HOCHSCHULE MITTWEIDA
University of Applied Sciences

Evaluierung von KI-basierten Modellen zur automatisierten Schwachstellenanalyse im Rahmen von Penetrationstests

XXX

Problemstellung & Motivation (1/2)

Problemstellung

- Pentests sind etabliert, aber teuer & aufwendig
  - hoher Zeitaufwand
  - Bedarf an Fachwissen
  - kostenintensiv
- Wachsende Angriffsflächen
  - Cloud
  - Microservices
  - DevOps erhöhen Komplexität und Angriffsrisiko
- Limitierungen klassischer Methoden
  - rein manuelle Tests skalieren nicht mehr
  - Ergebnisse abhängig von Erfahrung einzelner Tester

Fachkräftemangel als Motivation

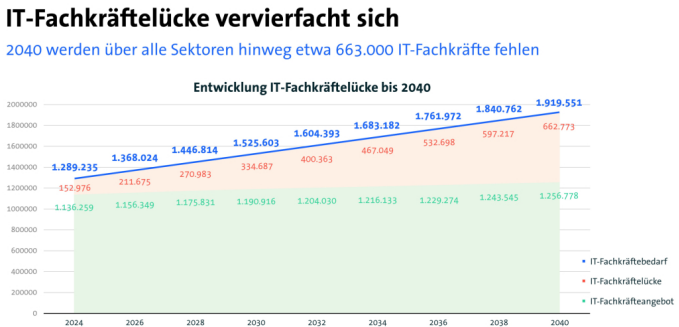

Abbildung: Bitkom: IT-Fachkräftelücke bis 2040

Ziel der Arbeit

Forschungsfrage: Wie können KI-Tools Sicherheitsexperten in Penetrationstests unterstützen und entlasten?

Vorgehen:

- Untersuchung von drei Tools:
  - RamiGPT (Privilege Escalation)
  - PentestGPT CLI (strukturierte Webtests)
  - PentestGPT Web (dialogorientierte Tests)

Teilziele:

- Entwicklung eines praxisnahen Bewertungsrahmens
- Durchführung reproduzierbarer Praxistests
- Vergleich: Stärken, Schwächen & Einsatzpotenziale

Grundlagen – Penetrationstests

- **Definition:** Gezielte Simulation von Angriffen auf IT-Systeme
- **Zweck:** Schwachstellen frühzeitig finden & Sicherheitsniveau bewerten
- **Abgrenzung:** Unterschied zu reinen Scans → kreatives, manuelles Vorgehen
- **Arten:**
  - Black-Box (keine Vorkenntnisse)
  - Grey-Box (teilweise Infos)
  - White-Box (vollständige Infos)

Grundlagen – Phasenmodell nach BSI

- **Vorbereitung:** Scope, Ziele & Rahmenbedingungen definieren
- **Informationsbeschaffung:** Scans, OSINT, Reconnaissance
- **Analyse & Bewertung:** Identifikation & Bewertung möglicher Schwachstellen
- **Exploitation:** Gezielte Ausnutzung, um Ausnutzbarkeit realistisch einzuschätzen
- **Abschluss & Reporting:** Ergebnisse dokumentieren & Empfehlungen ableiten

Grundlage (1/4)

Penetrationstests

- Ziel: Schwachstellen finden, bevor Angreifer sie ausnutzen
- BSI-Phasenmodell:
  1. Vorbereitung
  2. Informationsbeschaffung
  3. Analyse & Bewertung
  4. Exploitation
  5. Abschluss & Reporting

Abbildung: Das Bild der Danke-Seite

Grundlagen – Typische Schwachstellen (OWASP Top 10)

- **A01 – Broken Access Control**
  - Fehlende oder fehlerhafte Zugriffsbeschränkungen
  - z. B. Manipulation von JWT, IDOR
- **A02 – Cryptographic Failures**
  - Unsichere oder falsch eingesetzte Verschlüsselung
  - z. B. schwache Hashes, Klartextübertragung
- **A03 – Injection**
  - Unsichere Eingabevalidierung → Angriffe möglich
  - z. B. SQL Injection, Cross-Site Scripting (XSS)

Grundlagen – Privilege Escalation

- **Ziel:** unberechtigter Zugriff auf höhere Rechte (Admin/Root)
- Häufig in Post-Exploitation-Phase
- **Typische Techniken (nach MITRE ATT&CK):**
  - Exploitation von Systemschwachstellen (T1068)
  - Missbrauch von Sudo/SetUID (T1548)
  - Manipulation von Zugriffstokens (T1134)
  - Nutzung gültiger, privilegierter Accounts (T1078)

Grundlagen – Künstliche Intelligenz & LLMs

- **KI:** Mustererkennung, Automatisierung, Entscheidungsunterstützung
- **Maschinelles Lernen (ML):**
  - Überwachtes Lernen (z. B. Klassifikation)
  - Unüberwachtes Lernen (z. B. Anomalieerkennung)
  - Bestärkendes Lernen (adaptives Verhalten)
- **Large Language Models (LLMs):**
  - Zerlegen komplexer Aufgaben in Schritte
  - Generieren von Exploit-Vorschlägen & Payloads
  - Automatisierte Dokumentation
- **Schwächen:** Halluzinationen, begrenztes Kontextfenster, fehlende Security-Spezialisierung

Methodik – Toolauswahl

- **RamiGPT**
  - Speziell für Privilege Escalation (Linux/Windows)
  - Kombination von KI-Logik & Tools wie LinPEAS, BeRoot
- **PentestGPT (CLI)**
  - Open-Source, textbasiert, strukturierte Workflows
  - Unterstützt systematische Web-Pentest-Phasen
- **PentestGPT (Web)**
  - Kommerziell, dialogorientiert, direkte Nutzung im Browser
  - Eignet sich für schnelle Ad-hoc-Analysen

Testumgebungen

- 3 VMs (Kali, Parrot, Ubuntu)
- OWASP Juice Shop (Web-Testumgebung)
- Isoliert & reproduzierbar

Bewertungsmatrix

- 6 Kriterien:
  - Schwachstellenabdeckung
  - Exploit-Vorschläge
  - Automatisierungsgrad
  - Kontextverständnis
  - Reporting
  - Kosten-Nutzen

RamiGPT – Szenario

- **Ziel:** Privilege Escalation unter Linux (Root-Rechte erlangen)
- **Setup:** Ubuntu-VM mit absichtlich fehlerhafter SetUID-Konfiguration („rootbash“)
- **Testmodus:**
  - Full-AI (komplett automatisch)
  - Halb-automatisch (mit manueller Unterstützung)

RamiGPT – Ergebnisse

- **Full-AI-Modus:**
  - scheiterte an sudo-Passwortabfrage
  - erkannte „rootbash“ nicht eigenständig
- **Halb-automatischer Modus:**
  - mit manuellen Eingaben erfolgreich
  - Privilege Escalation durch rootbash möglich
- **Fazit:** Nur mit Benutzerhilfe nutzbar, geringe Automatisierung

PentestGPT (CLI) – Szenarien

- **Zielsystem:** OWASP Juice Shop (verwundbare Web-App)
- **Testschwerpunkte (OWASP Top 10):**
  - Broken Access Control → JWT-Manipulation, IDOR
  - Cryptographic Failures → unsichere Hashes, „Weird Crypto“-Challenge
  - Injection → SQLi, DOM-basiertes XSS
- **Interaktion:**
  - Strukturierte Workflows über CLI-Befehle (`next`, `more`, `discuss`)
  - GPT-gestützte Vorschläge, manuelle Ausführung durch Nutzer

PentestGPT (CLI) – Ergebnisse

- **Stärken:**
  - Liefern valider Payloads (z. B. `' OR '1'='1` → Login-Bypass, Admin-Zugriff)
  - Systematische Struktur didaktisch wertvoll (Ausbildung, Training)
  - Gute Unterstützung bei IDOR & XSS durch Payload-Beispiele
- **Schwächen:**
  - Keine echte Automatisierung → alles manuell auszuführen
  - Teilweise generische Antworten, Detailtiefe nur mit Nachfragen
  - Kein Reporting-Export, nur Logfiles
- **Fazit:** Hilfreiches Assistenztool, besonders für strukturierte Tests und Ausbildung

PentestGPT (Web) – Szenarien

- **Zielsystem:** OWASP Juice Shop (wie CLI-Version)
- **Testschwerpunkte:**
  - Broken Access Control (z. B. JWT-Manipulation, IDOR)
  - Cryptographic Failures („Weird Crypto“)
  - Injection (SQLi, XSS)
- **Interaktion:**
  - Dialogorientiert, ähnlich wie ChatGPT
  - Prompts in natürlicher Sprache (DE & EN)
  - Schnelle Ad-hoc-Analysen im Browser

PentestGPT (Web) – Ergebnisse

- **Stärken:**
  - Schnelle & präzise Antworten → sofort Exploit-Beispiele
  - Einfache Bedienung, keine Installation notwendig
  - Kontextsensitiv (Deutsch/Englisch kein Unterschied)
- **Schwächen:**
  - Keine Automatisierung → alles manuell auszuführen
  - Kein Reporting-Export, Ergebnisse nur im Chat
  - Volle Funktionen nur in kostenpflichtiger Version
- **Fazit:** Praktisch für schnelle Analysen & Proof-of-Concepts, weniger für strukturierte Tests

Vergleich der Tools

- **Bewertungskriterien:** Schwachstellenabdeckung, Exploit-Vorschläge, Automatisierung, Kontextverständnis, Reporting, Kosten-Nutzen
- **Gesamtpunkte (max. 12):**
  - RamiGPT: 5/12
  - PentestGPT (CLI): 8/12
  - PentestGPT (Web): 7/12
- **Stärken & Schwächen:**
  - RamiGPT: interessant für Privilege Escalation, aber unreif, wenig Automatisierung
  - CLI: methodisch klar, gute Payloads, aber langsamer & manuell
  - Web: schnell & flexibel, aber limitiert ohne Automatisierung/Reporting

Stärken & Schwächen der KI-Tools

Stärken

- Unterstützung bei Routineaufgaben (z. B. Payload-Generierung)
- Nützliche Exploit-Vorschläge und Erklärungen
- Methodische Unterstützung (CLI) bzw. schnelle Ad-hoc-Analysen (Web)
- Niedrige Einstiegshürden für Einsteiger & Ausbildung

Schwächen

- Geringe Automatisierung → keine End-to-End-Pentests
- Schwaches Kontextverständnis (z. B. RamiGPT bei Passwortabfragen)
- Ergebnisse oft nicht reproduzierbar
- Fehlende Reporting-/Exportfunktionen

Fazit

- **KI = Unterstützung, kein Ersatz** **\uf0a0** menschliche Expertise bleibt unverzichtbar
- **Nutzen:**
  - Effizienzsteigerung bei Routineaufgaben
  - Hilfreich für Ausbildung & strukturierte Analysen
  - Schnelle Proof-of-Concepts möglich
- **Grenzen:**
  - Keine vollständige Automatisierung
  - Ergebnisse nicht immer reproduzierbar
  - Eingeschränktes Kontextverständnis

Ausblick

- **Integration in DevSecOps** → KI-gestützte Tools als Teil kontinuierlicher Sicherheitsprozesse
- **Technische Weiterentwicklung** → Größere Kontextfenster & verbesserte Modelle → Retrieval-Augmented Generation (RAG) für aktuelles Wissen
- **Anwendung in der Ausbildung** → KI als interaktiver Trainingspartner → Unterstützung beim Erlernen von Angriffstechniken & Abwehrmaßnahmen
- **Langfristige Perspektive** → KI erweitert klassische Pentests → Richtung: skalierbare & adaptive Sicherheitsprüfungen

Literatur I

Bitkom (2023):
Mangel an IT-Fachkräften droht sich zu verschärfen.
https://www.bitkom.org/Presse/Presseinformation/Mangel-an-IT-Fachkraeften-droht-sich-zu-verschaerfen

Vielen Dank

xxx

Hochschule Mittweida
University of Applied Sciences
Technikumplatz 17 | 09648 Mittweida
Angewandte Computer- und Biowissenschaften