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:	Wintersemester 2022
Abgabedatum:	xxx. XXXXXX XXXX
Sperrvermerk:	Ja/Nein

Inhaltsverzeichnis

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 eingeführt und wird heute von dem Forum of Incident Response and Security Teams verwaltet (Security Insider, 2019) .

National Institute of Standards and Technology (NIST) Eine US-Behörden, die für die Reglungen, 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 auch passive Reconnaissance genannt werden (Yeboah-Ofori, 2018).

Open Web Application Security Project®(OWASP®) Eine 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 anderem folgenden Informationen: Umgang mit sensitiven Daten, Notfallkontakten, Identifikation der zu testenden Ziel-

systeme und Einschränkungen des Testsystem (Triaxiom Security, 2018).

Damn Vulnerable Web Application (DVWA) ist eine mit Schwachstellen absichtlich entwickelte Webanwendung für Test- und Lernumgebung. Sie wird meist von Entwicklern 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 es, dass eine Schwachstelle ausnutzbar ist. (Malwarebytes, 2022).

Cross-Site Scripting (XSS) ist ein Angriff, wo bösartige Code in eine Webanwendung absichtlich hinzugefügt wird, um an Anmeldedaten oder Sitzungsinformationen zu gelangen. In diesem Fall ist das Ziel des Angriffes, 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 ist, Cortex 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 eigenständig 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, lähmen, kontrollieren oder die Integrität deren Daten zu stehlen oder zu dominieren (NIST, 2020).

Cybersicherheit Diese Domäne umfasst Kenntnisse und Methoden für den Schutz, Prävention, 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).

GraphQL eine von Facebook entwickelte Sprache für die Kommunikation zwischen Anwendungen, um die genau angefragte Information zu bekommen (Brito and Valente, 2020).

Javascript ist eine Programmiersprache, die in Webanwendungen verwendet wird, um komplexe Strukturen wie Animationen, Bilder, Ton 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 (Tanembaum and Wetherall, 2011).

Python ist eine im Jahr 1991 entwickelte Programmiersprache, die Flexibilität, Lesbarkeit und Wiederverwendung anbieten sollte. Python stellt verschiedene Module zur Verfügung, die sich leicht an anderen Anwendungen anpassen sollen (Python, 2022).

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).

Skript in dem Informatikbereich bezieht sich meistens auf eine Textdatei, die Kommandos in einer Programmiersprache, wie Python, Javascript, Bash oder anderen, beinhalt. Sie dienen dazu, Aufgabe zu automatisieren, die sonst manuell durchgeführt werden sollten (Christensson, 2006).

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

TheHive Project TheHive ist eine Open Source Plattform für die Verwaltung und Weiterbearbeitung von Sicherheitsvorfällen. Es integriert andere Plattformen und Anwendungen, wie Cortex, um Informationen und Handlungen bereitzustellen, damit die Arbeit vom 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 sein. Die meisten Webanwendungen sind mit einer Datenbank verbunden. Webseiten sind anderseits statische Seiten, deren Inhalt nicht dynamisch aktualisiert ist (Essential Designs, 2019).

Websocket Netzwerkprotokoll auf der Transportschicht, das eine bidirektionale Verbindung zwischen *Client* und *Server* anbietet. Während bei Hypertext Transfer Protocol (HTTP) findet die Verbindung immer auf Basis von *Request* und *Response* statt, bei Websockets wird eine Verbindung hergestellt und Daten werden im Laufzeit dieser Verbindung in beiden Richtung geschickt, ohne dass *Client* und/oder *Server* sich erneut identifizieren müssen (Pimentel and Nickerson, 2012).

Abkürzungen

CIA Confidentiality, Integrity and Availability.

CVSS Common Vulnerability Scoring System.

DVWA Damn Vulnerable Web Application .

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.

XSS Cross-Site Scripting.

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 schicken konnte, sehr eingeschränkt. Die Firma WALLSEC GmbH in Walldorf hatte eine offene Stelle für ein Praxissemester. Da diese Stelle genau meinen Zielen entsprach, bewarb ich mich für diese Position. Die Aufgaben in der Stellenbeschreibung bezogen sich hauptsächlich auf die Durchführung von Penetrationstests, Analyse von Source-Code und auf die Evaluierung und Entwicklung von Sicherheitsprozessen. Als Voraussetzung verlangte WALLSEC gewisse Fachkenntnisse im Bereich Sicherheit und Penetration Testing, aber vor allem verlangte großes Interesse am Thema und Lernbereitschaft (Wallsec Security, 2022).

Der gesamte Prozess von Bewerbung bis zum Einstig 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, Baden Württemberg, gegründet. Laut der Beschreibung der Webseite fokussiert sie sich auf die Planung, Be-

reitstellung und Risikoanalysen von Sicherheits-Infrastrukturen von Firmen verschiedener Größen (Wallsec Security, 2022). WALLSEC bietet folgenden Leistungen an:

- Penetrationstests
- Schwachstellenmanagement
- Sicherheitsrichtlinien
- 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 Angreifern veröffentlicht wurden oder deren Dienste aufgrund eines Cyberangriffes unerreichbar sind. Da solche Situationen immer öfter vorkommen, ist das Interesse und die Forschung in dem Bereich Cybersicherheit in den letzten Jahren rasant gestiegen (Tanembaum, 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 wie folgt beschrieben: Schutz gegen unautorisierte Informationsgewinnung; Schutz gegen unautorisierte Datenmanipulation und Zugriffsgewährleistung für authentifizierte und autorisierte Subjekte (Wendzel, 2018). Ein Angriff zielt auf eine Schwachstellen oder auf die Verwundbarkeiten eines Systems. Diese könnten dann zu einer Bedrohung werden.

2.1 Theorie über Penetration Testing

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

Der Begriff ist auch als “Ethical Hacking”, “Pentest” oder als “White Hats (weiße Hüte)”

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

2.2 Phase und Methodologie eines Penetration Testing

Ein Penetration Testing findet in einer systematischen Reihenfolge mit drei Hauptphasen statt: Vorbereitung, Implementierung und Analyse (Shebli and Beheshti, 2018). Während der Vorbereitung werden der Umfang, die Ziele und die Dauer definiert. Bei der Implementierung wird das System oder das Produkt nach Schwachstellen untersucht und diese werden ausgenutzt (*exploited*). In der letzten Phase werden die gefundenen Verwundbarkeiten zusammen mit sämtlichen Vorschlägen zu mitigierenden Maßnahmen dem BAuftraggeber mitgeteilt. In manchen Fällen kann das getestete Zielsystem bezüglich seiner Sicherheit mithilfe der Common Vulnerability Scoring System (CVSS) bewertet werden. Dieser Punktmechanismus stellt eine international anerkannte Evaluierung Zielsystems dar.

Es gibt drei bekannte Methodologien, wie 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 Zielsystem, 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 Zielsystem (Ehmer and Khan, 2012).

2.3 Penetration Testing in Webanwendungen

Webanwendungen bieten ihren Nutzern eine dynamische und interaktive Umgebung, ohne dass man Anwendungen auf dem eigenen Rechner installieren muss. Im Gegensatz zu Desktop-Anwendungen erlauben Webanwendungen den gleichzeitigen Zugriff und Ver-

wendung mehrerer Nutzer. Der Zugriff findet über verschiedene Plattformen statt, wie Handys, Desktop, Tablet und Laptops. Webanwendungen bieten auch günstigere und schnellere Wartungsmaßnahmen an, da die Hardwarekonfiguration nicht ständig aktualisiert werden muss(Techtarget, 2019).

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

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

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

3 Durchführung der Aufgabe

In diesem Kapitel beschreiben wir konkret, wie die Arbeit sich 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 Aufgaben durchzuführen. In der folgenden Tabelle wird der Ablauf systematisch und undetailliert beschrieben. Im zweiten Teil dieses Kapitels geben wir eine ausführliche Beschreibung eines Projekts.

Bei WALLSEC hat jedes Projekt einen festgelegten Aufbau. Dieser kann in den folgenden Punkten zusammengefasst werden:

1. *Kick-off Meeting* mit dem Kunden, um grundsätzliche Informationen über die Anwendung zu bekommen

2. Definition des Testumfangs, wie Anmeldedaten, Rolle der Nutzer, Tenants und mögliche Einschränkungen

3. Durchführung von Tests nach einer vorgegebenen Checkliste

4. Dokumentation der durchgeführten Tests, der gefundenen Schwachstellen und Vorschläge zur Härtung der Anwendung

5. Abschlussmeeting mit dem Auftraggebern, um die Schwachstellen und deren Ausnutzung zu demonstrieren

3.1 Wöchentliche Zusammenfassung meines Praxissemesters

Auflistung der Aufgabe	
Woche	**Aufgabeschreibung**
1 - 2	Einarbeitung: • Installation von einer virtuellen Maschine für die Testumgebungen • Einführung in der Arbeitsablauf der Firma • Einführung, Installation und Einstellungen von Burp Suite • Einführung in ein laufendes Projekt, um über das Ablauf- und Dokumentationsverfahren zu lernen • Durchführung und Wiederholungen von einigen Tests, um mich an den gegebenen Tools zu gewöhnen • Teilnahme an einem Abschlussmeeting des laufenden Projekts, um das Verfahren und den Ablauf des Kundenkontakts kennenzulernen und später zu wiederholen
3 - 4	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an einer Versicherungsanwendung mit den obigen beschriebenen Schritten (3)
5 - 6	Weiterarbeit an der Installation, Einstellungen und Nutzung der Tools TheHive Project und Cortex. Bereitstellung von Skripts zum Herunterladen von statistischen Daten der Anwendungen und zur Automatisierung deren Nutzung.
7 - 8	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an einer Marketing-Webanwendung mit den oben beschriebenen Schritten (3)
9 - 10	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts in Netzwerk-Umgebungen mit den oben beschriebenen Schritten (3). Die durchgeführten Tests konzentrierten sich auf die Sicherheit eines Netzwerks in einer Cloud-Umgebung. Für dieses Projekt spielten die Tools nmap und Scout Suite eine wichtige Rolle, da das Ziel war, Hosts, Dienste und ihre Einstellungen und Schwachstellen zu erkennen
11	Start, Durchführung und Abschluss eines Pentestings von einer umfangreichen Webanwendung mit verschiedenen Tenants und Nutzerrollen für die Verwaltung von Business-Prozessen.
11 - 14	Start, Durchführung und Abschluss eines Pentestings wessen Aufbau auf Hypertext Transfer Protocol (HTTP) und GraphQL basiert war. Da diese beiden Technologien mir ganz neu waren, beschäftigte ich mich intensiv damit, sie in ihrem Aufbau und möglichen Schwachstellen zu erforschen.
15	Weiterarbeit an das Projekt mit TheHive Project und Cortex, indem verschiedene Skripte in Python geschrieben wurden, um Kommunikation zwischen TheHive Project und Cortex und anderen Anwendungen aufzubauen. Diese Skripte sollten auch Daten zu den Anwendungen importieren und exportieren.

Auflistung der Aufgabe	
Woche	**Aufgabeschreibung**
16 - 17	Start, Durchführung und Abschluss eines neuen Pentesting-Projekts an einer Webanwendung mit den oben beschriebenen Schritten (3). Der Kern dieser Anwendung ist die Erstellung und Verwaltung von Lieferkette-Prozessen von seits der Anbieter und der Verkäufer. Bei diesem Projekt war der Test darauf fokussiert, die festgelegten Schritte der Lieferkette zu erkennen und, als potentieller Angreifer, diese Schritte willkürlich zu steuern.
18	Das Projekt dieser Woche war eine Fortsetzung desjenigen von Woche 15. Der Kunde hat andere Teile der Anwendung zur Verfügung gestellt, sodass wir ihre Sicherheit überprüfen können. Die durchgeführten Tests konzentrierten auf strukturellen Ebenen der Anwendung, besonders auf der Herstellung von Verbindung mit Websockets.
19	Diese Woche war eine Fortsetzung an das Projekt von Woche 15. Hier wurde der Skript bis zu Ende geschrieben und bezüglich der Anmeldeart an der Umgebung des Beauftragter angepasst.
20	**Das Projekt dieser Woche war eine klassische Webanwendungen-Pentesting. Die Besonderheit bei diesem Projekt war meine Beteiligung an den drei letzten Schritten: Durchführung, Dokumentation und Abschlussmeeting mit dem Auftraggeber. Bei den vorherigen Projekten bekam ich ständig Hinweise von meinem Betreuer in der Firma. Diesmal gab er mir völlig Kontrolle über diesen drei erwähnten Schritten. Ich war sicher nicht allein und konnte immer Frage stellen, zu denjenigen Punkten, wo ich mich immer noch unsicher fühlte. Vor der endgültigen Abgabe haben wir zusammen eine Review gemacht, um sicher zu sein, dass es nichts fehlte. Ich kann diese Woche als wichtiger Zeitpunkt für meine Karrierelaufbahn klassifizieren: vor ungefähr vier Monaten war ich ein fast erfahrungsloser Praktikant und danach wurde mir vertraut, fast selbständig ein Penetration-Testing durchzuführen.**
21	TBD
22	TBD
23	TBD
24	TBD
25, 26, 27	Vertraglicher vereinbarter Urlaub. Diese Zeit benutzte ich, um dieses Bericht zu schreiben und zu verbessern.

3.2 Ausführliche Beschreibung eines Penetration Testing innerhalb des Praxissemesters

Vor der Durchführung eines jeden Tests sind WALLSEC und ihre Mitarbeiter dazu verpflichtet, eine Vertraulichkeitserklärung zu unterschreiben. Es dürfen Informationen weder über die Firma noch über die verwendeten Methoden dürfen in irgendeiner Form veröffentlicht werden. Aus diesem Grund werden die hier demonstrierten Methoden und Tests in der Test- und Lernumgebung Damn Vulnerable Web Application (DVWA) gezeigt. In den realen Tests verwenden wir ähnliche Methoden, manchmal mit mehr oder weniger Details, um die Sicherheit der Anwendungen zu überprüfen.

Da dieses Thema zu umfangreich für den Zweck dieses Berichts ist, demonstrieren wir in den nächsten zwei Unterkapiteln nur einige Methoden, die wir verwenden, um die Zielanwendung in ihrer Struktur zu prüfen und auszunutzen.

3.2.1 Sammlung von Informationen über das Zielsystem

Obwohl jede Webanwendung ihre eigenen Eingenschaften und Ziele hat, besitzen fast alle eine ähnliche Struktur und Aufbau. Bei jedem 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:

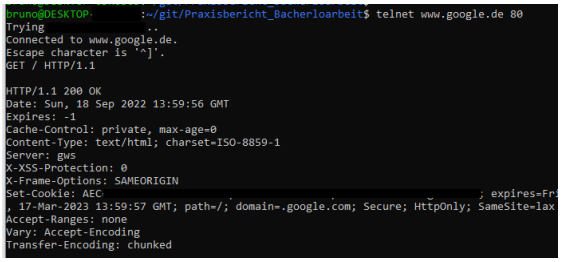

Abbildung 1: Banner Grabbing mithilfe vom Tool telnet

Eine Webanwendung ist eine Gruppierung von verschiedenen Verzeichnissen. Jedes Verzeichnis soll dem Nutzer eine Information oder Interaktion anbieten. Manche sind aber nicht für Nutzer gedacht, sondern dienen zur Verwaltung von Einstellungen. Da solche Verzeichnisse nicht direkt aufrufbar sind, benutzen wir andere Methoden, um zu finden, was die Entwickler im Hintergrund beibehalten wollten. Es gibt verschiedene Tools, die mithilfe von sogenannten *wordlists*, viele Anfragen an eine Anwendung schicken, um herauszufinden, was nicht direkt von dem Browser aufrufbar ist. Solche *wordlists* sind Textdateien, die häufig verwendete Wörter für Webanwendungen, Nutzername oder Passwörter beinhalten. Da viele Webanwendungen ähnliche Strukturen haben, ist auch meistens erwartet, dass gewöhnliche Wörter 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 Anfragen geschickt, jede mit einem verschiedenen Wort, um zu sehen, welche positive Antworten liefern. Das folgende Bild zeigt die Durchführung und das Ergebnis des Scanverfahrens mithilfe des Tools Dirbuster, Gobuster, usw.:

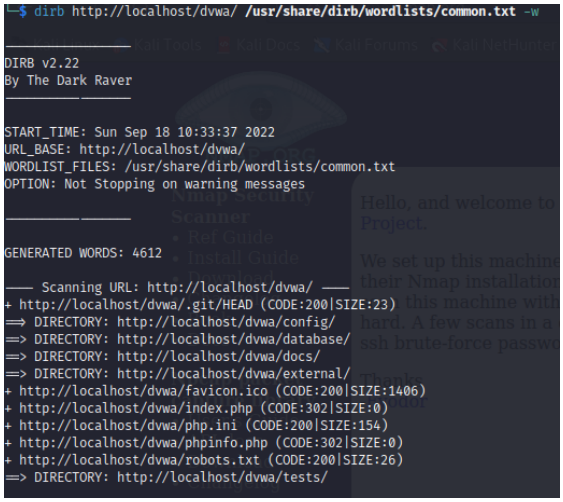

Abbildung 2: Brute force Scan für Verzeichnisentdeckung

Der nächste Schritte wäre eine manuelle Analyse des entdeckten Materials, um zu prüfen, ob sensitive Informationen ausgeliefert wurden. Falls ja, würden wir dann versuchen diese Schwachstelle, zu erkennen und auszunutzen.

Ein Netzwerk-Scan ist auch eine häufig verwendete Methode, um Dienste innerhalb des zu testenden Zielsystems zu finden. Dieser Scan schickt verschiedene Anfragen an das Ziel und so können wir die Reaktion des Zielsystems beobachten. Während wir uns bei dem ersten Scan auf die Webanwendung fokussierten, bearbeiten wir hier eine Ebene, die nicht für die gewöhnliche Nutzung gedacht ist. Unser Fokus liegt auf dem Server, wo die Anwendung läuft. Dafür testen wir die sogenannten Ports. Aus diesem Scan lassen sich meistens viele nützliche Informationen herausfiltern, wie z.B. das Betriebssystems, auf dem die Webanwendung läuft, Name und Versionen der existierenden Dienste. Mit diesen Informationen ist dann es möglich zielgerichtete Angriffe vorzubereiten, um Schwachstellen auszunutzen.

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

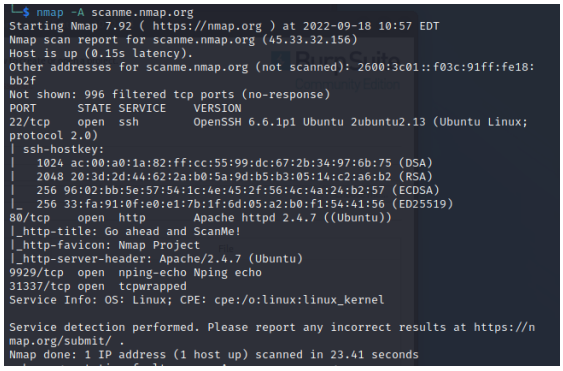

Abbildung 3: Brute force Scan für Verzeichnisentdeckung

Aus diesem Scan erfahren wir welches Betriebssystem und welche Version die Webanwendung benutzt. Auch wenn solche Versionen gegen Angriffe geschützt sind, ist es unsere Aufgabe den Kunden zu informieren, dass sensitive Informationen für alle sichtbar sind. Ein böswilliger Nutzer könnte diese Informationen nutzen, um eine Schwachstellen in dieser Anwendungen zu entdecken und diese auszunutzen.

3.2.2 Ausnutzung der Zielanwendung

Nachdem die vorherigen Scans durchgeführt 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 versteckten Daten oder unerlaubten Aktionen ein Angreifer durchführen kann, um die CIA der Anwendung zu verletzen. Für die folgenden Tests benutzen wir unter anderem auch das Tool Burp Suite.

Unser erster Test soll überprüfen, ob es möglich ist, in ein Eingabefeld Daten einzutragen und das normale Verhalten der Anwendung zu ändern. Wir versuchen eigenen Code hinzuzufügen und wenn uns das gelingt, fügen wir weiteren Code hinzu, um Daten von Nutzer zu stehlen oder das normale Verhalten der Anwendung zubeschädigen. 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.

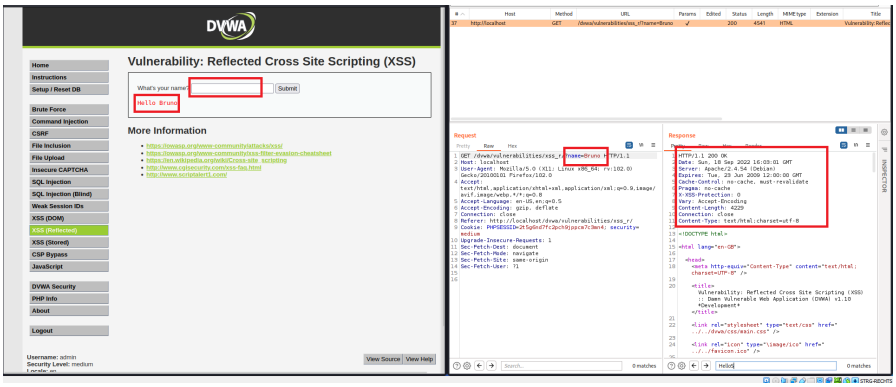

Abbildung 4: Beobachtung der Anwendung unter normale Nutzung

Abbildung 4 zeigt den ersten Test. Aus dieser Aufnahme der Anfrage können wir sehen, dass die Nutzereingabe direkt in dem Browser stattfindet. In der Antwort sehen wir Informationen über den Aufbau der Anwendung und wie sie auf unsere Anfrage reagiert. Auf dem nächsten Bild versuchten wir bösartigen Code hinzuzufügen, um den normalen Ablauf der Anwendung zu verletzen. Dafür verwendeten wir Javascript Code. Wir konnten unseren Code direkt in die Anwendung, in einen selbst gebastelte Request oder in Burp Suite eingeben:

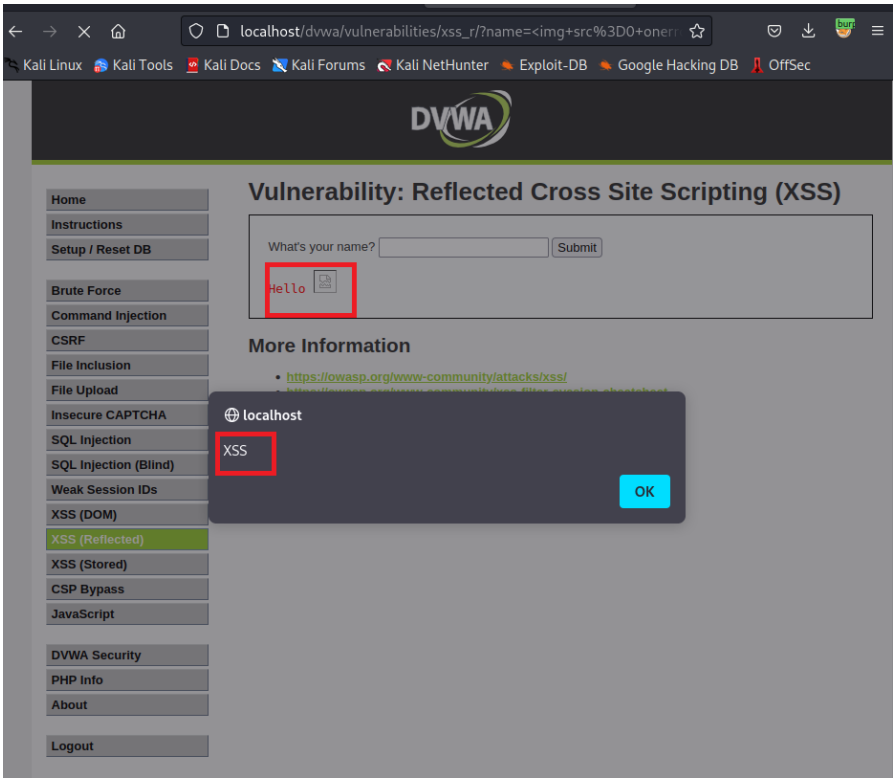

Abbildung 5: Einführung vom bösartigem Code und Beobachtung der Reaktion der Anwendung.

Für diesen Test haben wir den Code *<img src=0 onerror='alert("XXS")'\>* hinzugefügt. Das Ziel dieses Codes ist ein nicht existierendes Bild hinzuzufügen, 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 dem Nutzer zeigt, 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.

3.2.3 Kundebericht

Je nachdem wie lange das Projekt läuft, können wir mehr oder weniger zeitintensive Tests durchführen. Am Ende des Projekts präsentieren wir dem Auftraggeber in einem Meeting unser Ergebnis und liefern einen Bericht mit detaillierten Informationen über die gefundenen Schwachstellen und die dazu verwendeten Methoden. Dieses Bericht wird so geschrieben, dass die Nichtfachleute es verstehen können.

In den ersten Abschnitten erklären wir mit wenigen technischen Begriffen, welche Tests durchgeführt wurden. In den folgenden Kapiteln erklären wir mit mehr Einzelheiten und technischen Details, wie wir zu unserem Ergebnis kamen. Anschließend geben wir Vorschläge für die Verbesserung der Sicherheit der Webanwendung und zum Schluss geben wir mithilfe von CVSS oder einer anderen von dem Kunden ausgewählter Metrik eine allgemeine Bewertung.

4 Fazit

Das Praxissemester bat mir eine sehr gute Möglichkeit an, 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 Testing.

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 2022.

Christensson, P. (2006). Script definition. https://www.python.org/doc/essays/blurb/. Zugriff am 17te Oktober 2022.

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/Pruefungsordnung/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. 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. 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. 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.

Pimentel, V. and Nickerson, B. G. (2012). Communicating and displaying real-time data with websocket. *IEEE Internet Computing*, 16(4):45–53.
https://doi.org/10.1109/MIC.2012.64. Zugriff am 13t November 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.

Python (2022). What is python? executive summary.
https://www.python.org/doc/essays/blurb/. Zugriff am 18te Oktober 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.