CURIUM

PSTVA

User Guide

Introduction

The PSTVA is an extensible platform for orchestrating network and web security scans using tools like Nmap, Nikto, Ffuf, and Kerbrute. It manages results in a Neo4j graph database, with FastAPI for REST orchestration, Docker for tool isolation, Redis for task brokering, and Celery for asynchronous processing. This manual provides step-by-step instructions for installation, operation, and advanced usage tailored to cybersecurity professionals.github

System Overview

Architecture Components

- **API Layer**: FastAPI handles scan requests, project management, JWT authentication, and OpenAPI docs.
- **Task Layer**: Celery workers execute scans in Docker containers, parse outputs, and chain tasks via Redis.
- **Data Layer**: Neo4j stores hosts, services, vulnerabilities as graph nodes with relationships; Redis tracks job states.

Scan Pipeline

1. Client API request → Auth → Task to Redis.
2. Celery launches Docker scanner → Parses output → Persists to Neo4j.
3. Results queryable via API with graph topology visualization.neo4j

Prerequisites

The toolkit requires Docker for all services including Neo4j, Redis, and scanners.

- **Operating System:** Linux, macOS, or Windows (with Docker Desktop).
- **Software Requirements:**
    - Docker Engine (20+), Docker Compose V2.
    - Git, Make utility.
- **Resources:** 4GB+ RAM, 2+ CPU cores; 8GB recommended for concurrent Ffuf scans.neo4j
- **Ports:** 8000 (API), 7474/7687 (Neo4j), 6379 (Redis), 5672 (Celery).

Verify: docker --version && docker compose version.neo4j

Download

Obtain source code, Dockerfiles, and Makefile:

```
git clone https://git.aegisresearch.eu/aegis-devs/aegis-security-assessment-toolkit.git
```

```
cd aegis-security-assessment-toolkit
```

Build Services

Makefile builds FastAPI, Celery, and scanner images:

- **Build all:**

```
make build-all
```

Check docker-compose.yml for Neo4j/Redis configs.neo4j

Startup Procedure

- **Start services:**

```
make start
```

- **Logs:**

```
make logs
```

- **Quick setup:**

```
make boot-start
```

Accessing the API

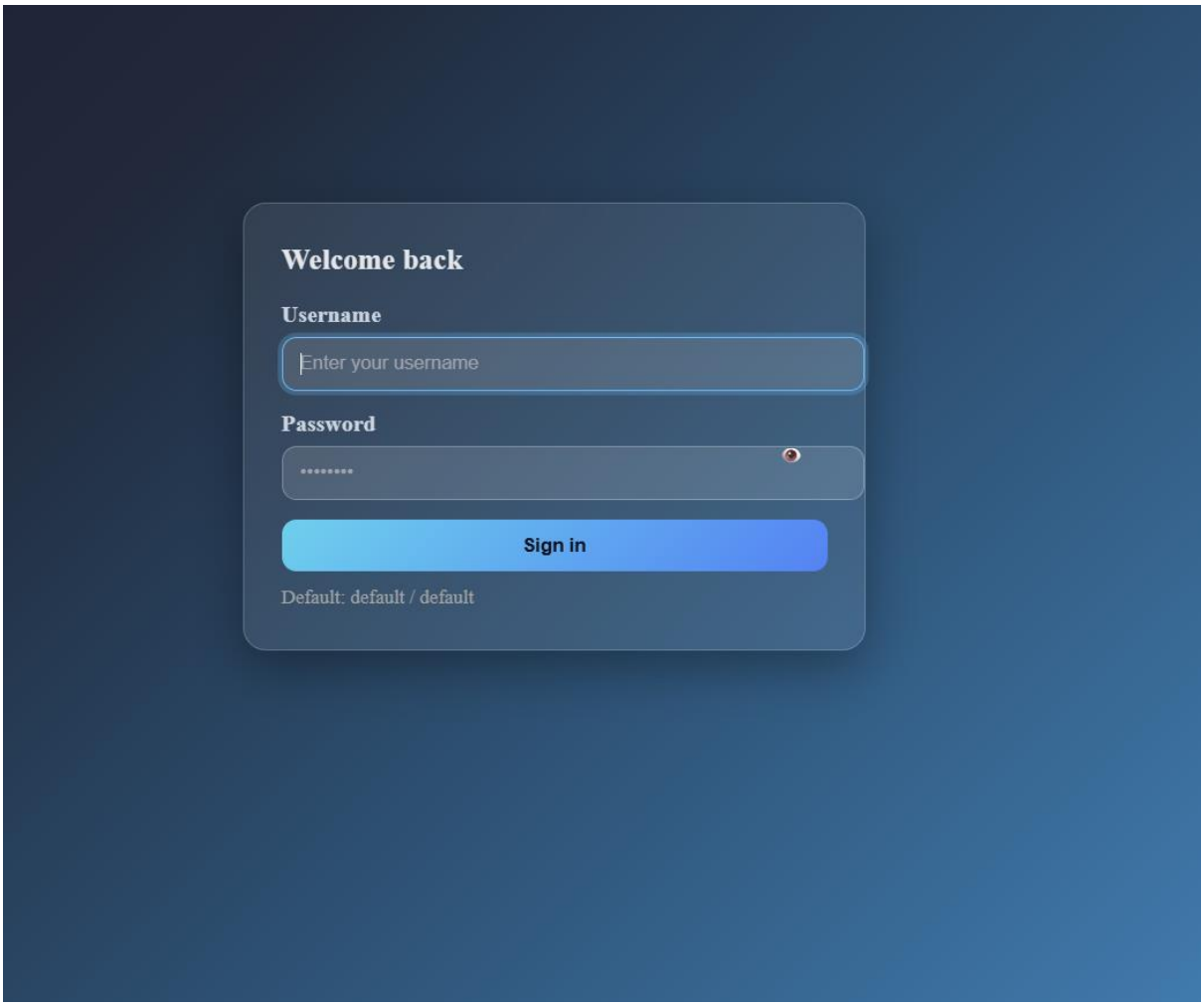
- Base: http://curium.localhost
- Docs: http://curium.localhost/back/docs
- Health: curl http:/curium.localhost/health
- Auth: JWT/API Key via /auth; scan endpoints for Nmap/Nikto/Ffuf/Kerbrute

Stopping Services

```
Bash make stop
```

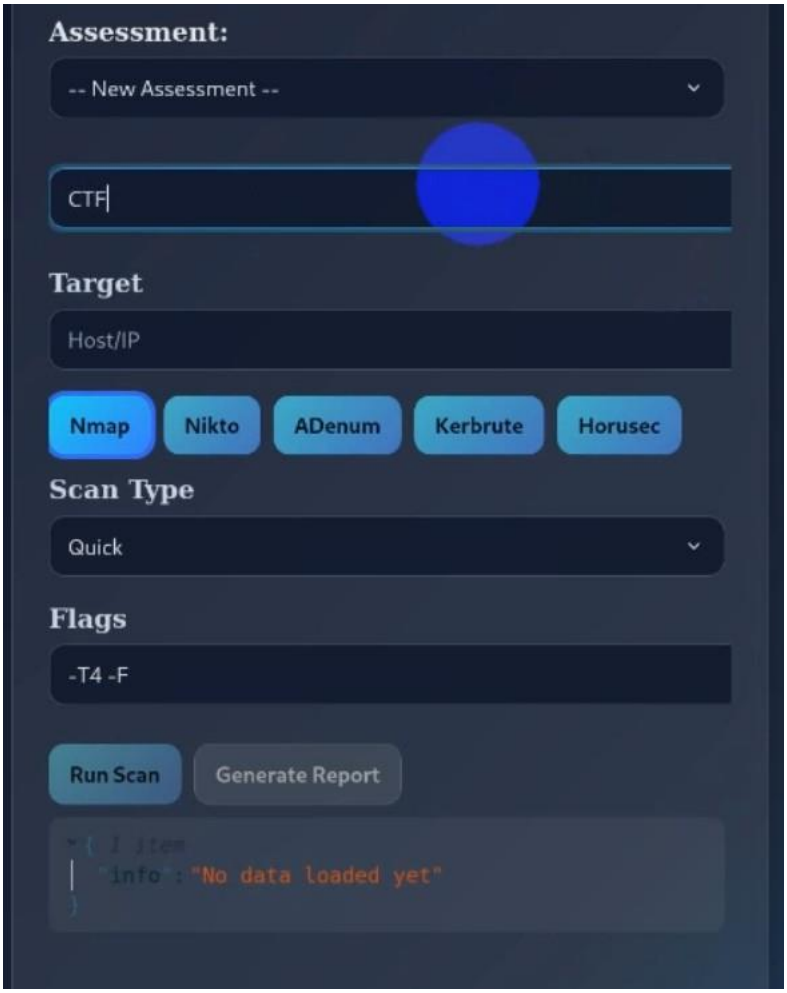
Scan Execution Workflow

1. Log in



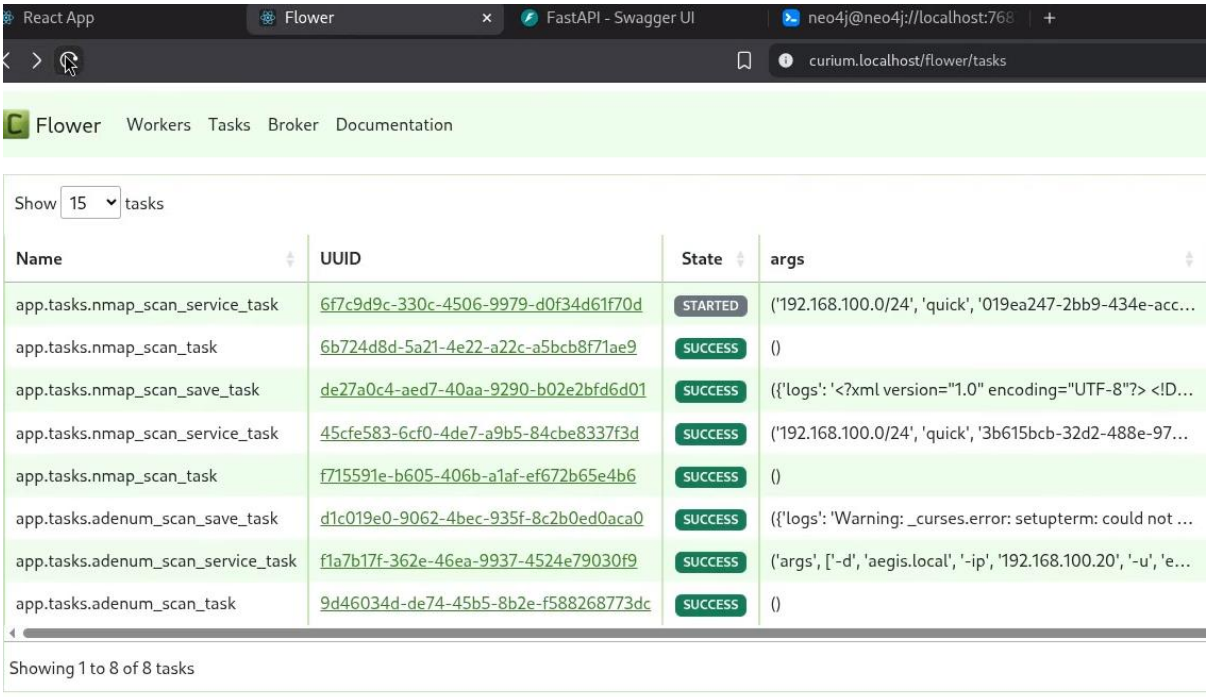
2. Create a project

3. Select tool



4. Selects Scan type

5. Check the process on celery tab

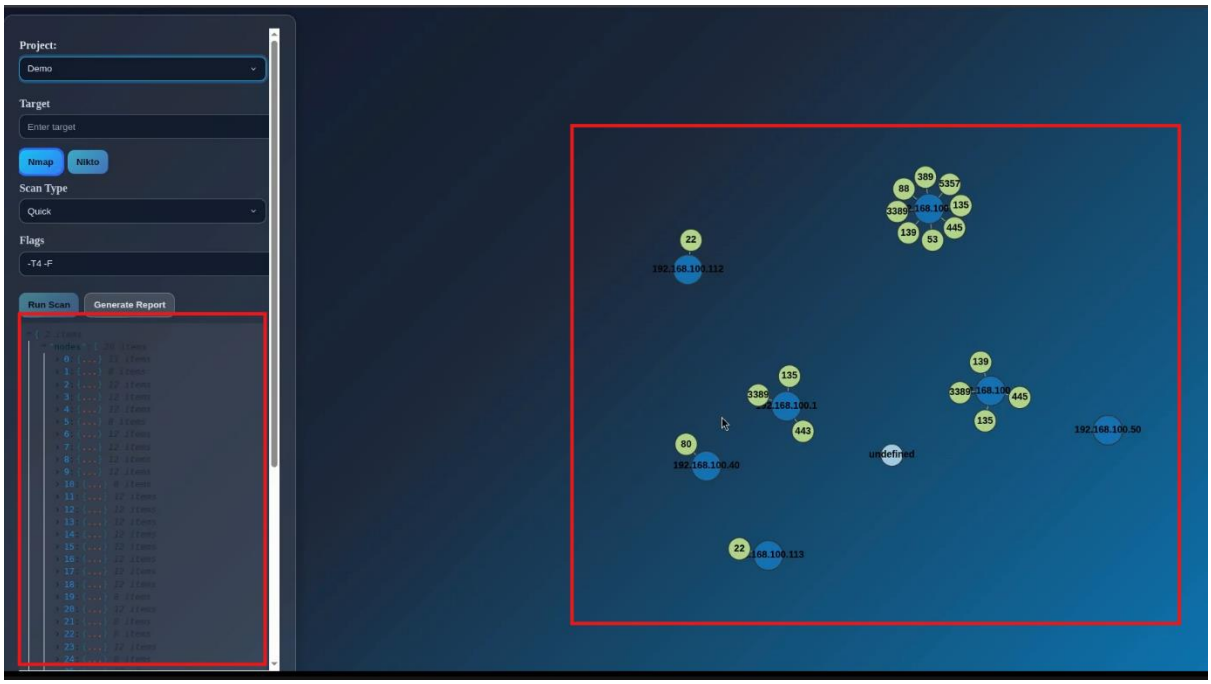


6. View derails of specific process

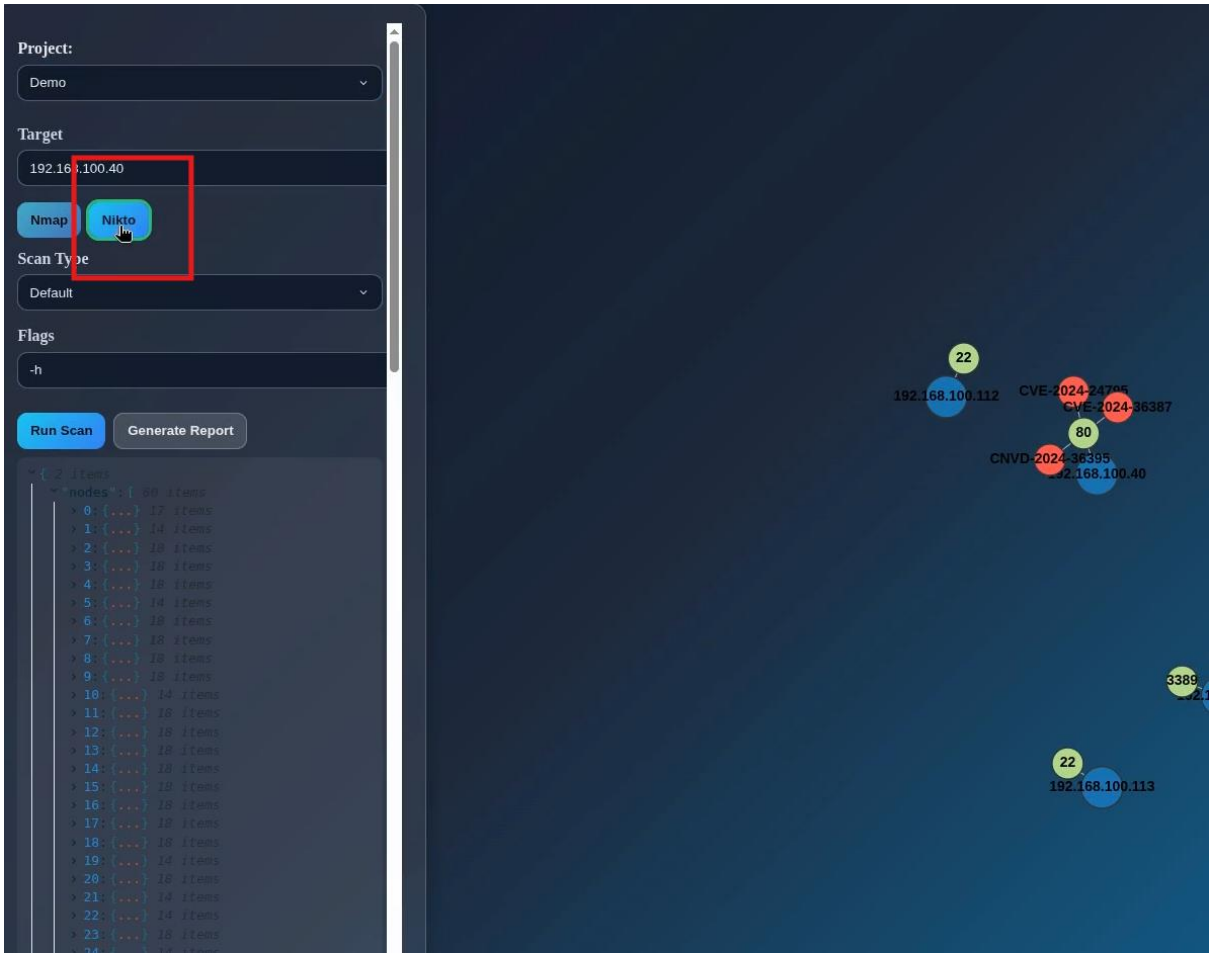
app.tasks.nikto_scan_save_task daa01f05-8b32-4dfa-9dfd-d3169a8391ac

Name	app.tasks.nikto_scan_save_task
UUID	daa01f05-8b32-4dfa-9dfd-d3169a8391ac
State	SUCCESS
args	({'logs': '- Nikto v2.1.5 ---------- + Target IP: 192.168.100.40 + Target Hostname: 192.168.100.40 + Target Port: 80 + Start Time: 2025-10-08 08:52:17 (GMT0) ---------- + Server: Apache/2.4.58 (Ubuntu) + The anti-clickjacking X-Frame-Options header is not present. + OSVDB-3268: /: Directory indexing found. + No CGI Directories found (use \'-C all\' to force check all possible dirs) + Allowed HTTP Methods: POST, OPTIONS, HEAD, GET + OSVDB-3268: /./: Directory indexing found. + OSVDB-3268: /?mod=node&nid=some_thing&op=view: Directory indexing found. + OSVDB-3268: /?mod=some_thing&op=browse: Directory indexing found. + /./: Appending \'/./\' to a directory allows indexing + OSVDB-3268: //: Directory indexing found. + //: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page. + OSVDB-3268: /?Open: Directory...', ...},)
kwargs	{'project_id': '019ea247-2bb9-434e-accb-221a1f572259'}
Result	{'log_id': None, 'vulnerabilities': 35, 'host': '192.168.100.40', 'ip': '192.168.100.40', 'port': '80'}
Received	2025-10-08 08:52:22.730908 UTC
Started	2025-10-08 08:52:22.732705 UTC
Succeeded	2025-10-08 08:52:23.028955 UTC
Retries	0
Worker	celery@487a32147beb
Timestamp	2025-10-08 08:52:23.028955 UTC
Runtime	0.2965762997046113
Clock	177229
Root	<Task: app.tasks.nikto_scan_task(f1ebec5d-1648-465b-812b-a8b96f5f7db8) SUCCESS clock:177217>
Root id	f1ebec5d-1648-465b-812b-a8b96f5f7db8
Parent	<Task: app.tasks.nikto_scan_service_task(1936ab30-2fc7-40ab-be3c-e2b4ff46b491) SUCCESS clock:177227>

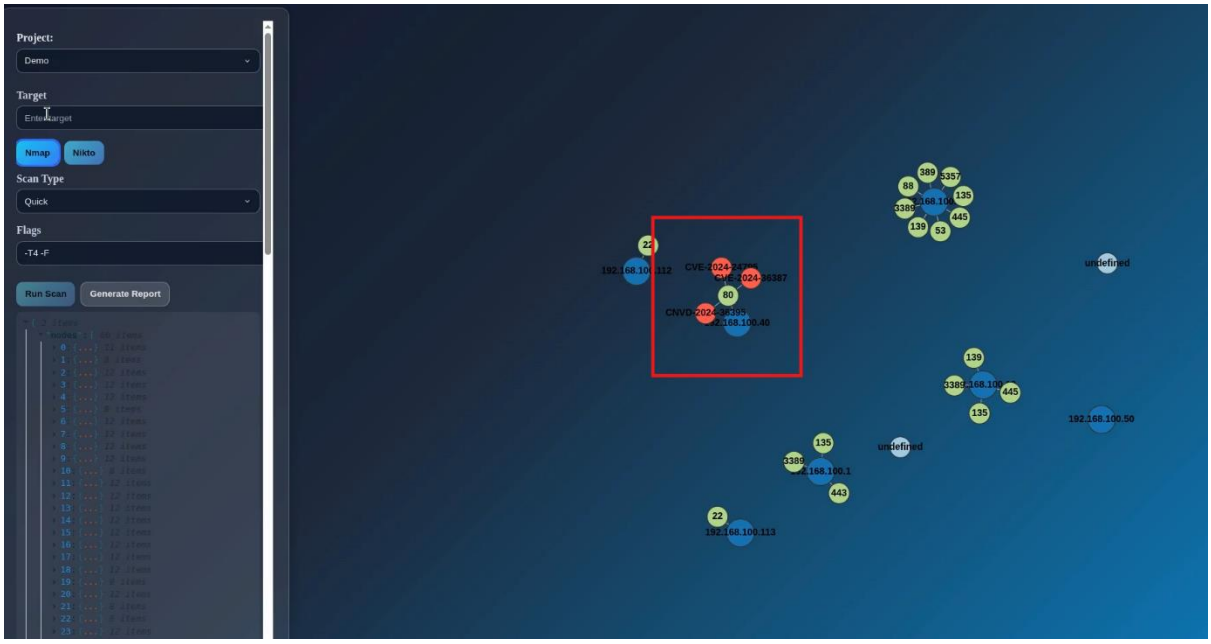
7. Get results of the scan when the process succeed



8. Select Vulnerability assessment (Nikto tool)



9. Get Vulnerability assessment results



10. Generate Report
11. Final results (PDF file)

Confidential – For Authorized Use Only

Web Application Assessment Report

Web Application Assessment Report

Client: Web Application Assessment Report

Prepared By: Security Team

Version: 1.0

Date: 2025-10-08

Classification: Confidential

12. Validate security posture through CVE

- Reference: https://vulners.com/githubexploit/B5E74010-A082-5ECE-AB37-623A5B33FE7D

4.0.3 3. CVE-2024-38474 (Critical)

Field	Value
ID	192.168.100.40:80-21
Severity	Critical
Affected	192.168.100.40:80 (Apache/2.4.58 (Ubuntu) Apache/2.4.58 (Ubuntu) -)
Status	Open
CVSS	9.8
Remediation Summary	Patch or mitigate the affected service.
References	https://vulners.com/cve/CVE-2024-38474

Vulnerability observed on `192.168.100.40:80`.

- Service: Apache/2.4.58 (Ubuntu)
- Product: Apache/2.4.58 (Ubuntu)
- Version: -
- Reference: https://vulners.com/cve/CVE-2024-38474

Using the Toolkit

- **Projects**: CRUD via API; group scans.
- **Scans**: Initiate Nmap (ports), Nikto (web), Ffuf (dirs), Kerbrute (enums).
- **Results**: GraphQL-like queries; export JSON/CSV.
- **Logs/Metadata**: Task states in Redis, persisted to Neo4j.

Advanced Features

- **Extend Tools**: Add Dockerfiles for new scanners; register in Celery tasks.
- **Task Chains**: Config YAML for Nmap→Nikto sequences.
- **Neo4j Custom**: Cypher queries for vuln topology; extend schemas.
- **Scaling**: make workers-up N for distributed scans.

Troubleshooting

- Full diagnostics: make logs.

Security Considerations

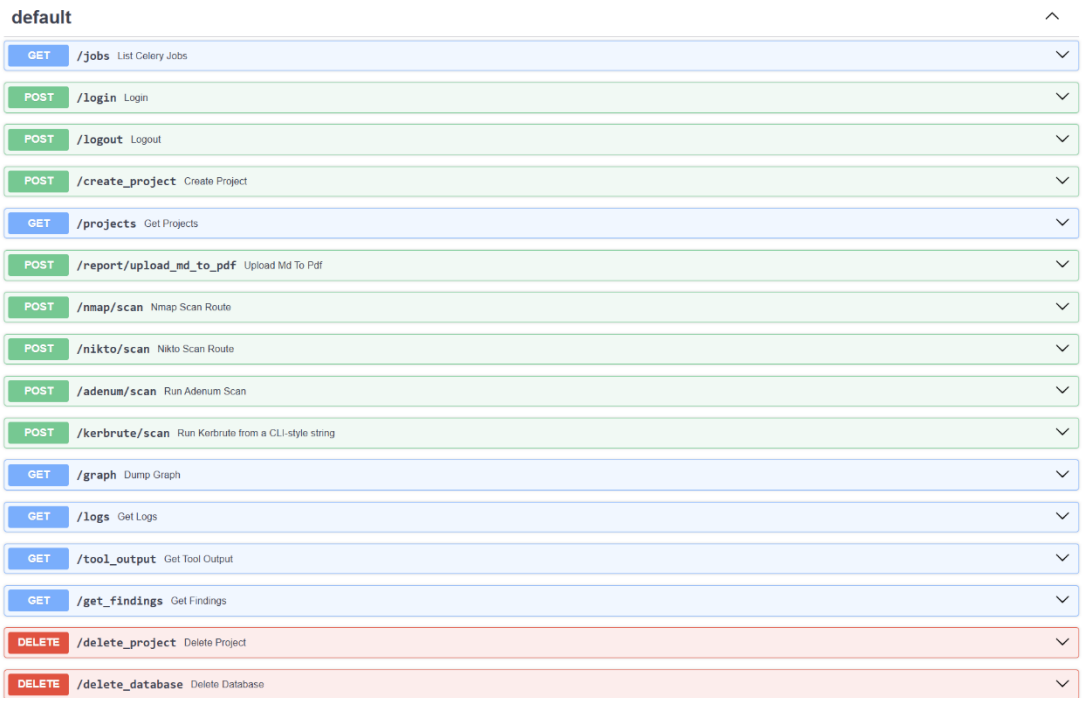
- Use JWT for API; rotate keys.
- Docker isolation prevents scanner escapes.
- Neo4j access restricted; validate scan targets.
- Run as non-root; audit Celery logs for anomalies.

Appendices

Glossary

- **Celery**: Task queue for async scans.
- **Neo4j**: Graph DB for vuln relationships.

API access through Swagger



Support

Issues: contact Aegis Research security-team@aegisresearch.eu