

Essay Spring 2025

# PenTestGPT in Web Application Penetration testing

Master essay, spring 2025

**Julia Vister**  
**juliavi@ifi.uio.no**

Department of Informatics  
Faculty of Mathematics and Natural Sciences

14th May 2025





**Julia Vister**

# Assessing the reliability of PenTestGPT

AI-Driven Web Application Testing

Supervisors:

Tamas György Bisztray, Rebeka Toth

## **Abstract**

This thesis explores the field of artificial intelligence in penetration testing in web applications, focusing on the effectiveness of PenTestGPT, which is an AI-powered penetration testing tool. With the rise of AI-driven cybersecurity tools, the help of large language models in web application penetration testing remains an open question. Therefore, this thesis aims to conduct a case study in which PenTestGPT and human penetration testers independently assess vulnerable web applications. The findings aim to provide information on the usefulness of integrating AI tools into security testing and its potential role in automated web application security assessments.

**Keywords:** AI in penetration testing, automated testing, PenTestGPT, Web Application penetration testing, Reconnaissance

# Contents

	Abstract . . . . .	2
1	Introduction . . . . .	7
	1.1 Motivation . . . . .	8
	1.2 Objective. . . . .	8
	1.3 Research Questions . . . . .	8
	1.4 Contribution. . . . .	8
2	Theoretical Background . . . . .	9
	2.1 Large Language Models (LLM) . . . . .	9
	2.2 Penetration Testing . . . . .	11
	2.2.1 Definition and Purpose . . . . .	11
	2.2.2 Types of Penetration Testing . . . . .	11
	2.2.3 Phases of Penetration Testing. . . . .	12
	2.2.4 Common Tools and Techniques . . . . .	13
	2.2.5 Web Application Penetration Testing . . . . .	14
	2.2.6 Legal and Ethical Considerations . . . . .	16
	2.3 AI in cybersecurity . . . . .	17
	2.3.1 AI-driven Pentesting: PtestGPT . . . . .	17
	2.4 Related Work . . . . .	18
	2.4.1 AI driven penetration testing . . . . .	18
3	Methodology . . . . .	19
	3.1 Research Design. . . . .	19
	3.2 Test Environment. . . . .	19
	3.2.1 Web Application Design . . . . .	19
	3.2.2 Vulnerability Structure. . . . .	19
	3.3 Participant Groups . . . . .	20
	3.3.1 Ethical Approval and Consent. . . . .	20
	3.4 Tools and Procedures . . . . .	20
	3.4.1 Task Execution . . . . .	20
	3.4.2 Data Collection . . . . .	20
	Bibliography . . . . .	21

## Contents

# List of Figures

2.1	Large Language Model Capabilities . . . . .	9
2.2	Architecture of a large language model. . . . .	10
2.3	An overview of pen testing stages, with six more defined phases. . . . .	13
2.4	Screenshot of a Nmap CLI command and the output, showing open port, service, and version of a website used for an ethical hacking course at UiO. This command specifies to look for open ports in a specific range (port 4500-5000). The output tells us that port 4746/tcp is open, that it is a ftp service and the version is ProFTPD. . . . .	15
2.5	Passive reconnaissance from a CTF challenge using ViewDNS.info to examine the IP history of bmw.com. This reveals previous IP addresses, their geographical locations, and hosting providers which is useful for infrastructure mapping during early pentesting stages. . . . .	15
2.6	The PentestGPT modules and how they correlate to each other . . . . .	17

## List of Figures

# **Chapter 1**

## **Introduction**

In an era where digital infrastructure supports nearly every aspect of modern society, web applications represent both a basis of technological progress and a significant vector for cyberthreats. As organizations increasingly rely on complex web-based systems to handle sensitive data and deliver critical services, securing these applications has become a top priority. Penetration testing remains one of the most effective strategies to identify and remediate vulnerabilities before they are exploited by malicious actors. Traditional penetration testing, however, is resource-intensive, and requires skilled professionals to navigate systems, analyze responses, and think creatively about how to bypass protections. Among the various stages of a penetration test, the reconnaissance phase is particularly vital. It serves as the foundation upon which subsequent exploitation and vulnerability analysis are built, as testers gather essential intelligence about the target system's structure, technologies, and potential weak points. At the same time, advances in artificial intelligence (AI) have opened new avenues for enhancing cybersecurity practices. Tools like PentestGPT - AI model trained to understand penetration testing workflows - have emerged, offering the promise of automating complex tasks that once required human intuition. This raises a question: to what extent can AI tools like PentestGPT substitute for human expertise in the reconnaissance phase of web application penetration testing?

This thesis investigates that question by creating a controlled experimental setup consisting of web applications with varying complexity and vulnerability profiles. These applications are tested by both human participants, who range from certified professionals to ethical hacking students, and PentestGPT. The goal is to compare performance across different recon tasks, identify where AI can effectively support or replace human testers, and understand the limits of current AI-driven approaches.

This chapter explores the field of study and sets the groundwork for research. It highlights the contributions and results of the thesis.

## 1.1 Motivation

## 1.2 Objective

The goal of this study is to evaluate the extent to which the reconnaissance phase of web application penetration testing can be automated or assisted using a large language model - specifically, PentestGPT - compared to human testers of varying experience levels.

## 1.3 Research Questions

- RQ1:
- RQ2:

## 1.4 Contribution

By exploring the intersection of manual expertise and intelligent automation, this research aims to contribute both practical insights for security practitioners and theoretical grounding for future studies in AI-driven penetration testing.

# Chapter 2

## Theoretical Background

This chapter will provide the theoretical background that is needed to understand this thesis. The chapter starts with an overview of large language models in Section 2.1, followed by a discussion of what penetration testing is in Section 2.2. AI in cybersecurity will be covered in Section 2.3. Lastly, related works will be explored in Section 2.4.

### 2.1 Large Language Models (LLM)

Large Language Models belong to the category of foundation models, which are neural networks that are trained on a massive set of data to supply the fundamental components necessary to enable multiple use cases and application scenarios. Because of this, LLMs are capable of understanding and generating human-like language, as well as other forms of content to handle diverse tasks. LLMs are able to understand context, produce coherent and contextually relevant content, translate languages, summarize text, answer questions, help writing, and also assist in code generation. Some well-known and easily accessible LLMs examples are Open AI's Chat GPT models or Google's BERT model. [6] Not only are LLMs helpful to a single individual with day-to-day tasks, they are also revolutionizing applications in a multitude of fields such as chat bots, research assistance, and content creation.

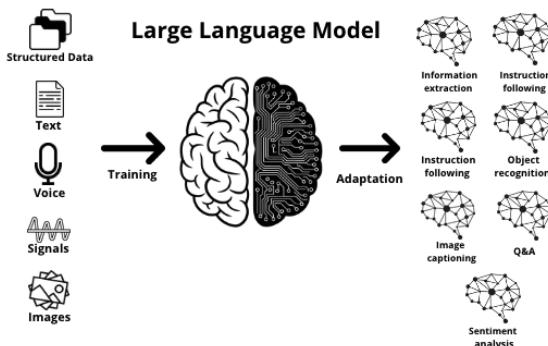


Figure 2.1: Large Language Model Capabilities

Large language models use deep learning techniques and huge amounts of textual data to function. They are built from multiple layers of neural networks each containing a parameter which can be fine-tuned. An attention mechanism further enhances these layers by making the model focus on relevant parts of the input. During the training, the model learns to predict the next word in a sentence by analyzing context and assigning probability scores to the likelihood of words - tokens - appearing. These tokens are then transformed into embeddings, numeric representation of the context. Once trained on training data, it can generate text by autonomously predicting the next word based on the input it receives. [6]

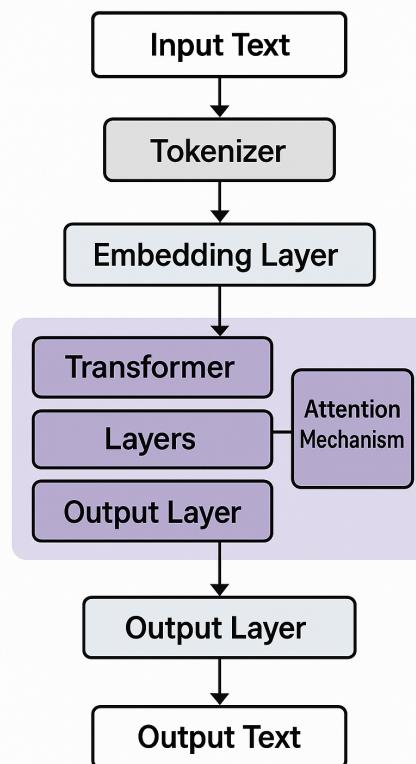


Figure 2.2: Architecture of a large language model

## 2.2 Penetration Testing

Penetration testing, often referred to as pentesting, is a technique used to identify security vulnerabilities, primarily within organizations. A penetration test simulates a cyberattack to find vulnerabilities in a computer system. By staging such an attack, a company's cybersecurity team can uncover critical security vulnerabilities and address them. This task is performed manually by experienced specialists known as penetration testers. They are professionals in ethical hacking, which is using hacking tools and techniques to fix and strengthen security weaknesses rather than causing harm. The phrase "penetration testing" describes one of the techniques utilized by ethical hackers. [7] Penetration testing should not be mistaken for vulnerability assessment, as the two serve distinct but similar purposes. While a vulnerability assessment is typically automated, performed regularly, and designed to detect common weaknesses, a penetration test goes beyond this routine scan. It simulates real-world attacks to uncover deeper, more complex security issues.

### 2.2.1 Definition and Purpose

There are three primary motivations for performing a penetration test.

In penetration testing, once vulnerabilities are identified, testers exploit them in simulated real-world attacks that mimic the behaviors of malicious hackers. This results in a more detailed and in-depth understanding of how attackers actually exploit vulnerabilities to gain unauthorized access to sensitive data or disrupt operations. With this in mind, an organization's security team can use the mock attack to design effective network security controls and defenses tailored to counter actual cyberthreats.

Pentesting combines automated tools and manual techniques to discover known and unknown vulnerabilities. In this case, pen testers actively attempt to exploit the weaknesses they find and, therefore, minimize false positives. If a pen tester can successfully exploit a vulnerability, so can cybercriminals. In addition, pen testing aids in meeting regulatory compliance by demonstrating that an organization's security controls function correctly. This ensures compliance with standards such as the General Data Protection Regulation (GDPR) and ISO standards.[2]

### 2.2.2 Types of Penetration Testing

There exist different types of pen tests that target different types of assets. The first one is the most crucial to understand for this thesis.

1. Application pen tests
2. Network pen tests
3. Hardware pen tests
4. Personnel pen tests

Application pen testing looks for vulnerabilities in applications and related systems; this also includes web applications, websites, application programming interfaces (APIs), mobile apps, and cloud apps. The most common method to use in this case is to look for known vulnerabilities listed in the OWASP Top 10, which is a list of the most critical vulnerabilities in web applications. [9] [7] In addition to this, it is common to look for less known security flaws that are unique to the app at hand. When it comes to Web applications, pen testing targets open source and custom web applications to identify and exploit vulnerabilities in authorization, security configuration, and data protection mechanisms. Some exploit techniques used for this are SQL injection, cross-site scripting (XSS), or cross-site request forgery (CSRF). [19] It examines the infrastructure, design, and configurations of a web application. [3] As mentioned in subsection 2.2.1 regarding compliance with security standards, Web application pen testing helps comply with standards and regulations such as HIPAA, GDPR, PCI-DSS and SOC-2. [19]

Network pen testing is divided into internal and external tests, and mimics attacks on a company's computer network both physically and more remotely. Hardware pen testing revolves around finding vulnerabilities in devices that are connected to the network. Examples of this are laptops, mobile devices, and operational technology (OT). Here, a pen tester may look for software flaws and physical vulnerabilities. Personnel pen testing looks for flaws in employee's security routines. It is meant to check how vulnerable a company is to social engineering attacks, often using methods such as phishing. [7]

### 2.2.3 Phases of Penetration Testing

This process is quite extensive and complex and involves multiple stages. There is no standard answer for how long the process should take, as it all depends on various variables such as the objectives, approach, and complexity of the attack surface. It includes gathering information about the target system, also called reconnaissance, identifying potential entry points, exploiting vulnerabilities, and documenting the findings in a detailed report. [16] There exists different ways to categorize penetration testing phases, however, the most common is to divide it into four larger processes or six more defined processes.[3] First is a planning phase where a scope for the test is defined. This scope outlines what systems shall be tested, when it will happen, and the methods to use. Another thing the scope defines is the amount of information that pen testers will have available in advance.[7].

1. **Reconnaissance:** The phase most crucial to this research. Gathers information on target system using various hacking techniques. If the target is an application, part of the recon may be to study the source code. Another method here is to use open source intelligence (OSINT), which is reading public information. This step is crucial for gathering critical information for the exploitation phase.[3]
2. **Target Discovery:** Pen testers use the information collected during the recon step to identify exploitable vulnerabilities. Examples of this is either using tools, such as a port scanner, or developing a fake story for a social engineering pen test.[7]
3. **Exploitation:** The actual attack occurs. Various types of attacks may be tried depending on the target, vulnerabilities, and scope, and they target each

vulnerability discovered during the reconnaissance. Once a vulnerability is exploited, pen testers can dig deeper and access more of the vulnerability. This part is called "vulnerability chaining". [3] [7]

4. **Cleanup and reporting:** Pen testers clean up any traces that are left behind, and any changes to the infrastructure will be restored to the state before testing began. A detailed report on the mock attack will also be issued. The report should include a list of all vulnerabilities and exploits that are categorized according to a risk level, as well as recommendations for improvements. [3]



Figure 2.3: An overview of pen testing stages, with six more defined phases.

#### 2.2.4 Common Tools and Techniques

The most common tools to perform a reconnaissance, discover vulnerabilities, and automate parts of the pen test process include [7]:

1. Specialized operating systems. The most popular is Kali Linux.
2. Credential-cracking tools, such as Hydra and John the Ripper.
3. Port Scanners, such as Nmap.
4. Vulnerability scanners and web vulnerability scanners, such as Burp Suite.
5. Packet analyzers/packet sniffers, such as Wireshark and tcpdump.
6. Penetration testing framework, Metasploit.

### 2.2.5 Web Application Penetration Testing

The web application penetration process builds upon general penetration testing methodologies, but is tailored to the intricacies of web-based systems. It generally follows the core phases outlined in subsection 2.2.3: planning, reconnaissance, discovery, exploitation and reporting. The planning phase remains the same with defining a scope, deciding which application pages need to be tested, and whether to perform internal, external, or both testing. This section emphasizes the reconnaissance phase, focusing on its steps, tools, and practical examples relevant to web applications.

#### Reconnaissance Phase

The reconnaissance phase is critical to understand the structure, components and weak points of a web application. Reconnaissance can be divided into passive and active reconnaissance. *Active reconnaissance* is when a pen tester directly interacts with the target to get information. While not commonly employed in legal penetration testing engagements, techniques such as watering hole attacks - where a malicious actor compromises websites frequented by the target through loopholes - represent advanced social engineering methods that fall under broader reconnaissance strategies. However, there are many other techniques used for active reconnaissance such as port scanning, network mapping, DNS enumeration and vulnerability scanning. *Passive reconnaissance* is the opposite of active, meaning pen testers collect information without interacting directly with the target. A pen tester collects public information about the target in this part, without alerting the target. Examples of passive recon are monitoring social media or public databases. Some techniques used for passive reconnaissance is search engine dorking, public data aggregation, social media analysis and email harvesting. [18] Common techniques and tools used to gather information about the target:

- **Nmap (port scanning):** A network scanner used to discover open ports, services running on those ports, and the operating system (OS) of the target system.

*Case:* A company focusing on tokenization services wanted to test two newly created web applications to find out if there are any security gaps. Nmap was used to scan the web application's infrastructure. This revealed open ports and services, including outdated versions of SSH and HTTP services. [15]

- **DNS enumeration:** Techniques and tools that uncover subdomains, domain structure, and misconfigured records.

*Case:* A multinational retail chain requested a security assessment on several web applications. Testers performed DNS enumeration to identify subdomains and uncovered legacy subdomains which pointed to deprecated services. [14]

- **Gobuster / Dirb:** Tools used to brute-forcing directories and file paths to discover hidden resources within web applications.

*Case:* In the same security assessment for the tokenization services provider, Dirb was used to brute-force directories on their web server. This led to the discovery of hidden administrative directories. [15]

- **SSL/TLS analysis:** Identifies weaknesses or misconfigurations in HTTPS setups, such as outdated protocols or missing certificate chains.

*Case:* An asset management company wanted a penetration test for their web application as they deal with a lot of sensitive data. It was revealed through

SSL/TSL analysis that the application was using deprecated TLS 1.1 protocols. [12]

- **WHOIS lookup:** Reveals domain registration and hosting information including contact details and infrastructure ownership. For example, a WHOIS lookup on a misconfigured server can reveal outdated administrative contacts and expired certificates - offering insights into organizational oversight and possible attack vectors.

*Case:* In a security evaluation for a US contract services company, WHOIS lookups - as part of open-source intelligence, were performed to gather information about domain registrations. This helped reveal contact information associated with the domain. [13]

- **Search Engine Dorking:** Utilizes advanced search operators (e.g., 'site:', 'intitle:', 'filetype:') to uncover exposed information indexed by search engines. Examples include login pages, configuration files, and backups.

```
(root㉿kali)-[~]
$ sudo nmap -sT -p4500-5000 kenobi.hackingarena.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-12 13:47 CEST
Nmap scan report for kenobi.hackingarena.com (158.39.48.133)
Host is up (0.043s latency).
Not shown: 500 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
4746/tcp  open  ftp      ProFTPD

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.35 seconds
```

Figure 2.4: Screenshot of a Nmap CLI command and the output, showing open port, service, and version of a website used for an ethical hacking course at UiO. This command specifies to look for open ports in a specific range (port 4500-5000). The output tells us that port 4746/tcp is open, that it is a ftp service and the version is ProFTPD.

IP Address	Location	IP Address Owner	Last seen on this IP
160.46.226.165	Germany	Bayerische Motoren Werke Aktiengesellschaft	2024-09-10
160.46.244.131	Munich - Germany	Bayerische Motoren Werke Aktiengesellschaft	2017-05-02

Figure 2.5: Passive reconnaissance from a CTF challenge using ViewDNS.info to examine the IP history of bmw.com. This reveals previous IP addresses, their geographical locations, and hosting providers which is useful for infrastructure mapping during early pentesting stages.

## Vulnerability Scanning

Automated tools are used to scan web applications for known vulnerabilities. These tools cross-reference the application's code, configuration, and dependencies against Common

Vulnerabilities and Exposures (CVEs) databases to identify potential weaknesses. This helps identify security misconfigurations and outdated components. This part is called vulnerability scanning. [19] Common vulnerability scanners are: [10]

- **Nikto:** An open-source scanners that can identify vulnerabilities specific to web applications, such as Cross-Site Scripting (XSS) and SQL injection.
- **Burp Suite:** security testing platform and proxy tool that allows pen testers to intercept and modify web traffic. Facilitates brute-forcing and fuzzing attacks. [8]

### Exploitation and Post-exploitation

Once the vulnerability scan is completed, manual exploitation is carried out to identify more complex or hidden vulnerabilities. This involves using specialized tools and techniques to gain unauthorized access to the system. Once a vulnerability is successfully exploited, pen testers can attempt to chain exploits together to achieve greater impact. Specific tools are often used to help exploit particular types of vulnerabilities, such as: [19]

- **SQLmap:** An automatic SQL injection and database takeover tool which can identify and exploit SQL injection vulnerabilities.[8]
- **Metasploit:** A widely used exploit framework with a library of exploits and payloads for known vulnerabilities which allows controlled compromise of systems.
- **Hydra:** A tool for performing brute-force attacks on login forms and protocols like HTTP, FTP and SSH.

### 2.2.6 Legal and Ethical Considerations

It is important to comply with and understand the legal requirements and restrictions that monitor penetration testing. The rules on pen testing vary country by country and by region. One of the most important legal considerations is the compliance with data protection laws. It is important that personal and sensitive data is adequately protected during the testing process. This means getting consent, anonymizing data and adhering to regulations such as the General Data Protection Regulation (GDPR) if in the European Union. Ensuring that the intellectual property rights of others are not violated is also essential before conducting a penetration test. Ethical guidelines have been established for penetration testers to outline the expected behavior and ethical responsibilities of ethical hackers, ensuring proper authorization and respect of confidentiality. [11] However, it is important to note that the scope of this research does not involve third-party systems or the handling of personal data. All testing is performed on self-constructed environments designed solely for experimental purposes. Therefore, while legal and ethical compliance is acknowledged as a foundational aspect of professional practice, it does not directly impact the methodology or execution of this study.

## 2.3 AI in cybersecurity

### 2.3.1 AI-driven Pentesting: PtestGPT

PentestGPT is an LLM-powered automated penetration testing framework that is open source on GitHub and created by a university student who names himself GreyDGL.

*"It is built on top of ChatGPT API and operate in an interactive mode to guide penetration testers in both overall progress and specific operations"* - GreyDGL, creator of PentestGPT [5]

PentestGPT is designed with three modules: parsing, reasoning, and generation. [4] The parsing module analyzes the output of the penetration testing tools and the content displayed on the Web interface and, therefore, acts as a support interface.[1] This module handles four types of information. The first (1) is user intentions - instructions provided by the user to guide subsequent actions. The second (2) consist of raw outputs generated by the security testing tools. The third (3) is raw HTTP web data derived from HTTP web interfaces, and the fourth (4) includes source codes extracted during the penetration testing process. [4] The test reasoning module interprets the test results to guide penetration testers on the next appropriate actions. [1] It receives the test results and prepares a testing strategy for the next steps. This strategy is then passed on to the generation module. The test generation module produces precise penetration testing commands and procedures that users can execute.[1] Serving as a bridge between the reasoning module and the subsequent actions. The generation module translates sub-tasks from the reasoning module into concrete commands and actionable steps. Additionally, it generates human-readable outputs to clearly convey the testing process. [4]

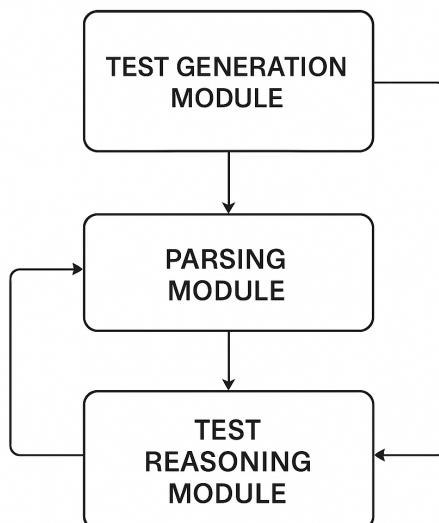


Figure 2.6: The PentestGPT modules and how they correlate to each other

PentestGPT has demonstrated capabilities that can assist or partially replace human efforts in several key tasks within the reconnaissance phase of web application penetration testing. Specifically, it can automate information gathering activities that follow structured patterns or rely on known techniques. For example, PentestGPT can analyze DNS records, suggest appropriate subdomain enumeration commands, interpret Nmap or Dirb outputs, and identify misconfigurations from WHOIS data or HTTPS headers. By synthesizing large volumes of output from tools like Nmap or Gobuster, and cross-referencing findings with known vulnerability patterns, it acts as an intelligent assistant that accelerates the initial mapping of the target environment. [20]

PentestGPT particularly excels in passive and semi-active reconnaissance tasks that involve interpreting structured data, forming hypotheses, and providing next-step suggestions. For instance, it can recommend relevant search engine dorks, parse results, and guide users in identifying public exposure without direct interaction with the target. [20] Its natural language capabilities allow it to translate technical findings into actionable intelligence, which is especially valuable for less experienced testers. However, PentestGPT's usefulness may be limited in dynamic or ambiguous situations where creativity, intuition, or contextual judgment is required. [4]

## 2.4 Related Work

The reliance on artificial intelligence has increased greatly in cybersecurity, leading to advances in automated penetration testing. Traditionally, penetration testing is a time-consuming and costly process that involves human security experts who identify and exploit vulnerabilities manually. AI-driven approaches that use large language models such as PenTestGPT have emerged as alternatives to human-led security testing. These models claim to automate reconnaissance, vulnerability detection, and exploitation. However, this raises questions about their effectiveness, reliability, and limitations compared to human penetration testers. This section will look at existing research on AI-driven penetration testing, identify knowledge gaps, and establish a foundation for further research of LLM in security assessments.

### 2.4.1 AI driven penetration testing

Artificial intelligence (AI) and its technologies have the ability to increase the efficiency and effectiveness of identifying vulnerabilities and performing an assessment. Businesses adopting AI technologies for security testing using machine learning and known automated tools can simplify the assessment process, reduce human error, and provide more accurate information on their system's vulnerabilities.

Machine learning algorithms are the core of automated penetration testing. This is because these algorithms are able to combine and analyze large datasets to identify patterns and also predict potential vulnerabilities based on data given to the algorithm. This paves the way for AI tools to improve their detection algorithm over time, leading it to become effective in discerning attack vectors and new threats. Another critical piece in these AI tools is natural language processing, as this is what allows computers to read and analyze human language. [17]

# **Chapter 3**

## **Methodology**

The goal of this chapter is to familiarize ourselves with the research methods used to answer the research questions presented in the introduction.

### **3.1 Research Design**

This is a comparative experimental study that uses a quasi-controlled environment. Two groups - PentestGPT and human testers - will perform reconnaissance on prebuilt web applications with known vulnerabilities. Their findings will be compared quantitatively and qualitatively.

### **3.2 Test Environment**

#### **3.2.1 Web Application Design**

Ten identical applications in layout and tech stack. Each version increases in complexity and obfuscation of vulnerabilities. There is varying security controls and detection difficulty. All applications will be internally documented with a ground truth list of vulnerabilities.

#### **3.2.2 Vulnerability Structure**

- Web application 1: Trivial vulnerabilities
- Web application 5: Intermediate complexity
- Web application 10: High complexity

Group	Description
A	Certified professional penetration testers
B	Hobbyists/CTF participants
C	Ethical hacking students at UiO

### 3.3 Participant Groups

Each participant will be assigned the same applications and allowed the same tools and time limits as PentestGPT.

#### 3.3.1 Ethical Approval and Consent

Participants will be recruited voluntarily, anonymized and approved under university research ethics.

### 3.4 Tools and Procedures

#### 3.4.1 Task Execution

All participants, including PentestGPT, are given a standardized instruction sheet. Fixed time budget per application, e.g., 30-60 minutes per application.

- Example task: "Perform reconnaissance on this web application and enumerate as much information as possible that could aid later exploitation".

#### 3.4.2 Data Collection

- Human Participants: Record command history, notes, outputs, tools used, results.
- PentestGPT: Prompt logs and output.

Screen recording or activity logs to capture user sessions for review.

# Bibliography

- [1] Balaji. Pentestgpt - a chatgpt powered automated penetration testing tool, 2024. Accessed: 05.05.2025.
- [2] BlackDuck. What is penetration testing and how does it work? | black duck.
- [3] CyberCX. The complete guide to penetration testing.
- [4] Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. Pentestgpt: An llm-empowered automatic penetration testing tool. *arXiv preprint arXiv:2308.06782*, 2023.
- [5] GreyDGL. Pentestgpt. Accessed: 05.05.2025.
- [6] IBM. What are large language models (LLMs)? | IBM.
- [7] IBM Security. What is penetration testing? | IBM.
- [8] MSBJ. 10 tools for web application penetration testing, 2023. Accessed: 03.05.2025.
- [9] Inc OWASP Foundation. Owasp foundation.
- [10] Inc OWASP Foundation. Vulnerability scanning tools. Accessed: 03.05.2025.
- [11] PenTestWizard. Achieving penetration testing compliance: Legal and ethical insights. Section: General.
- [12] ScienceSoft. Web application and netowrk penetesting for an asset managment company. Accessed: 06.05.2025.
- [13] ScienceSoft. Web application and network penetration testing for a us contract services company. Accessed: 06.05.2025.
- [14] ScienceSoft. Web application penetration test for a multinational retail chain. Accessed: 06.05.2025.
- [15] ScienceSoft. Web application penetration test for a tokenization services provider in 7 days. Accessed: 06.05.2025.
- [16] R. S. I. Security. What is the penetration testing execution standard?
- [17] Mariam Soltanifar. AI-driven penetration testing: Automating vulnerability assessments. 2(1). Number: 1.
- [18] Muhammad Usman. Active and passive reconnaissance techniques. Accessed: 05.05.2025.

## Bibliography

- [19] Jinson Varghese. What is web app penetration testing? [how to conduct it], 2025. Accessed: 03.05.2025.
- [20] YESCHATAI. Pentestgpt. Accessed: 07.05.2025.