage können wir sehen, sehen, dass die Nutzereingabe direkt in dem Browser stattfindet. Wir sehen in der Antwort Informationen über den Aufbau der Anwendung und wie sie auf unsere Anfrage reagiert. Auf dem nächsten Bild versuchten wir einige bösartige Code hinzufügen, um den normalen Ablauf der Anwendung zu verletzen. Dafür verwenden Javascript Code. Wir können unseren Code direkt in die Anwendung, in eine selbst gebastelte Request oder in Burp Suite eingabe:

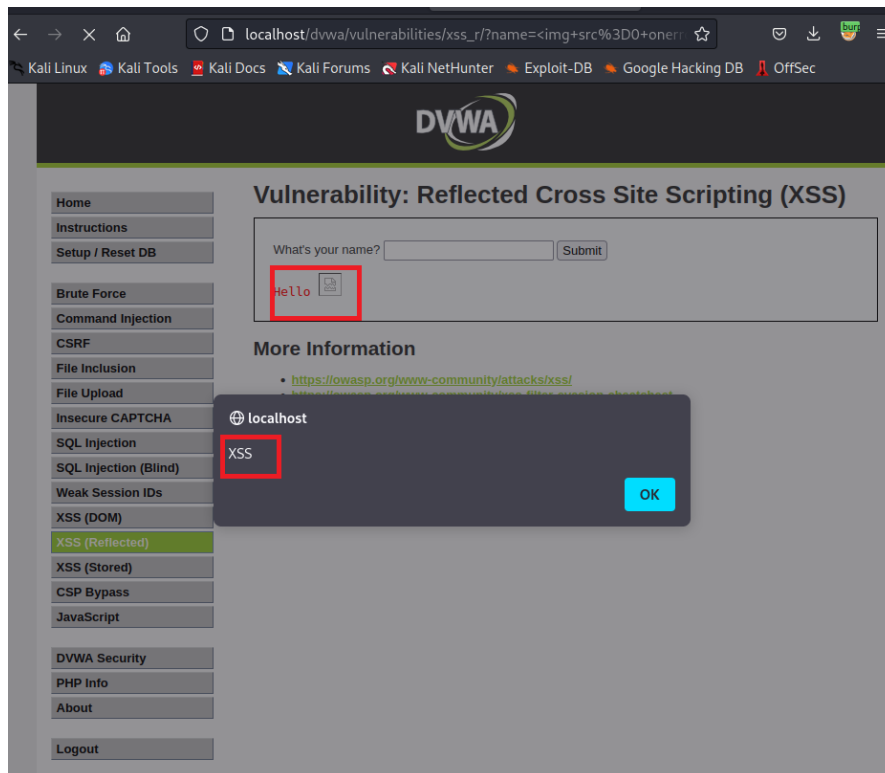


Abbildung 5: Einführung von böartigen Code und Beobachtung der Reaktion der Anwendung.

Für diesen Test haben wir den Code `<img src=0 onerror='alert("XSS")'|>` hinzugefügt. Das Ziel dieses Codes ist ein nicht existierende Bild hinzufügen, um einen absichtlichen Fehler zu provozieren. Dieser Fehler zeigt ein kleines Fenster in der Anwendung mit dem Text "XSS". Eine geschützte Anwendung würde entweder den Code und ihren Zeichen "< >" ignorieren oder diese zu anderen übersetzen. Es kann auch sein, dass die Anwendung den Nutzer sagt, dass die eingegebenen Zeichen nicht erlaubt nicht. Aus dem Bild sehen wir aber, dass die Anwendung alle Zeichen akzeptiert und sogar erlaubt, dass der Code ausgeführt wird. Aus dieser Situation hätten wir einen Proof of Concept (PoF), dass die Anwendung gegen diese Art von Angriff anfällig ist.

### 5.3 Kundebericht

Je nachdem wie lange das Projekt läuft, können wir mehr oder weniger zeitintensive Tests durchführen. Am Ende des Projekts präsentieren wir dem Beauftragten in einem Meeting unser Ergebnis und liefern wir ein Bericht mit detaillierte Informationen über die gefundenen Schwachstellen und die dazu verwendete Methode, um sie zu finden. Dieses Bericht wird so geschrieben, damit non-native Nutzer es verstehen können.

In den ersten Abschnitten erklären wir mit weniger technischen Begriffe, wie was für Tests durchgeführt wurden. In den folgenden Kapitel erklären wir mit mehr Einzelheiten und technische Details, wie wir zu unserem Ergebnis kamen. Anschließend geben wir Vorschläge für die Verbesserung der Sicherheit der Webanwendung und am Ende geben wir mithilfe von CVSS oder von den Kunden ausgewählten Metrik eine allgemeine Bewertung.

## 6 Fazit

Das Praxissemester bat mir eine sehr gute Möglichkeit an, die mein theoretisches Wissen von der Hochschule in der Praxis anzuwenden. Zusätzlich konnte ich weitere und tiefere Kenntnisse in dem Bereich Sicherheit und Penetration Testing für mich gewinnen. Alle Module meines Studiums waren eine Vorbereitung und gutes Vorwissen für die Praxis. Sie gaben mir eine solide Grundlage und die richtigen Soft Skills für die Weiterentdeckung des Sicherheitsbereiches Penetration Testings.

Die entwickelten Eigenschaften werden mein zukünftiges Berufsleben prägen und werden von großer Bedeutung für die Durchführung einer hochwertigen Arbeit sein. Da aber diese Kenntnisse nicht ausreichend sind, stehe ich erst am Anfang meines Entdeckungswegs, der dazu führt, meine Karriere in Ethical Hacking aufzubauen.

## Literaturverzeichnis

- Bishop, M. (2007). About penetration testing. *IEEE Security & Privacy*, 5(6):84–87.
- Brito, G. and Valente, M. T. (2020). Rest vs graphql: A controlled experiment. In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 81–91.  
<https://doi.org/10.1109/ICSA47634.2020.00016>. Zugriff am 9te Oktober.
- DVWA TEAM (2016). Damn vulnerable web application.  
<https://github.com/digininja/DVWA>. Zugriff am 18te September 2022.
- Ehmer, M. and Khan, F. (2012). A comparative study of white box, black box and grey box testing techniques. *International Journal of Advanced Computer Science and Applications*, 3.  
<http://dx.doi.org/10.14569/IJACSA.2012.030603>. Zugriff am 31 Juni 2022.
- Essential Designs (2019). Website vs web app: What’s the difference?  
<https://www.triaxiomsecurity.com/rules-of-engagement-important-to-penetration-test/>. Zugriff am 7te August 2022.
- Goel, J. N. and Mehtre, B. (2015). Vulnerability assessment & penetration testing as a cyber defence technology. *Procedia Computer Science*, 57:710–715.  
<https://www.sciencedirect.com/science/article/pii/S1877050915019870/>. Zugriff am 31 Juni 2022.
- Healthcare Computing (2021). Was ist bzw. tut das national institute of standards and technology (nist)?  
<https://www.healthcare-computing.de/was-ist-bzw-tut-das-national-institute-of-standards-and-technology-nist-a-1022210/>. Zugriff am 31 Juni 2022.
- Hochschule Worms (2018). Fachspezifische prüfungsordnung (fpo 2018).  
[https://www.hs-worms.de/fileadmin/media/fachbereiche/informatik/AInf/P\\_ruefungsordnung/AnInf\\_FPO\\_2017-12-19\\_FINAL.pdf](https://www.hs-worms.de/fileadmin/media/fachbereiche/informatik/AInf/P_ruefungsordnung/AnInf_FPO_2017-12-19_FINAL.pdf). Zugriff am 31 Juni 2022.
- Junmei, W. and YanChengkang (2021). Automation testing of software security based on burpsuite. In *2021 International Conference of Social Computing and Digital Economy (ICSCDE)*, pages 71–74.  
<https://doi.org/10.1109/ICSCDE54196.2021.00025>. Zugriff am 7te August 2022.
- KALI TOOLS (2022). dirb, gobuster.  
<https://gitlab.com/kalilinux/packages/dirbuster>. Zugriff am 18te September 2022.
- Mahmoud, S. K., Alfonse, M., Roushdy, M. I., and Salem, A.-B. M. (2017). A comparative analysis of cross site scripting (xss) detecting and defensive techniques. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 36–42.  
<https://doi.org/10.1109/INTELCIS.2017.8260024>. Zugriff am 18te September 2022.
- Malwarebytes (2022). Proof of concept.

- [https://www.malwarebytes.com/glossary/proof-of-concept#:~:text=A%20proof%20of%20concept%20\(PoC,12th%20Floor](https://www.malwarebytes.com/glossary/proof-of-concept#:~:text=A%20proof%20of%20concept%20(PoC,12th%20Floor). Zugriff am 18te September 2022.
- Manzoor, J., Drago, I., and Sadre, R. (2017). How http/2 is changing web traffic and how to detect it. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9.  
<https://doi.org/10.23919/TMA.2017.8002899>. Zugriff am 9te Oktober 2022.
- Mozilla Corporation (2022). What is javascript?  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). Zugriff am 18te September 2022.
- nccgroup (2022). Scoutsuite.  
<https://github.com/nccgroup/ScoutSuite>. Zugriff am 11 September 2022.
- NIST (2020). Cyber attacke.  
[https://csrc.nist.gov/glossary/term/Cyber\\_Attack](https://csrc.nist.gov/glossary/term/Cyber_Attack). Zugriff am 31 Juni 2022.
- Nmap.org (2021). Nmap-referenz-handbuch (man page)m.  
<https://nmap.org/man/de/index.html>. Zugriff am 21 August 2022.
- Openvpn (2022). How cybersecurity has changed in the last decade.  
<https://openvpn.net/blog/how-cybersecurity-has-changed-in-the-last-decade/>. Zugriff am 31 Juni 2022.
- OWASP (2001). Who is the owasp® foundation?  
<https://owasp.org/>. Zugriff am 7te August 2022.
- Project, T. (2021). Thehive - a 4-in-1 security incident response platform.  
<https://thehive-project.org/>. Zugriff am 14 August 2022.
- ProSec (2019). Der job als penetration tester.  
<https://www.prosec-networks.com/blog/der-job-als-penetration-tester/>. Zugriff am 7te August 2022.
- Security Insider (2019). Was ist cvss?  
<https://www.security-insider.de/was-ist-cvss-a-853465/>. Zugriff am 31 Juni 2022.
- Shebli, H. M. Z. A. and Beheshti, B. D. (2018). A study on penetration testing process and tools. In *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–7.  
<https://doi.org/10.1109/LISAT.2018.8378035>. Zugriff am 31 Juni 2022.
- Tanembaum, A. S. (2009). *Moderne Betriebssysteme*. Pearson, München.
- Tanembaum, A. S. and Wetherall, D. (2011). *Computer Networks*. Prentice Hall, München, 5 edition.
- Techtarget (2019). Web application (web app).  
<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>. Zugriff am 7te August 2022.
- Triaxiom Security (2018). Why are rules of engagement important to my penetration



test?

<https://www.triaxiomsecurity.com/rules-of-engagement-important-to-penetration-test/>. Zugriff am 31 Juni 2022.

Wallsec Security (2022). About us.

<https://www.wallsec.de>. Zugriff am 31 Juni 2022.

Wendzel, S. (2018). *IT-Sicherheit für TCP/IP- und IoT-Netzwerke*. Springer Vieweg, Wiesbaden.

Wendzel, S. and Plötner, J. (2007). *Praxisbuch Netzwerk-Sicherheit: Risikoanalyse, Methoden und Umsetzung; für Unix-Linux und Windows; VPN, WLAN, Intrusion Detection, Disaster Recovery, Kryptologie*. Galileo Computing, Bonn.

Yeboah-Ofori, A. (2018). Cyber intelligence and osint: Developing mitigation techniques against cybercrime threats on social media. *International Journal of Cyber-Security and Digital Forensics*, 7:87–98.

<http://dx.doi.org/10.17781/P002378>. Zugriff am 31 Juni 2022.