AssetMapper Architecture Overview

Service Design and Engineering
Simone Avancini
2024/2025

1. Introduction

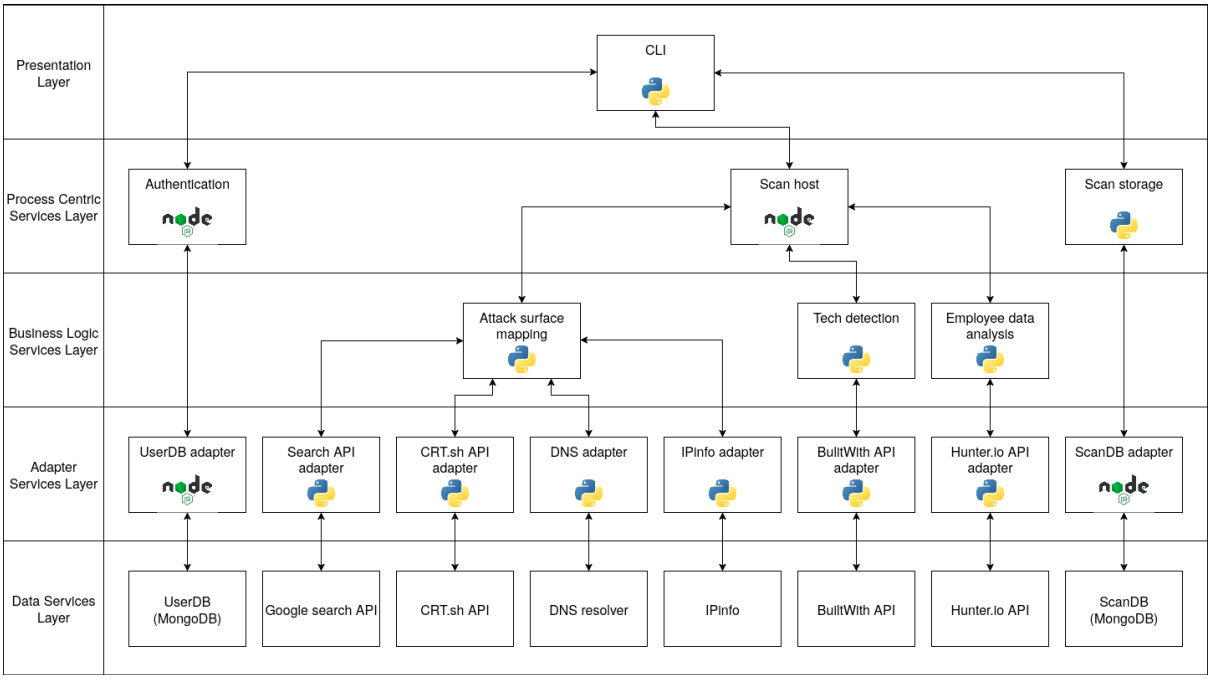
This project implements a service-oriented application designed to gather, correlate, and analyze publicly available information about a domain or organization. The system is primarily aimed at penetration testers or security researchers for the reconnaissance phase of a security assessment.

The application integrates multiple external services in a modular architecture. Each service encapsulates a specific responsibility such as authentication, attack surface mapping, technology detection, or employee data analysis. This modularity supports extensibility, scalability, and maintainability.

The primary user interface is a Command Line Interface (CLI), which orchestrates the various services to perform a scan and deliver a report to the user.

2. Architectural Layers

The diagram below provides a high-level overview of the architecture and the relationships between the services in each layer.



2.1. Presentation Layer

This is the top-most layer and the sole point of interaction for the end-user.

- **Command Line Interface (CLI):** The user interacts with the entire system through the CLI. It is responsible for taking user input (like the target domain), initiating scans, handling user authentication, and displaying the final results. It communicates with the services in the Process Centric Services Layer.

2.2. Process Centric Services Layer

This layer is responsible for orchestrating the business logic and managing the overall workflow of a scan.

- **Authentication:** This service manages user registration and login. It interacts with the *UserDB adapter* to verify credentials and issue authentication tokens (JWTs). These tokens are then verified by the other process centric services using a shared secret, before granting access to the users.
- **Scan Host:** This is the central orchestrator. When a user initiates a scan via the CLI, the *Scan Host* service is invoked. It calls upon the various services in the Business Logic Layer (Attack surface mapping, Tech detection, Employee data analysis) to gather all the required information about the target.
- **Scan Storage:** After a scan is completed by the *Scan Host*, this service is responsible for persisting the results. It interacts with the *ScanDB adapter* to save the scan report into the database in the form of a json object. It also handles retrieving and deleting past scan results.

2.3. Business Logic Services Layer

This layer contains the core application logic, where data is processed and aggregated into meaningful information.

- **Attack surface mapping:** This service is responsible for identifying the digital footprint of the target domain. It discovers subdomains, resolves their IP addresses, and gathers information about the hosting infrastructure. It achieves this by coordinating calls to several adapter services, including the *Search API adapter*, *CRT.sh API adapter*, *DNS adapter*, and *IPinfo adapter*.
- **Tech detection:** This service identifies the technologies used by the target's web applications. It communicates with the *BuiltWith API adapter* to gather details about frameworks, libraries, and other software components.
- **Employee data analysis:** This service focuses on gathering information about employees associated with the target domain, primarily by finding publicly available email addresses. It relies on the *Hunter.io API adapter* for this purpose.

2.4. Adapter Services Layer

The Adapter Layer provides a crucial abstraction between the business logic and the underlying data sources. Each adapter wraps a specific database or external API, exposing a consistent interface to the services in the Business Logic Layer.

- **UserDB adapter:** Stores users authentication data by communicating directly with the *UserDB (MongoDB)*.
- **Search API adapter:** Interfaces with a search engine (like Google) to find subdomains of the target.
- **CRT.sh API adapter:** Connects to the *crt.sh* service to find subdomains from certificate transparency logs.
- **DNS adapter:** Interacts with a *DNS resolver* to get IP addresses for given domain names.
- **IPinfo adapter:** Connects to the *IPinfo* service to gather geographical and network information about IP addresses.
- **BuiltWith API adapter:** Interfaces with the *BuiltWith* service to perform technology detection.
- **Hunter.io API adapter:** Connects to the *Hunter.io* API to find employee email addresses.
- **ScanDB adapter:** Manages the storage and retrieval of scan results from the *ScanDB (MongoDB)*.

2.5. Data Services Layer

This is the lowest layer of the architecture, representing the actual data stores and third-party services.

- **UserDB, ScanDB (MongoDB):** A NoSQL database chosen to store users information and scan results.
- **Google search API, CRT.sh API, DNS resolver, IPinfo, Wappalyzer API, Hunter.io API:** These are external, third-party services and APIs that provide the raw data for the reconnaissance process.

3. Conclusion

The described service-oriented architecture provides a robust and scalable foundation for the domain intelligence platform. By decoupling concerns into distinct layers and services, the system is easy to maintain, extend, and scale. The use of adapters for all external interactions isolates the core business logic from the specifics of external data sources, allowing for greater flexibility and resilience to change.