Orange Sage - Technical Documentation

1. Introduction

This document provides a comprehensive overview of the Orange Sage system, including its architecture, technologies, APIs, deployment, and development team roles. It is intended for evaluators, technical leads, and developers involved in the project.

2. Project Overview

Orange Sage is an AI-powered cybersecurity assessment platform that autonomously performs security testing using specialized AI agents. It is designed to provide businesses and security professionals with a modern web interface for managing security assessments, monitoring vulnerabilities, and generating detailed reports. By integrating multiple AI models and automating complex tasks, Orange Sage streamlines the traditionally manual and time-intensive process of cybersecurity analysis, delivering accurate, automated, and adaptive insights in real time.

3. Objectives

The primary objectives of Orange Sage are to automate vulnerability assessments through AI agents, enable secure and real-time orchestration of penetration testing tasks, and provide businesses with clear, actionable insights through detailed reports and dashboards. Additionally, the system is built with scalability and modularity in mind, ensuring that it can be integrated into enterprise-level environments with minimal friction.

4. Key Features

Core Functionalities

Orange Sage includes secure user authentication with role-based access control, as well as project and target management capabilities that allow users to create, organize, and track security assessments efficiently. It employs AI-driven scan orchestration, where multiple agents collaborate to perform black-box and white-box testing. The platform also features real-time monitoring of scanning progress, centralized findings management for tracking vulnerabilities, and automated reporting in multiple formats, including PDF, DOCX, and HTML. Settings management enables users to configure API keys, integrations, and system preferences as needed.

AI Agent Capabilities

The AI agents in Orange Sage perform autonomous testing, executing self-directed scans without human supervision. These agents dynamically adapt their strategies based on target responses to ensure comprehensive coverage across different security dimensions. They are capable of detecting OWASP Top 10 vulnerabilities and beyond. Each agent specializes in a specific phase of testing, working collaboratively to deliver accurate and holistic security results.

5. System Architecture

Orange Sage follows a modular microservice architecture with distinct separation of concerns across the frontend, backend, and service layers.

The frontend, built with React, Tailwind CSS, and shadcn/ui, provides the dashboard interface for users, handling client-side routing, state management, and API communication. The backend, developed using FastAPI, Celery, and PostgreSQL, serves as the REST API layer responsible for managing user authentication, database operations, and orchestration of AI agents. Redis and Celery manage asynchronous task execution and caching to improve scalability and performance. For object storage, the system uses MinIO (S3-compatible) to securely store generated reports and scan artifacts.

The AI layer, powered by LiteLLM, connects to OpenAI and Google Gemini APIs to perform intelligence-based analysis and natural language processing.

Data Flow:
User → Frontend UI → Backend (API request) → Celery Task → AI Agent Processing → Database/MinIO → Response sent back to UI.

6. Tech Stack

Layer	Technologies
Frontend	React, shadcn/ui, Tailwind CSS, React Query
Backend	FastAPI, SQLAlchemy, Pydantic, Uvicorn
Database	PostgreSQL
Task Queue	Celery + Redis
Storage	MinIO (S3-compatible)
AI / LLM Integration	LiteLLM (OpenAI & Gemini APIs)
Containerization	Docker + Docker Compose
Version Control	Git + GitHub
Deployment	Google Cloud with GitHub Actions (CI/CD)
Secrets Management	GitHub Secrets

7. Workflow

The user begins by registering or logging in via the frontend interface. Once authenticated, they can create a new security project and define the targets, such as URLs, repositories, or uploaded files. The backend then schedules tasks through Celery, which invokes AI-driven agents to analyze the target autonomously.

These agents utilize integrated large language models (LLMs) for reasoning and pattern recognition during the testing process. The analysis results are stored in PostgreSQL, while artifacts such as reports or evidence are saved in MinIO storage. The frontend continuously fetches updates using React Query to display real-time progress. Upon completion of the scan, users can generate, view, and download their reports in the desired format.

8. Deployment

Orange Sage is deployed on Google Cloud and utilizes GitHub Actions for continuous integration and continuous deployment (CI/CD). Environment variables and secrets are securely managed through GitHub Secrets, ensuring a protected deployment pipeline. The system is containerized using Docker Compose, which orchestrates all core services such as PostgreSQL, Redis, and MinIO to maintain consistency across development and production environments.

9. Security & Privacy

Security and privacy are central to Orange Sage's architecture. The platform uses JWT Authentication to manage secure user access and Role-Based Access Control (RBAC) to differentiate permissions for administrators, developers, and auditors. All user input is validated through Pydantic schemas, while SQLAlchemy ORM mitigates risks of SQL injection. CORS policies are properly configured to ensure safe communication between the frontend and backend, maintaining both data integrity and user privacy.

10. Testing & Validation

Comprehensive testing has been performed to ensure the reliability of Orange Sage. Manual testing was conducted for all core modules, including authentication, project creation, and reporting functionalities. Automated unit tests were developed using Pytest to validate backend routes and logic. Additionally, API testing was carried out using Swagger and Postman collections to confirm endpoint accuracy and performance under various conditions.

11. Team members & Roles

- Sufi Hassan Asim – Dashboard Developer
  - Built the analytics and reporting dashboard.
- Ali Shan – Frontend developer
  - Implemented UI components and React integration.
- Nasir Ali Khan – Backend Developer
  - Designed FastAPI routes, models, and Celery tasks.
- Haniyya Hussain – UI/UX Designer and Documentation

    - Designed the user interface and authored all documentation.
- Anas Zia – Developer
    - Assisted in frontend integration and testing.
- Abdullah Arif – Developer
    - Contributed to backend logic and Docker setup.

12. Conclusion

Orange Sage successfully demonstrates how AI-driven automation can transform cybersecurity assessment workflows. Its modular architecture, multi-agent orchestration, and scalable design provide a modern and efficient way to evaluate digital assets for vulnerabilities. Through its combination of automation, AI intelligence, and user-centered design, Orange Sage delivers a powerful yet accessible solution for businesses aiming to strengthen their cybersecurity posture.

13. References

- https://fastapi.tiangolo.com/

- https://github.com/BerriAI/litellm

- https://react.dev/

- https://www.postgresql.org/

- https://docs.docker.com/compose/