PentesterFlow: Project Overview

🎯 General Goals & Vision

Q: What is PentesterFlow?

_ PentesterFlow is a professional-grade **Agentic Dynamic Application Security Testing (DAST)** platform. It is designed to act as an autonomous security researcher that doesn't just scan for bugs but "understands" the context of the applications it tests.

_ هو منصة احترافية لاختبار أمن التطبيقات الديناميكي (DAST) تعتمد على الوكلاء الذكيين (Agentic). تم تصميمه ليعمل كباحث أمني مستقل لا يكتفي بالبحث عن الأخطاء البرمجية فحسب، بل "يفهم" سياق التطبيقات التي يختبرها.

Q: Why was PentesterFlow created?

_ Traditional security scanners often produce a lot of "noise" (false positives) and lack the context to understand complex application flows. PentesterFlow was created to bridge this gap by using **AI Agents** that can reason about vulnerabilities, validate findings, and provide actionable security intelligence.

_ غالباً ما تنتج أدوات الفحص الأمني التقليدية الكثير من "الضجيج" (النتائج الإيجابية الكاذبة) وتفتقر إلى السياق اللازم لفهم تدفقات التطبيقات المعقدة. تم إنشاء PentesterFlow لسد هذه الفجوة باستخدام وكلاء ذكاء اصطناعي (AI Agents) يمكنهم التفكير في الثغرات الأمنية، والتحقق من النتائج، وتوفير معلومات أمنية قابلة للتنفيذ.

🚀 Core Features

Q: What makes PentesterFlow "Agentic"?

_ Instead of a linear script, the platform uses specialized AI agents:

- **Recon Agent**: Maps out the target and finds hidden entry points.
- **وكيل الاستطلاع (Recon Agent):** يقوم برسم خريطة للهدف وإيجاد نقاط الدخول المخفية.
- **Attack Agent**: Crafts specific payloads based on the technologies it discovers
- **وكيل الهجوم (Attack Agent):** يصيغ حمولات (Payloads) محددة بناءً على التقنيات التي يكتشفها.
- **Validation Agent**: Uses LLMs (Google Gemini) to double-check if a vulnerability is real or a false alarm.
- **وكيل التحقق (Validation Agent):** يستخدم نماذج لغوية كبيرة (Google Gemini) للتحقق المزدوج مما إذا كانت الثغرة حقيقية أم مجرد إنذار كاذب.

Q: How does it handle asset discovery?

_ It integrates deeply with **Nmap** for network scanning, OS detection, and service fingerprinting. It then visualizes this data in an interactive **Network Topology Graph**, making it easy to see the "attack surface."

_ يتكامل النظام بشكل عميق مع أداة Nmap لمسح الشبكة، وتحديد أنظمة التشغيل، وبصمات الخدمات. ثم يقوم بتصور هذه البيانات في "رسم بياني تفاعلي لطوبولوجيا الشبكة"، مما يسهل رؤية "سطح الهجوم".

Q: What is the vulnerability engine?

_ It leverages **Nuclei**, a powerful template-based scanner, to look for thousands of known vulnerabilities, misconfigurations, and CVEs across web services, protocols, and network layers.

_ يعتمد على Nuclei، وهو ماسح ضوئي قوي يعتمد على القوالب، للبحث عن آلاف الثغرات المعروفة، والأخطاء في الإعدادات، والـ CVEs عبر خدمات الويب، والبروتوكولات، وطبقات الشبكة.

🛠️ Technology Stack

Q: What technologies are used in the Backend?

- **Language**: Python 3.10+
- **Framework**: **FastAPI** (for high-performance asynchronous API endpoints).
- **Database**: **PostgreSQL** with **SQLAlchemy** ORM.
- **Task Queue**: **Celery** with **Redis** (for handling long-running scans in the background).
- **Security Tools**: **Nmap** (discovery), **Nuclei** (scanning), **Playwright** (browser automation).

Q: What technologies are used in the Frontend?

- **Framework**: **React** with **Vite** (for a fast, modern build experience).
- **Styling**: **Tailwind CSS** (for a clean, professional UI).
- **Visualization**: **React Force Graph** and **D3.js** (for the network topology view).
- **Icons**: **Lucide React**.

Q: How is AI integrated?

_ The platform uses **Google Gemini 1.5 Flash** via the google-generativeai SDK. Gemini acts as the “brain,” reasoning through scan results and filtering out false positives.

_ تستخدم المنصة Google Gemini 1.5 Flash عبر حزمة SDK الخاصة بـ google-generativeai. يعمل Gemini بمثابة "العقل"، حيث يقوم بتحليل نتائج المسح وتصفية النتائج الإيجابية الكاذبة.

📂 Project Structure

Q: How is the repository organized?

- **/backend:** Contains the FastAPI server, AI agent logic, scan orchestrators, and database models
- **/frontend:** Contains the React source code, components, and dashboard UI.
- **/lab_config:** Contains scripts and configurations for the virtual simulation lab.

- **docker-compose.yml:** The main configuration to launch the entire platform.
- **docker-compose.lab.yml:** Configuration for the 5-node virtual corporate network.

⚙️ Installation & Setup

Q: How do I get PentesterFlow running?

1. **Clone the Repo**: bash

   git clone https://github.com/omarkapil/the-dashboard-project-.git

   cd the-dashboard-project-

2. **Configure AI**: Create a .env file in the root and add your Gemini API key:

   env GEMINI_API_KEY=your_actual_key_here

3. **Launch with Docker**: bash

   docker-compose up -d --build

4. **Access the UI**: Open your browser to

   http://localhost:5173.

Q: How do I test it without a real target?

_ You can launch the **Simulated Corporate Network** (Virtual Lab):

docker-compose -f docker-compose.lab.yml up -d

This creates 5 virtual nodes (Router, Windows PC, Linux Servers) with intentional vulnerabilities for you to practice scanning.

🔮 The Future

Q: What's next for PentesterFlow?

- **Vision Integration**: Using AI to "see" screenshots of web pages to find UI-level security issues.
- **دمج الرؤية (Vision Integration):** استخدام الذكاء الاصطناعي "لرؤية" لقطات شاشة لصفحات الويب للعثور على المشكلات الأمنية على مستوى واجهة المستخدم.
- **Enterprise Features**: Role-Based Access Control (RBAC) and integration with Jira/GitHub.
- **ميزات المؤسسات:** التحكم في الوصول القائم على الأدوار (RBAC) والتكامل مع Jira/GitHub.
- **Cloud Scanning**: Specialized agents for AWS/Azure security posture management.
- **المسح السحابي (Cloud Scanning):** وكلاء متخصصون لإدارة الوضع الأمني في AWS/Azure.