AI-Powered Vulnerability Prioritization Dashboard

Automated Web Vulnerability Detection & Risk-Based Prioritization

---

Overview

A centralized platform that:

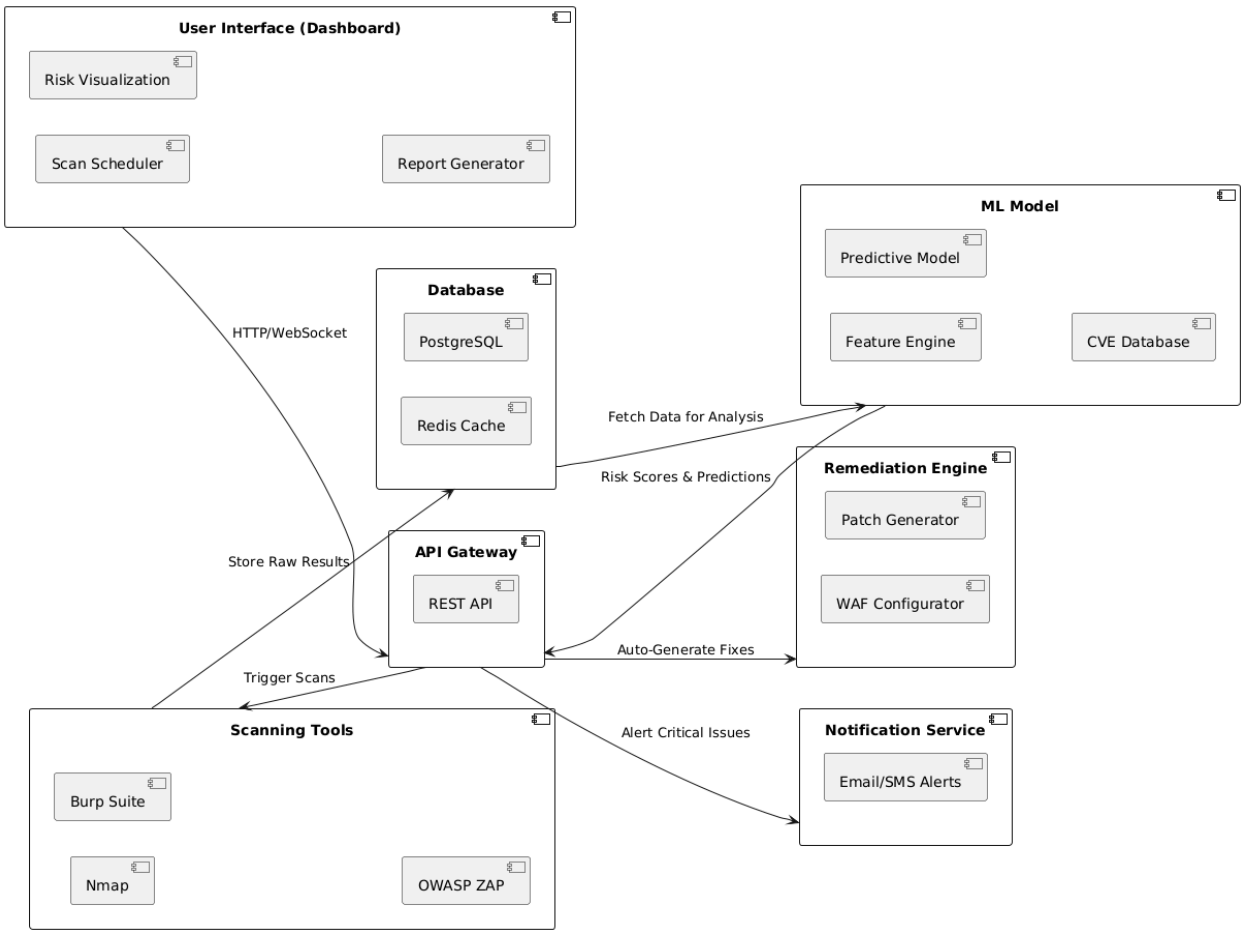
1. Integrates with **scanning tools** (Nmap, Burp Suite) to detect vulnerabilities
2. Uses **ML** to prioritize risks based on exploit likelihood and business impact
3. Provides **actionable remediation steps** via an interactive dashboard

---

Key Features

- **Automated Scanning**: Schedule scans for networks/web apps
- **Threat Intelligence**: Cross-references CVEs and exploit databases
- **Risk Scoring**: Custom algorithm combining CVSS, asset value, and attack trends
- **Remediation Automation**: One-click fixes for common vulnerabilities (e.g., WAF rule generation)

Architecture:



Workflow

1. **Input**: User defines target (IP/URL) and scan type (network/web)

2. **Scan Execution**:

    - Utilize **Nmap** for open ports/services
    - Employ **Burp Suite/ZAP** for web app vulnerabilities (XSS, SQLi)

3. **Data Enrichment**:

    - Match findings to **CVE database**
    - Fetch **exploit PoCs** from Exploit-DB

4. **Risk Scoring**:

python

```
# Simplified scoring formula
def calculate_risk(cvss, asset_value, exploitability):
    return (cvss * 0.6) + (asset_value * 0.25) + (exploitability * 0.15)
```

5. **Dashboard Display**: Color-coded vulnerabilities (critical/high/medium/low)

Tech Stack

The following table outlines the key layers and their corresponding technologies:

Layer	Technologies
Frontend	React.js + D3.js
Backend	Python (FastAPI)
ML	Scikit-learn + TensorFlow (LSTM for attack prediction)
Database	PostgreSQL (structured data) + Neo4j (attack graph visualization)
DevOps	Docker, Kubernetes, GitHub Actions

Future Enhancements

1. **Integration with Cloud Providers**: Auto-scan AWS/Azure environments
2. **Attack Simulation**: Purple teaming scenarios using Caldera
3. **CTF Mode**: Hands-on vulnerability exploitation practice

---

Example Use Case

Scenario: E-commerce website scan

1. **Findings**:
    - SQL Injection (CVSS 9.8)
    - Outdated jQuery (CVSS 7.2)
2. **Prioritization**: SQLi flagged as critical due to active exploit kits

3. **Remediation**: Auto-generate .htaccess rules to block malicious patterns