Duale Hochschule Baden-Württemberg Mannheim

Projektarbeit

Automatisierte Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen

Studiengang Informatik

Studienrichtung Angewandte Informatik

Verfasser:	Joel Dag
Matrikelnummer:	4811224
Firma:	Atos Information Technology GmbH
Kurs:	TINF21 AI1
Studiengangsleiter:	Prof. Dr. Holger Hofmann
Abgabedatum:	06.10.2022
Bearbeitungszeitraum:	03.01.2022 - 06.10.2022
Firmenbetreuer:	Daniel von Schierstedt

Unterschrift Betreuer: _____

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Titel "*Konzeptentwicklung und Implementierung einer Pausenfunktion für den T4TC-Runner*" selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Mannheim, 05.10.2022
Joel Dag

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

T4TC	Testautomation for Teamcenter
PLM	Product Liefcycle Management
GUI	Graphical User Interface
UI	User Interface
XML	Extensible Markup Language
HTML	Hypertext Markup Language

1 Einleitung

Die Bearbeitung dieser Arbeit findet im betrieblichen Umfeld der Firma Atos Information Technology GmbH am Standort Paderborn in der Abteilung Offensive-Defensive Security (ODS) im Bereich des Penetration Testings statt.

1.1 Einordnung des Themas

Im Zuge der fortschreitenden Digitalisierung und dem zunehmenden Gebrauch von Internet-basierten Anwendungen, haben IT-Systeme einen zentralen Stellenwert in vielen Bereichen des täglichen Lebens und der Geschäftswelt erlangt. Sie ermöglichen die Verwaltung von Geschäftsprozessen, die Kommunikation mit Kunden und die Speicherung von wichtigen Daten.
Allerdings erhöht die zunehmende Verbreitung von IT-Systemen auch die Gefahr von Angriffen. Angreifer nutzen häufig Schwachstellen aus, um unbefugte Zugänge zu erhalten und sensiblen Daten für ihre eigenen Zwecke zu missbrauchen.
Häufig werden dabei Schäden verursacht, die Kosten in Millionenhöhe zur Folge haben. Durch die starke Abhängigkeit von IT-gestützen Systemen werden sogar ganze Infrastrukturen aufgrund von Cyberangriffen stillgelegt.

Um solche Systeme zu Schützen und mögliche Sicherheitslücken aufzudecken, werden in der Regel Penetration Tests (Pentests) aus der Sicht eines Angreifers durchgeführt. Die Pentester nutzen dabei sowohl manuell, als auch automatisierte Vorgehensweisen.

Diese Arbeit beschäftigt sich mit dem Thema der automatisierten Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen. Hierbei werden verschiedene Tools und Techniken untersucht und evaluiert, die verwendet werden um Schwachstellen in Web-Applikationen zu erkennen und auszunutzen.

Dadurch soll ein Überblick über verschiedene Methoden und Tools zur Erkennung von Schwachstellen in Web-Applikationen und dessen Vor- und Nachteile geschaffen werden. Auf Basis der Untersuchungen dieser Arbeit soll eine Empfehlung für eine effektive Vorgehensweise bei der automatisierten Erkennung und Ausnutzung von Schwachstellen abgegeben werden.

1.2 Problemstellung

Die monetäre Kosten, die jährlich durch Cyberattacken entstehen, sind beträchtlich und steigen stetig an. Sie sind mittlerweile vergleichbar mit Ausgaben ganzer Staatshaushälter einzelner Länder. Der Einfluss von Cyberattacken umfasst nicht nur Einzelpersonen, Unternehmen und öffentliche Einrichtungen, sonder auch die gesamte Infrastruktur, auf der zahlreiche Prozesse unserer modernen Gesellschaft beruhen.

In den letzte 10 Jahren hat sich de Zahl der Cyberangriffe in Deutschland rapide erhöht. Laut Bundesamt für Sicherheit in der Informationstechnik (BSI) gab es im Jahr 2018 mehr als 400.000 gemeldete Cyberangriffe, was eine Steigerung von 37% im Vergleich zum Vorjahr entspricht. [1]
Auch die Schäden, die durch Cyberangriffe verursacht werden, haben zugenommen. Laut der aktuellen Ausgabe der Cost of Data Brach Study betrugen die durchschnittlichen Kosten einer Datenpanne in Deutschland 3,62 Millionen US-Dollar. [2]
Dabei ist wichtig zu beachten, dass viele Cyberangriffe nicht gemeldet werden und daher die tatsächlichen Zahlen wahrscheinlich höher sind. Außerdem haben sich die Methoden, mit denen Cyberkriminelle Angriffe durchführen, in den letzten Jahren erheblich fortentwickelt, was es für Unternehmen und Organisationen immer schwieriger macht, ihre Systeme zu schützen.

Bei der Erkennung von Schwachstellen spielen Pentesting-Tools eine besonders wichtige Rolle. Unter der zahlreichen Auswahl verschiedener Tools, die bestimmte Funktionen automatisieren, verliert man schnell den Überblick. Daher ist nur schwer ersichtlich, welche Praktiken und zugehörige Tools geeignet sind um automatisiert Schwachstellen in Web-Applikationen zu erkennen und auszunutzen.

1.3 Ziel der Arbeit

Da alle Tools unterschiedliche Funktionalitäten und Ansätze bieten Schwachstellen zu erkennen, ist es wichtig eine umfassende Untersuchung dieser Tools durchzuführen, um so verschiedene Praktiken bei der Erkennung und Ausnutzung von Schwachstellen zu evaluieren.
Das Ziel dieser Arbeit ist es, ein Verständnis dafür zu entwickeln, wie diese Techniken funktionieren und wie sie optimiert werden können. Dabei konzentriert sich das Vorhaben

auf die Identifikation der besten Methoden und Tools, um Schwachstellen zu erkennen und zu beseitigen. Durch die Untersuchungen dieser Arbeit sollen praktische Lösungen für die Automatisierung der Schwachstellenanalyse in Web-Applikationen bereitgestellt werden, um so die Sicherheit dieser Anwendungen für die Nutzer zu erhöhen. Zudem sollen neue Ansätze und Methoden vorgestellt werden, die dabei helfen können, Schwachstellen effizienter zu erkennen und zu beseitigen.
Insgesamt soll dadurch die zielgerichtetere Erkennung von Schwachstellen in Web-Applikationen erhöht werden, wodurch die Schadensanfälligkeit von Web-Applikationen verringert werden soll.

1.4 Vorgehensweise

Um eine systematische Vorgehensweise zu garantieren, wird zunächst eine Literaturrecherche durchgeführt, wodurch eine Grundlage für den weiteren Verlauf der Arbeit geschaffen wird. Hierbei wird auch der Grundablauf eines Penetration Testers und die Integration von passenden Tools in die Vorgehensweise thematisiert. Für die Recherche werden sowohl wissenschaftliche Publikationen als auch praxisorientierte Ressourcen herangezogen.

Danach wird eine Auswahl der zu testenden Tools und Techniken. Es werden dazu Kriterien, eine Vorgehensweise und eine passende Testumgebung festgelegt, um die Untersuchung unter möglichst realen Bedingungen vorzunehmen.
Anschließend wird die Analyse der unterschiedlichen Tests unter Berücksichtigung der Literaturrecherche durchgeführt.
Die hier entstandenen Ergebnisse werden daraufhin durch die aufgestellten Kriterien evaluiert, wodurch eine umfassende Bewertung der verschiedenen Methoden und Tools ermöglicht wird.

Abschließend werden die Ergebnisse der Literaturrecherche und der praktischen Evaluierung zusammengeführt, um eine detaillierte Validierung der verschiedenen Tools und Techniken zu ermöglichen. Hierbei werden auch mögliche Implikationen für die Praxis diskutiert, um Empfehlungen für die Anwendung von Tools und Techniken zur automatisierten Erkennung von Schwachstellen in Web-Applikationen abzugeben.

1.5 Aufbau der Arbeit

Die folgende Arbeit gliedert sich in sieben Bestandteile. Dabei bildet dieser Abschnitt die Einleitung. Hier wird ein Überblick zu dem Thema angefertigt und die Zielsetzung der Arbeit unter Berücksichtigung der Problemstellung definiert.

Diese Arbeit bewegt sich im praktischen Umfeld des Penetration Testing und der Schwachstellenanalyse von Web-Applikationen. Daher wird im Teil zwei der Arbeit zunächst eine wissenschaftliche Grundlage zu diesen Thematiken geschaffen.

Das dritte Kapitel beschäftigt sich mit der Auswahl der zu analysierenden Tools und Techniken. Außerdem wird hier eine Vorgehensweise und eine Testumgebung festgelegt, um optimale Bedingungen und eine systematische Vorgehensweise zu garantieren.

Kapitel vier befasst sich mit der Analyse der ausgewählten Tools, die zur automatisierten Erkennung von Schwachstellen in Web-Applikationen eingesetzt werden. Hierbei werden Funktionalitäten, Vorgehensweisen und Ergebnisse untersucht und verglichen.

Im darauffolgenden Teil, dem Fünften, wird sich mit der automatisierten Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen unter Verwendung von Künstlicher Intelligenz beschäftigt. Dabei werden jüngste Entwicklungen aus der Forschung berücksichtigt.

Das sechste Kapitel thematisiert die Evaluierung der verschiedenen Tools und Techniken und dient dazu, Stärken und Schwächen der einzelnen Ansätze herauszuarbeiten. Hierbei wird auch ein Vergleich der verschiedenen Tools und Techniken anhand festgelegter Kriterien durchgeführt.

Im letzten Teil der Arbeit werden die Ergebnisse zusammengefasst und evaluiert. Darüber hinaus wird eine Empfehlung für die automatisierte Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen abgegeben.

2 Grundlagen

Um eine wissenschaftliche Grundalge dieser Arbeit zu schaffen, wird zunächst das Thema Pentesting generell betrachtet, in dessen Umfeld sich die Arbeit in der Praxisphase befindet. Dann werden verschiedene Vorgehensweisen und Methoden thematisiert, die zur automatisierten Schwachstellenerkennung verwendet werden.

2.1 Penetration Testing

Pentration Testing beschreibt den autorisierten und legalen Versuch Computer-Systeme ausfindig zu machen und dessen Schwachstellen erfolgreich auszunutzen, um diese Systeme sicherer zu machen.
Der Prozess umfasst sowohl die Identifizierung von Schwachstellen, sowie der Bereitstellung von Proof-of-Concept (POC) Angriffen, um zu demonstrieren, dass die Schwachstellen existent sind.

Ein ordnungsgemäßer Penetrationtest endet mit einer spezifischen Empfehlung zur Behebung der während des Tests entdeckten Probleme. Insgesamt wird dieser Prozess dazu verwendet, um Computer und Netzwerke gegen zukünftige Angriffe zu sichern. [B1]

Pentester nutzen Tools und Techniken, die auch von böswilligen Angreifern (black hats) verwendet werden, um Schwachstellen in Systemen und Netzwerken zu finden. Dies beinhaltet unter anderem die Überprüfung von Passwortschutz, Netzwerksicherheit und Anwendungssicherheit. [B1]

Darüber hinaus ist Pentesting ein wichtiger Teil des Risikomanagement-Prozesses, da es Unternehmen ermöglicht, ihre Systeme auf potenzielle Bedrohungen zu überprüfen und Gegenmaßnahmen zu ergreifen, bevor Angriffe stattfinden. Jedoch handelt es sich beim Pentesting um eine simulative Übung und bietet daher keine Garantie für die tatsächliche Sicherheit eines Systems. [8]

2.1.1 Pentesting Frameworks

Um bei einem Penetration Test erfolgreiche Ergebnisse Bei einem Penetration Test spielt die Methodik eine entschediden Ohne die Verwendung einer festen Methodik

Beim Penetration Testing spielt eine gut definierte Methodik eine Schlüsselrolle beim Erreichen von erfolgreichen Ergebnissen. Diese Ergebnisse werden sind jedoch entscheidend um Daten, Anwendungen und die zugrunde legende Infrastruktur zu schützen. Ohne ein etablierte Methodik während der Durchführung eines Pentests, kann es schwierig sein Lokalisieren von Schwachstellen schwierig sein oder sogar ein falsches Sicherheitsgefühl der Anwendung erwecken.
Nach Willhelm P. sind Penetration Tests Projekte, welche durch effektive, wiederholbare und qualitätssteigenderen Prozessen durchgeführt werden sollten, um dadurch die Qualität dieser Tests deutlich anzuheben.
Hierbei spielt die Verwendung einer festgelegten Methodik eine wesentliche Rolle.

2.1.2 Standards und Vorgehensweisen

Wie die meisten Prozesse kann auch ein Penetration Test in einzelne Schritte heruntergebrochen werden. Um möglichst effektive Ergebnisse zu garantieren, halten sich die meisten Pentester Standards und Vorgehensweisen, die eine systematische Überprüfung von Systemen erleichtern.

Je nach Detailorientiertheit wird ein Pentration Test daher in vier bis sieben Schritte eingeteilt. Patrick Engebretson unterteilt die Methodik eines Penetrationstests in vier grundlegende Schritte: [B1]

Der erste Schritt beschreibt die Reconnaissance, also das Sammeln von Informationen und Auskundschaften eines Systems. Hierbei wird zunächst versucht jede mögliche Information über eine System zu erhalten und abzuspeichern. Die in dieser Phase gesammelten Informationen sind für den weiteren Verlauf des Penetrationtests entscheidend. [B1]
Man unterscheidet dabei zwischen aktiver und passiver Reconnaissance. Bei der aktiven Reconnaissance wird direkt mit dem Zielsystem interagiert, um bestimmte Informationen zu erhalten. Hierbei werden aktiv Anfragen an das Zielsystem gesendet. Dabei ist nicht auszuschließen, dass das Zielsystem durch die aktive Reconnaissance alarmiert wird.
Bei der passiven Reconnaissance wird nicht aktiv mit dem Zielsystem interagiert. Hierbei

werden öffentliche Inhalt verwendet, um Informationen über das Ziel zu sammeln. Das Zielsystem wird dabei nicht alarmiert. [B1] [B4]

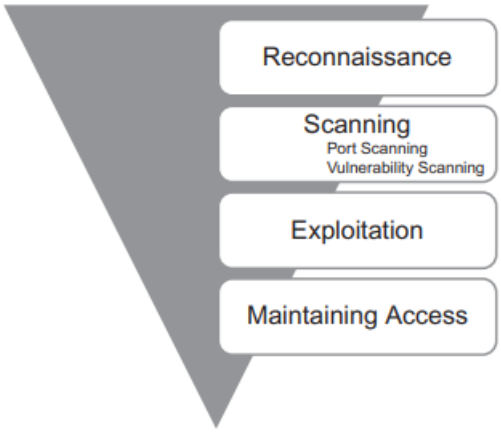

Abbildung 2.1: Phasen eines Penetrationtests
[B1]

Das in der Abbildung 1.1 abgebildete Dreieck symbolisiert dabei, dass eine breit angesammelte Menge an Informationen kombiniert mit der passenden Abstraktion über die weiteren Phasen schließlich zu einem erfolgreichen Pentest führt.

Der zweite Schritt, das Scanning, kann in zwei weitere Bestandteile aufgeteilt werden. Mit den in der Reconnaissance herausgefundenen IP-Adressen kann nun ein Port-Scan durchgeführt werden, um offene Ports und potentielle Services zu identifizieren. [B2]
Als nächstes wird das Vulnerability Scanning ausgeführt. Dabei werden Schwachstellen in der Software und in den Services des Zielsystems lokalisiert und identifiziert. [B3]

!! Den Text hier unten nochmal durchchecken, der ist später dazu gekommen !!

Anhand der Ergebnisse unserer Scans können wir nun mit der Exploitation-Phase, also der Ausnutzung von Schwachstellen beginnen. Da wir nun genau wissen welche Ports und Services Angriffsfläche bieten und sogar die Schwachstellen kennen, können wir nun beginnen unser Zielsystem systematisch anzugreifen. [B1]
Beim exploiting gibt es zahlreiche Ansätze und Vorgehensweisen. Hauptziel ist jedoch immer, Zugriff auf das Zielsystem zu erlangen und seine Zugriffsrechte auf dem System so zu eskalieren, dass man Administratorrechte auf dem Zielsystem erlangt. [B4]

Die letzte Phase die durchschritten wird ist die Maintaining Access Phase. Haufig wird bei einer Ausnutzung einer Schwachstellen nur temporärer Zugriff auf das Zielsystem möglich. Da viele Sicherheitslücken keinen dauerhaften Zugriff erlauben, wird während des temporären Zugriffs im Zielsystem versucht, eine weiter Möglichkeit zu schaffen auf das Zielsystem zu gelangen. [6]

Formal wird einem Pentest in der Regel noch eine letzte Phase, das Reporting hinzugefügt, an welchem oftmals die Qualität eines Pentests gemessen wird. Ein Report beinhaltete alle relevanten Informationen über den vollzogenen Pentest. Hierbei werden Sicherheitslücken aufgedeckt und an den nötigen Stellen wird zusätzlich erläutert, welche Schritte im Pentest stattgefunden haben. Gegebenenfalls werden vom Pentester noch Vorschläge gemacht, wie die Sicherheitslücken ausgebessert werden können. [B1]

2.2 XY

Weiterer wissenschaftlicher Input zu Schwachstellen in Web Applikationen z.B. OWASP top 10

3 Auswahl der Tools und der Umgebung

In diesem Kapitel ...

- Hier kurz einmal Erklären welche Tools verwendet werden und evtl. 1-2 Sätze zu jedem.

- Erläutern wie nach welchen Kriterien die Tools verglichen werden

- Erläutern, dass der Juice Shop als Zielsystem verwendet wird. Oder doch vlt WAVSEP ? - WAVSEP Ist denke ich bessser, weils extra dafür gemacht ist + eine umfangreichere Sammlung an Schwachstellen hat https://github.com/sectooladdict/wavsep `https://owasp.org/www-project-benchmark/`

http://projects.webappsec.org/w/page/13246986/Web

`https://norma.ncirl.ie/4165/1/mandarprashantshah.pdf`

Scanner: nmap, sslscan, nikto nmap sslscan fuff sql map nikto nuclei burp community, metasploit

Unterschiedung Kategorisierung der Tools

wstg owasp, vergleichbar gegenüber halten

3.1 Testkriterien

Um die Auswertung und Validierung der Tests möglichst strukturiert zu gestalten werden nachfolgend Kriterien festgelegt, nach denen die einzelnen Tools und Techniken verglichen werden. Hierfür werden bereits vorhandene und gängige Kriterien mit zusätzlich weiteren wichtigen Kriterien kombiniert.

Paar Kriterien (Brainstorming): `https://norma.ncirl.ie/4165/1/mandarprashantshah.pdf`

- **(Scanning) Speed**

- **Vulnerability Detection Rate**
- **True Positive and False Positive reported**
- **Einfache Bedienung und Nachvollziehbarkeit**
  dd

3.1.1 OWSAP Benchmark test

Der OWASP Benchmark test ist eine Java basierte open source Tetsumgebung zur Erkennung von Schwachstellen. OWASP benchmark ist eine bewährte Testumgebung, die regelmäßig geupdatet wird. Die Testumgebung umfasst dabei über 2740 potentielle Schwachstellen.

Um den Testdurchlauf möglichst strukturiert und systematisch zu gestalten, werden nachfolgend Kriterien aufgestellt, nach denen die einzelnen Tools und Techniken verglichen werden.

In den bisherigen Abschnitten dieser Arbeit haben wir uns bereits mit den Grundlagen von Pentesting, sowie der generellen Vorgehensweisen und Pentesting Frameorks beschäftigt. Unter zahlreichen möglichen Zielsystem hat sich das Framework OWASP Benchamrk zum Testen und Vergleichen verschiedener Pentesting Tools als sehr geeignet bewährt. OWASP Benchmark bietet dazu zahlreiche Testcases und wird immernoch aktiv geupdatet.

Eines dieser Frameworks

3.2 Auswahl der Tools

Kategorisierung: In dieser Arbeit werden

3.3 Zielsystem

Um die Zuverlässigkeit während dem Vergleich der einzelnen Tools zu gewährleisten, ist es wichtig die richtige Testumgebung zu wählen. Nachfolgend werden einige Kriterien beschrieben, die eine passende Testumgebung benötigt:

- **Umfassende Abdeckung**
  Die Testumgebung sollte eine Vielzahl bekannter Schwachstellen und Angriffsszenarien bereitstellen, um ein umfassendes Testen der Tools zu ermöglichen.
- **Konsistenz**
  Damit die Unterschiede, sowie Vor- und Nachteile der einzelnen Tools aufgedeckt werden können, sollte die Testumgebung möglichst einheitliche Verfahren und Messmethoden bereitstellen.
- **Realitätsnähe**
  Die zu untersuchenden Tools begleiten zahlreiche Pentester täglich in realen Szenarien. Daher sollen die in der Testumgebung bereitgestellten Angriffsszenarien und Schwachstellen möglichst realitätsnah sein.
- **Einfache Bedienung und Nachvollziehbarkeit**
  Die Testumgebung sollte eine einfache Bedingung und eine Dokumentation bereitstellen um eine schnelle und problemlose Verwendung zu ermöglichen. Außerdem sollten die Ergebnisse und Methodiken seitens der Testumgebung nachvollziehbar sein.

Aufgrund der aufgestellten Kriterien wird das Web Application Vulnerability Scanner Evaluation Project (WAVSEP) als adäquate Testumgebung gewählt. WAVSEP ist eine bewährte Testumgebung, die auf den Vergleich von Pentestools ausgelegt ist.
Dabei bietet WAVSEP ein umfassendes Set an passende Angriffsszenarien, die optimale Bedingungen schaffen, um die Genauigkeit und Zuverlässigkeit von Pentesttools direkt miteinander zu vergleichen. Dabei verwendet WAVSEP einheitliche und nachvollziehbare Messmethoden, die den direkten Vergleich einzelner Tools erleichtern.
WAVSEP ist daher eine hervorragende Wahl für die Bewertung von Penetrationstools.

3.4 Testumgebung

Die Durchführung der Tests findet auf einer Kali Linux VW mit einer Windows host Maschine statt. Um das Testen zu ermöglichen muss zunächst alle nötige Software installiert werden, um das Zielsystem schließlich angreifen zu können

4 Analyse

In diesem Kapitel ...

5 Schwachstellenerkennung unter Verwendung von künstlicher Intelligenz

In diesem Kapitel:

- kurzer allgemeiner Text, über das Thema

- Beschreibung/Erklärung von http://repository.kpi.kharkov.ua/handle/KhPI-Press/54956

- Etwas über zukunftsfähigkeit und Vorteile/Nachteile schreiben

Bevor in Kapitel sechs eine audäquate Evaluierung und Auswertung stattfinden soll, wird vorher noch eine weitere Methodik/Technik untersucht, die unter Verwendung von künstlicher Intelligenz Schwachstellen erkennt und versucht diese auszunutzen. Diese Methodik gilt im Vergleich zu den bisher vorgestellten Tools und Techniken zu den eher jüngeren Technologien.

6 Evaluierung und Auswertung

In diesem Kapitel ...

- Auswertung der Anlalyseergebnisse
- Vergleiche zwischen den Tools
- Vergleiche zwischen manuell und automatisiert
- Vor und Nachteile einzelner Dinge
- Rückschlüsse ziehen auf die vorherigen Kapitel

7 Abschluss

A Literaturverzeichnis

[B1] Engebretson, P. (2013). The basics of hacking and penetration testing: ethical hacking and penetration testing made easy. Elsevier.
http://dspace.fudutsinma.edu.ng/xmlui/bitstream/handle/123456789/845/ebooksclub.org__The_Basics_of_Hacking_and_Penetration_Testing__Ethical_Hacking_and_Penetration_Testing_Made_Easy.pdf?sequence=1

[B2] Yaworski, P. (2018): Web Hacking 101. No Starch Press.
https://digtvbg.com/files/books-for-hacking/Web%20Hacking%20101%20-%20How%20to%20Make%20Money%20Hacking%20Ethically%20by%20Peter%20Yaworski.pdf

[B3] Mastering Defensive Security by Cesar Bravo, Darren Kitchen.
https://learning.oreilly.com/library/view/mastering-defensive-security/9781800208162/B16290_FM_Final_JC_ePub.xhtml

[B4] Penetration Testing - A Hands-On Introduction to Hacking

A.1 Quellenerzeichnis

[1] Siemens Indusrty Software Inc.
Intelligentere Entscheidungen, bessere Produkte durch umfassendes PLM
https://www.plm.automation.siemens.com/de_de/Images/tc_overview_tcm73%-62348.pdf

[2] Cost of data breach 2022
https://www.ibm.com/reports/data-breach

[3] Patrick Engebretson "The Basics of hacking and penetration testing"
http://dspace.fudutsinma.edu.ng/xmlui/bitstream/handle/123456789/845/ebooksclub.org__The_Basics_of_Hacking_and_Penetration_Testing__Ethical_Hacking_and_Penetration_Testing_Made_Easy.pdf?sequence=1

[4] Deep Learning und Pentesting
http://repository.kpi.kharkov.ua/handle/KhPI-Press/54956

[5] CHIEM TRIEU PHONG, A Study of Penetration Testing Tools and Approaches
https://openrepository.aut.ac.nz/bitstream/handle/10292/7801/ChiemTP.pdf?sequence=3&isAllowed=y

[6] A Study on Penetration Testing Process and Tools
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8378035&casa_token=XNKWbXqAGogAAAAA:zi_TswzVWQeWmdEiCATwmOxCQ_L8-exnSsgQQQBM3ahGCG2iCg5ciNeyWkrBJxe6sd
tag=1

[7] Shah, M. P. (2020). Comparative analysis of the automated penetration testing tools (Doctoral dissertation, Dublin, National College of Ireland).
https://norma.ncirl.ie/4165/1/mandarprashantshah.pdf

[8] Penetration Testing in a Web Application Environment
http://www.diva-portal.org/smash/get/diva2:356502/fulltext01.pdf

Das hier später noch Aufräumen und Citavi oder so verwedent

weitere Quellen

https://sectooladdict.blogspot.com/

https://owasp.org/www-community/Vulnerability_Scanning_Tools

A.2 Bildverzeichnis

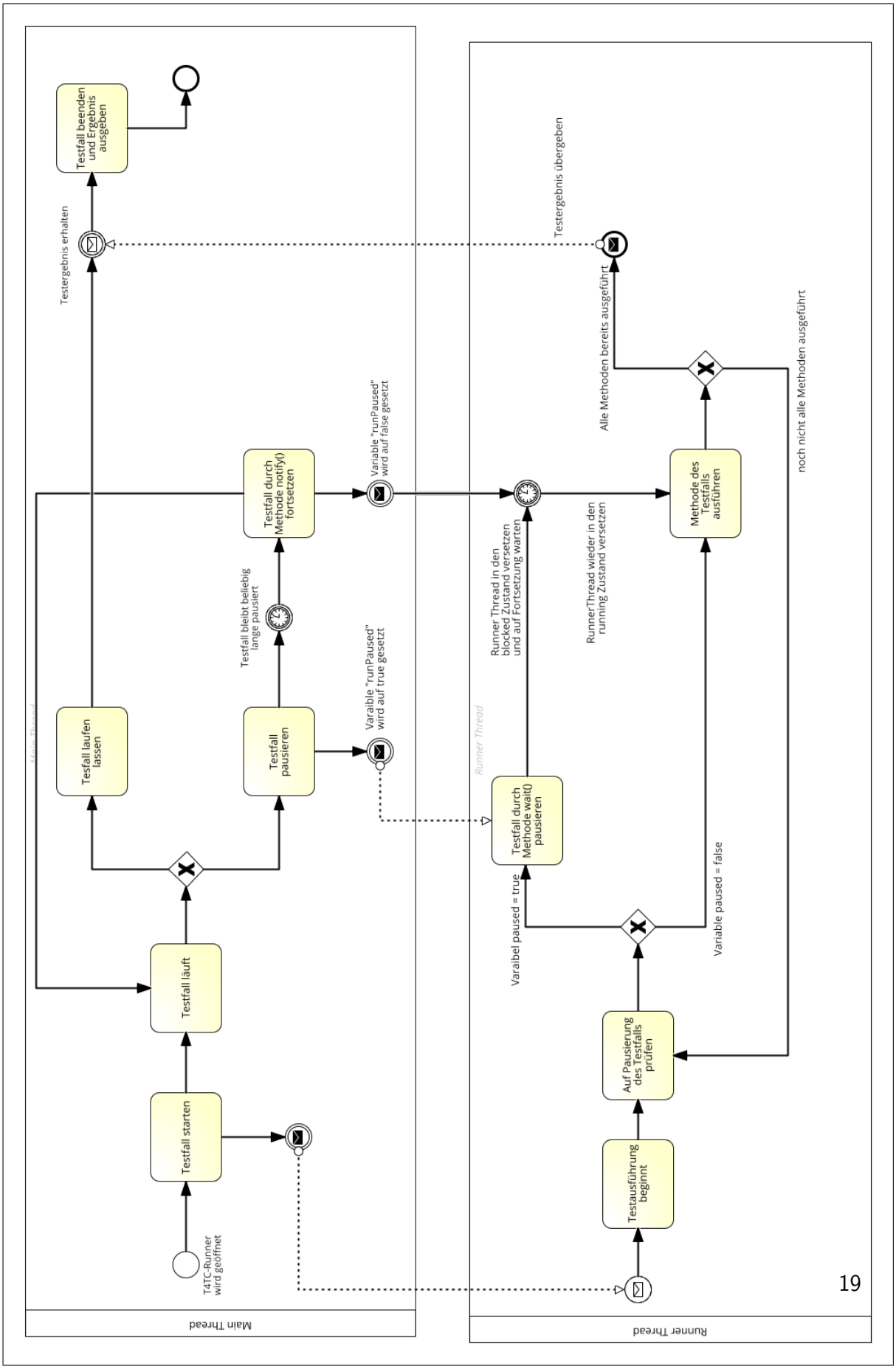

Abbildung A.1: vereinfachtes Prozessdiagramm der neuen Testausführung
[Quelle: eigene Darsetllung]