

Security Assessment

pinewheel.ai

Mahadir Bin Muhammed | 28/02/2025

# Confidentiality Statement

This document contains **sensitive security information** and is intended **solely for authorized individuals** conducting security assessments within an approved and ethical framework. Any unauthorized access, distribution, or use of this document is strictly prohibited. All findings, methodologies, and recommendations must be handled with the highest level of discretion to prevent misuse.

All parties involved must adhere to **ethical hacking principles**, applicable **cybersecurity laws**, and **organizational security policies** to ensure compliance and responsible disclosure.

# Disclaimer

This assessment is conducted for **educational and security research purposes only**. The techniques and tools used in this engagement are intended for **legitimate security testing** and must **not** be used for unauthorized activities. Unauthorized access, exploitation, or data manipulation of any system **without explicit permission** is illegal and may result in **legal consequences**.

The author(s) and security personnel involved in this assessment **assume no liability** for any unintended consequences arising from the use or misuse of this information. Organizations and individuals are responsible for ensuring compliance with **applicable laws and regulations** before executing any security testing procedures.

If any vulnerabilities are identified, they should be **reported responsibly** to the relevant stakeholders and **not disclosed publicly** without prior approval.

# Overview

This security assessment focuses on conducting an in-depth penetration test of the **Pinewheel AI platform** (<https://testing.pinewheel.ai>) to identify vulnerabilities and security weaknesses. The assessment involves both **authenticated and unauthenticated testing** using a combination of **manual techniques and automated tools available in Kali Linux**.

The primary objective is to **identify, exploit, and document** potential vulnerabilities such as **SQL Injection (SQLi), Cross-Site Scripting (XSS), Server-Side Request Forgery (SSRF), Server-Side Template Injection (SSTI), Remote Code Execution (RCE), and Prompt Injection**. The audit will include **enumerating services and endpoints**, assessing **authentication mechanisms**, and **evaluating access control measures**.

The approach follows a structured methodology:

1. **Reconnaissance & Information Gathering** – Identifying exposed services, subdomains, and open ports.
2. **Enumeration & Attack Surface Mapping** – Discovering endpoints, authentication mechanisms, and potential misconfigurations.
3. **Vulnerability Testing** – Actively probing for security flaws using tools like **Burp Suite, Nmap, SQLmap**.
4. **Exploitation & Post-Exploitation** – Validating vulnerabilities and assessing their impact.
5. **Post-Authentication Testing** – Testing restricted areas after logging in.
6. **Reporting & Documentation** – Delivering a comprehensive penetration test report detailing **findings, proof-of-concepts (PoCs), severity ratings, and remediation recommendations**.

The final deliverables include:

* A **detailed pentest report** summarizing the identified vulnerabilities.
* A **command log file** documenting all tools and techniques used, along with their results and justifications.

This security audit will help strengthen the platform’s resilience against cyber threats and improve its overall security posture.

Tools and Results

**Tool:**

1. **Burp Suite – Automated Testing**

Web Application Security Testing:

***Audit Summary:***

Total audit items: **32**

Requests: **20340**

Network Error: **6**

***Vulnerabilities:***

Total Informational: 16

Total Low: 7

**Reason:**

* **Intercept & Modify Requests** – Captures and alters HTTP/S traffic between browser and server.
* **Automated Scanning** – Identifies vulnerabilities like SQLi, XSS, SSRF**, etc.**
* **Manual Testing Tools** – Includes Repeater, Intruder, Sequencer**,** and **Decoder** for in-depth testing.
* **API Security Testing** – Imports OpenAPI definitions and tests endpoints.
* **Session & Authentication Testing** – Analyzes cookies, tokens, and login **mechanisms** for weaknesses.
* **Bypass Security Controls** – Helps in WAF bypassing, privilege escalation, and fuzzing.
* **Active & Passive Crawling** – Maps web applications and discovers hidden endpoints.
* **Extension Support** – Uses Burp Extensions & BApp Store tools for enhanced testing

1. **nslookup:**

(Name Server Lookup) is a command-line tool used for querying **DNS** records. It helps in gathering subdomains, IP addresses, mail servers, and other DNS information about a target system.

**Command: nslookup** [**https://testing.pinewheel,ai**](https://testing.pinewheel,ai)

**Output: Server: 192.168.247.2**

**Address: 192.168.247.2#53**

1. **nmap:**

**Command: nmap -A -T4 192.168.247.2**

**Output: Port: 53/tcp**

**STATE: Open**

**Service: Domain**

**Version: dnsmasq 2.87**

**MAC Address: 00:50:56:FA:CA:FC**

**Aggressive OS guesses: VMware Player virtual NAT device (99%), Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (93%), DD-WRT v24-sp2 (Linux 2.4.37) (91%), Microsoft Windows XP SP3 (91%), Actiontec MI424WR-GEN3I WAP (90%), DVTel DVT-9540DW network camera (89%), Linux 3.2 (89%), Linux 4.4 (89%)**

**No exact OS matches for host (test conditions non-ideal).**

**Network Distance: 1 hop**

**TRACEROUTE**

**HOP RTT ADDRESS**

**1 1.15 ms 192.168.247.2**

**Reason:**

* Initial Reconnaissance – Quickly gather system details.
* Enumerating Open Ports & Services – Find vulnerable services.
* Testing Firewalls & IDS Detection – See if aggressive scanning gets blocked.
* Finding Exploit Paths – OS & service versions help map attack vectors.

1. **sqlmap**

***Command:***

* 1. **Identify Vulnerable Parameters Automatically:**

**sqlmap -u "https://testing.pinewheel.ai/login?username=admin&password=pass" --dbs --batch --level=5 --risk=3 --tamper=between,randomcase --random-agent --threads=10 --flush-session --dump-all**

* 1. **Dump Full Database & Bypass WAF:**

**sqlmap -u "https://testing.pinewheel.ai/products?id=5" --dump --tamper=space2comment,charencode --level=5 --risk=3**

* 1. **Test for OS Command Execution via SQLi:**

**sqlmap -u "https://testing.pinewheel.ai/admin?id=1" --os-shell --batch**