KintsuKai

CVSSense

CVSSense Community Edition

System Documentation

Published by Kintsukai

Date: 08.07.2026 | **Version:** 0.5

Change History

This section tracks the revision history and version control of the CVSSense documentation.

Version	Date	Author	Description
v0.5	08.07.2026	K3res	Initial draft version of the system documentation (Work in Progress).

Table of Contents

8. External Knowledgebase & Legal Disclaimers

Quick Start

To immediately deploy the Community Edition using Docker, run the following commands:

```
git clone git@github.com:K3res/cvssense-community-docker.git
cd cvssense-community-docker
docker-compose up -d
```

Access the application at `http://localhost:8080/dashboard`.

1. Product Requirements & System Overview

1.1 Product Vision and Purpose

CVSSense is a self-hosted CVSS 4.0 scoring workbench for analysts who need to evaluate vulnerabilities in a structured, time-efficient and reproducible way.

The application guides users through the scoring process by making CVSS metric choices transparent, showing score changes immediately and producing a clear vector, severity rating and assessment result. This reduces manual lookup effort, shortens repetitive scoring work and helps analysts apply the CVSS methodology consistently.

CVSSense is especially useful when vulnerability information must be translated into a documented score that can be reviewed, shared, stored as a preset or exported for later use in operational, audit or reporting workflows.

The Community Edition is tailored for individual security analysts and small operational teams that need an efficient local tool without enterprise infrastructure overhead or external data exposure.

1.2 Target Audience and User Personas

This documentation and the underlying system are designed for the following core technical personas:

- **IT Security Professionals:** Responsible for interpreting vulnerability data, determining systemic risks, and prioritizing patching efforts across corporate infrastructure.
- **Vulnerability Analysts:** Power users who perform daily technical assessments, break down vulnerability disclosures, and generate precise CVSS vectors.
- **System Administrators:** Responsible for the deployment, container hardening, initial port configurations, and ongoing maintenance of the self-hosted instance.

Prerequisite Note: While no prior programming or software development knowledge is required to operate the Community Edition, users are strongly encouraged to possess a baseline familiarity with information security concepts and the fundamental structure of the CVSS metric framework.

1.3 Scope and Boundary Conditions (Community Edition)

To maintain a clear separation of concerns across the product lifecycle, the operational scope of the Community Edition is explicitly defined below:

In-Scope Features:

- **Full Offline Capability:** Once the container image is pulled from the repository, the application executes entirely within the local host boundary. It operates without an internet connection and processes no metrics or metadata via external APIs.
- **Core CVSS 4.0 Calculation:** Real-time calculation engine that instantly evaluates base scores, severity rankings, and full vector strings based on user selections in the Metric Assessment Area.
- **Decision Assistant (Simple Mode):** A guided, 11-step sequential wizard designed specifically for experienced analysts who favor an accelerated, structured workflow over manual metric tables.

- **Local Preset Management:** Provision of local storage capability to save up to 5 individual scoring profiles directly within the analyst's browser cache.

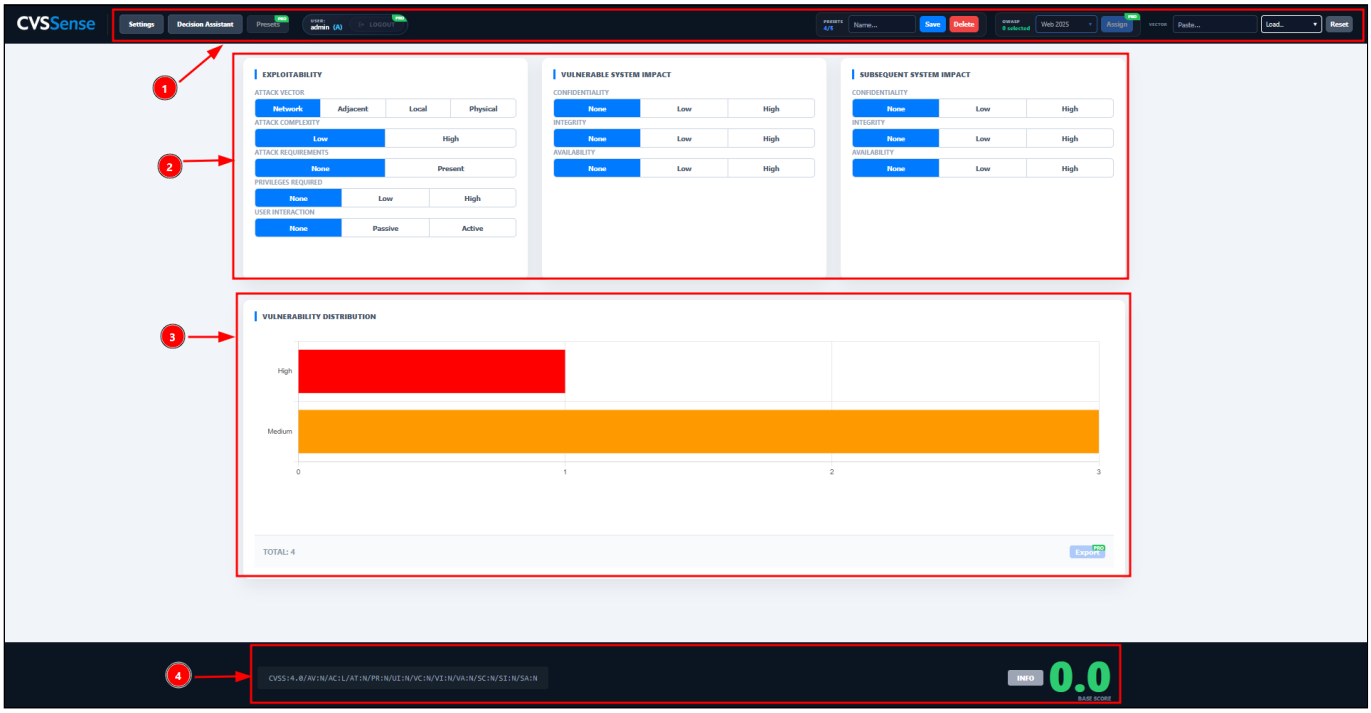
Out-of-Scope Features (Reserved for PRO / Enterprise Tiers):

- **Multi-User Role Management & RBAC:** Fine-grained Role-Based Access Control and centralized user account administration.
- **Persistent Server-Side Storage:** Database-backed preset storage and cross-device synchronization.
- **Audit Logging & Compliance Tracking:** Persistent audit trails and automated OWASP Top 10 / CWE mapping.
- **AIKA — Local AI Assistant:** On-device AI assistant for guided scoring support.
- **Scoring Wiki & Metric Explanations:** In-app contextual guidance and reference content for each CVSS metric.
- **Advanced Scenario-Based Assistant:** Extended decision wizard for complex, multi-environment vulnerability scenarios.
- **Multi-Language Support:** Localization and internationalization of the user interface.
- **Chart Diagram Export:** Export of vulnerability distribution charts as image files.
- **Scoring & Preset REST API:** Authenticated API access for CVSS calculations, batch scoring and server-side presets. Basic third-party integration is available in PRO; fully custom report workflows are reserved for Enterprise.
- **Custom AI Provider Integration (BYOK) *(Enterprise only)*:** Bring-Your-Own-Key connectivity to external AI providers (e.g., Google Gemini, OpenAI).
- **Automated Risk & Executive Summaries *(Enterprise only)*:** AI-generated stakeholder reports translating CVSS vectors into business-risk language.

2. UX Design

2.1 Core Dashboard Layout and Functional Zones

The CVSSense user interface is optimized for single-screen efficiency, minimizing vertical scrolling so analysts can maintain a steady line of sight across all metric variations. The main workspace is divided into four distinct functional zones:



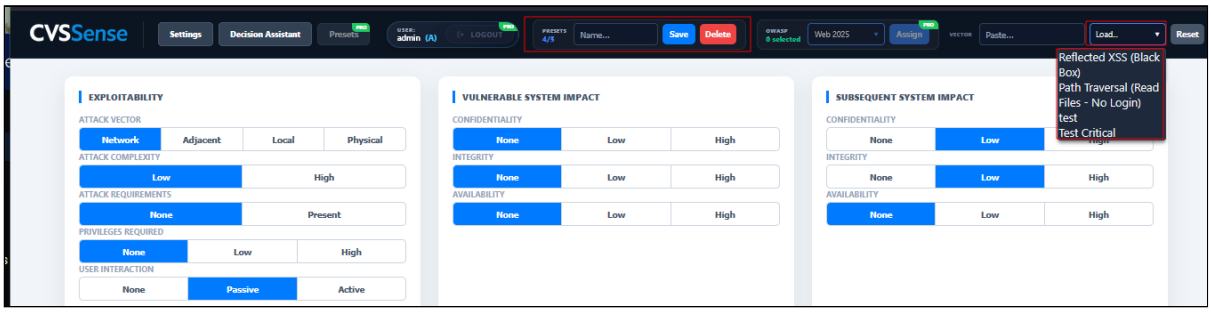
Zone ID	UI Component	UX Purpose & Interaction Model
1	**Header & Navigation**	Central control hub for system navigation, profile management (Presets), and manual CVSS vector input.
2	**Metric Assessment Area**	The primary workspace containing segmented control grids corresponding to the official FIRST.org CVSS 4.0 metric groups. Users click directly on metric blocks to set values, giving immediate tactile and visual feedback on the currently selected vector path.
3	**Vulnerability Distribution Chart**	An on-demand data visualization segment that renders a graphical breakdown of saved presets categorized by severity levels. This component can be toggled on or off depending on user preference via the Settings panel.
4	**Result Footer**	A sticky, high-contrast bottom bar that aggregates real-time calculation outputs. It displays the final numeric Base Score (0.0–10.0), a color-coded Severity label, and the raw CVSS 4.0 Vector String. Clicking anywhere on the Vector String automatically copies it to the system clipboard.

2.2 Deep Dive: Header & Toolbar (Zone 1)

This section details the functional workflows available within the header navigation and toolbar:

- **Navigation & System Control**:
  - **Settings**: Located on the far left, providing access to system-wide configurations.
  - **Decision Assistant**: Switches the view from the manual dashboard to the guided wizard workflow.

- **Preset Management**:
  - **Saving**: Enter a custom name in the profile input field and click `Save` to store your current configuration.
  - **Loading & Deleting**: Select a profile from the dropdown to load it; click `Delete` to remove it.

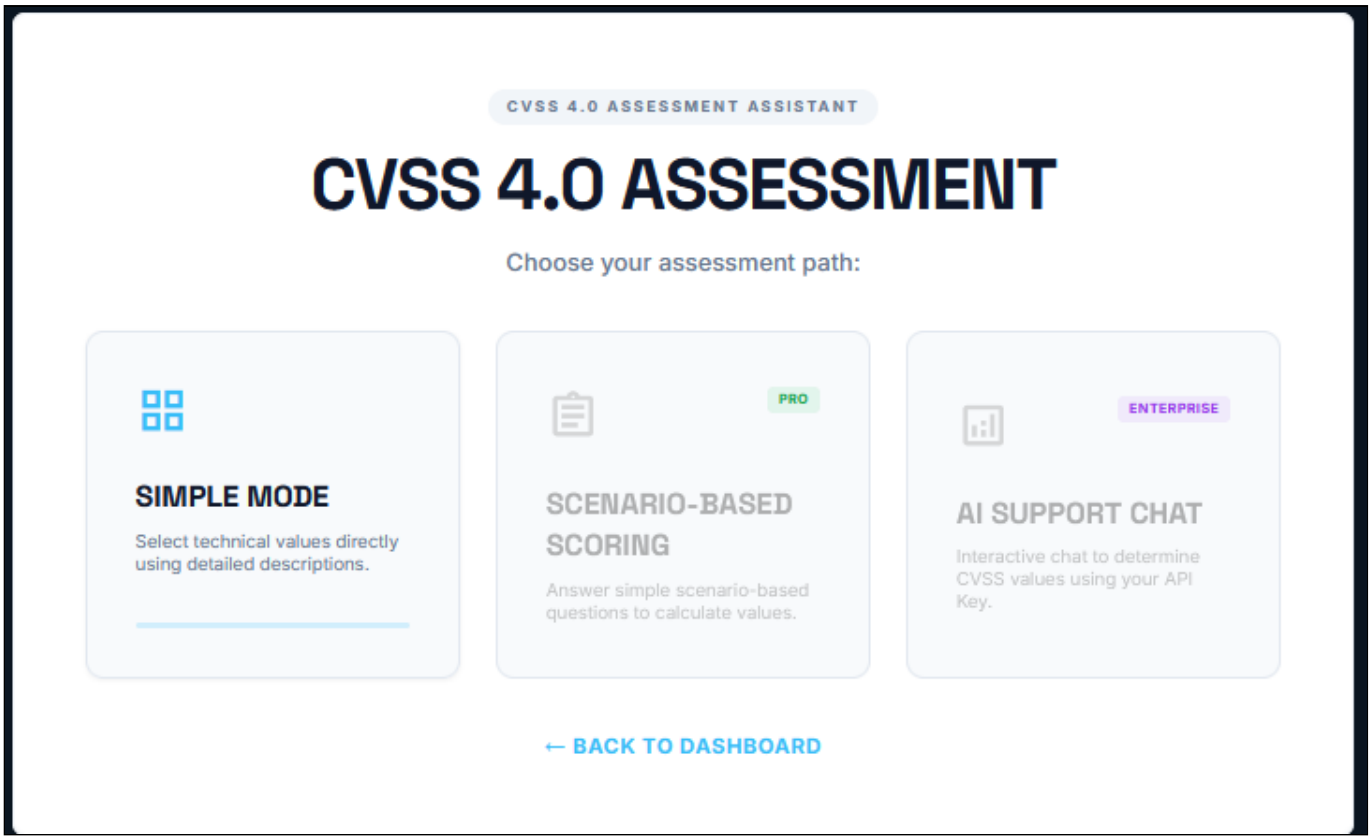


- **Scoring & Vector Utility**:
  - **Reset**: Click the `Reset` button (far right) to revert all metrics and the Base Score to `0.0`.
  - **Vector Input**: Paste a raw CVSS string into the `Vector` field to auto-populate metrics.

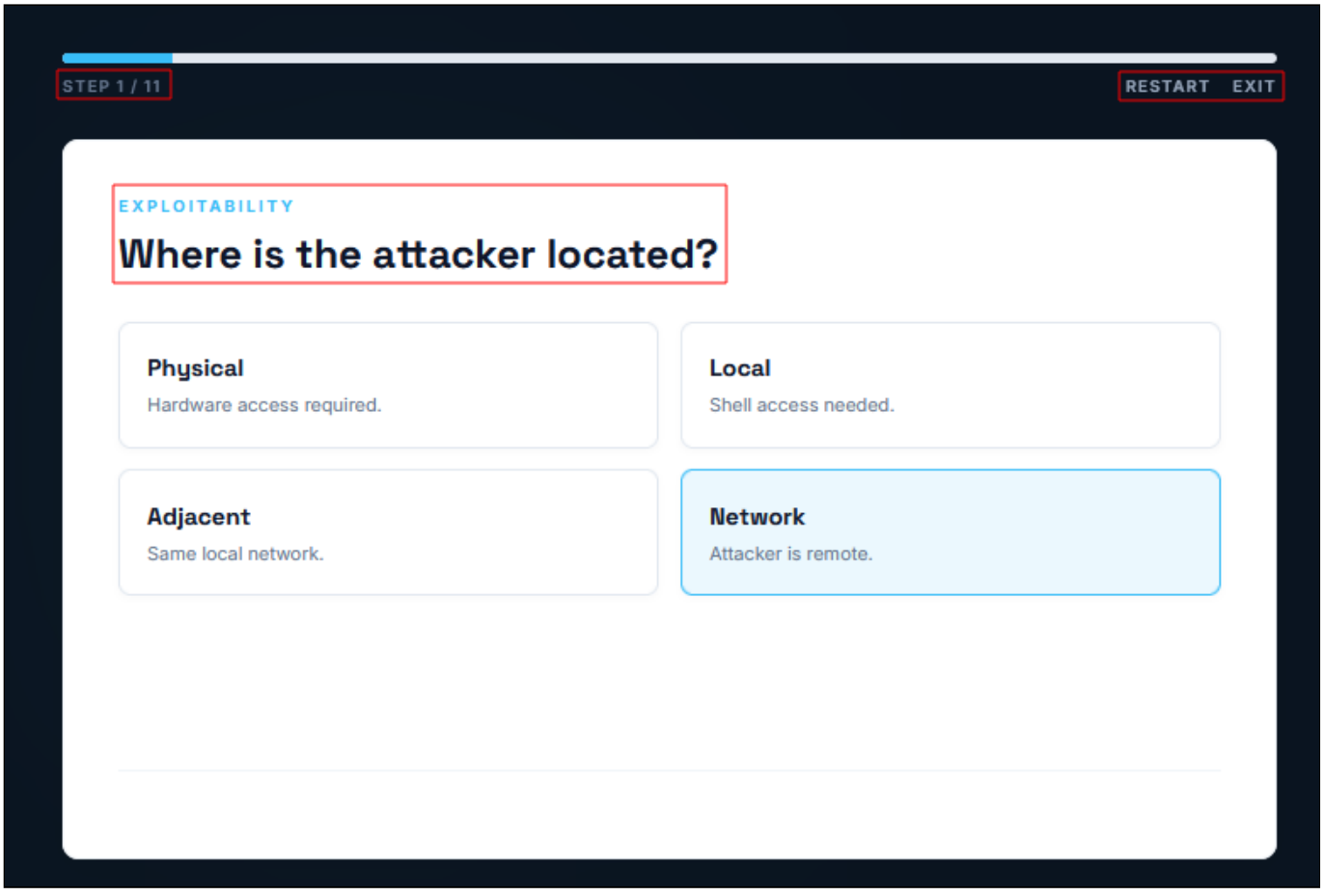
2.3 Decision Assistant Wizard & Component Layout

The Decision Assistant provides a sequential, cognitive-load-reducing alternative to the manual matrix. It translates abstract CVSS metrics into clear, isolated technical questions.

Upon entering the Decision Assistant, users are first prompted to select their preferred assessment path depending on their active license tier:



2.3.1 Interaction Pattern & Step Sequencing

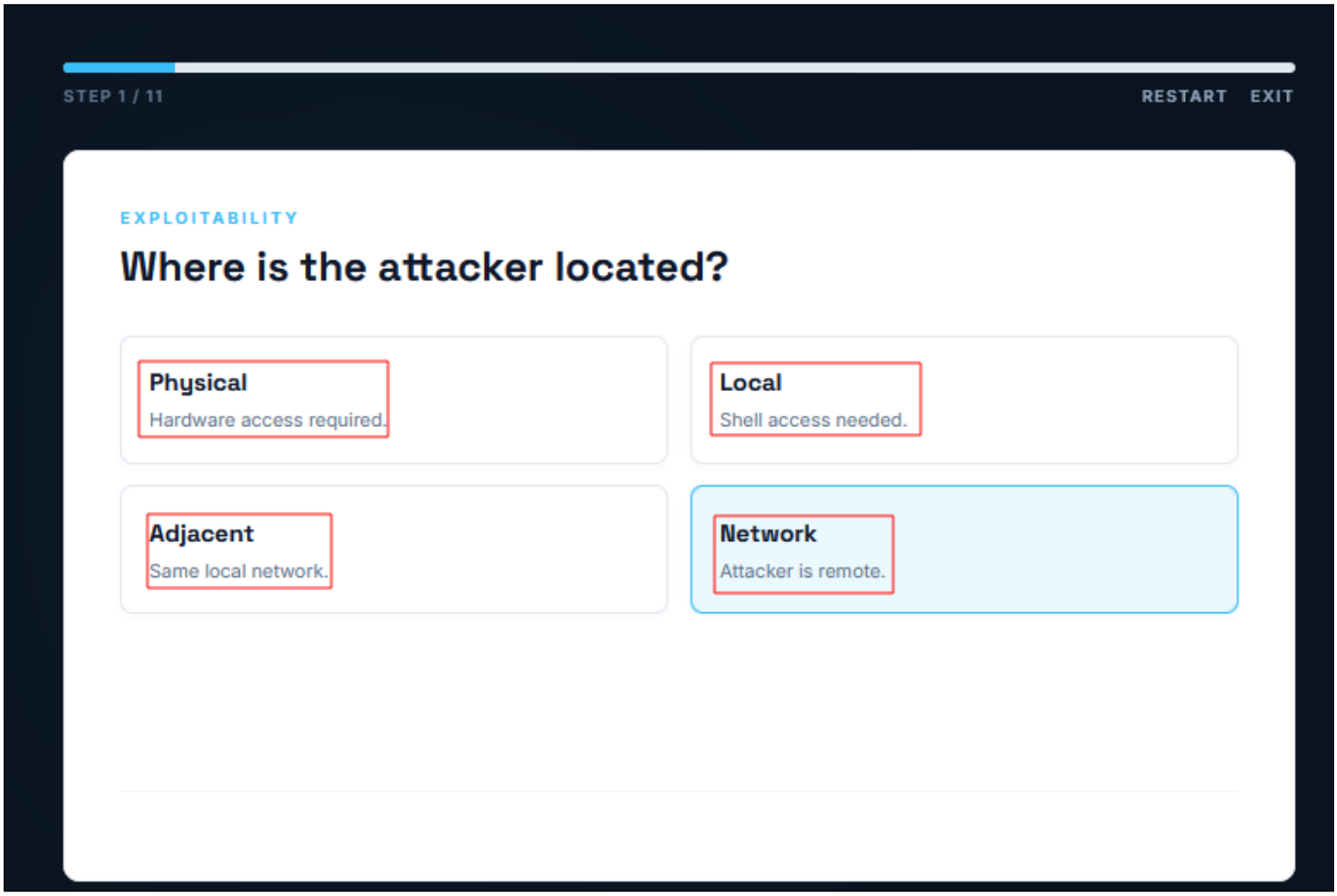


- **Wizard Framework:** The assistant is built as an 11-step linear progress wizard. Users cannot skip ahead without selecting a valid metric state, ensuring data completeness.

- **Category Breadcrumbs:** The top-left of the wizard display features a tracker indicating the current metric group tier (e.g., `EXPLOITABILITY`).
- **Clean Phrasing:** Each step uses a prominent, human-readable question (e.g., "*Where is the attacker located?*") rather than raw CVSS metric shorthand.
- **Header Controls**: Located in the top right corner:
    - `RESTART`: Clears the current progress and begins a new guided assessment from step one.
    - `EXIT`: Closes the Decision Assistant and returns to the main dashboard without applying the calculated vector.

2.3.2 Option Selection Card Architecture

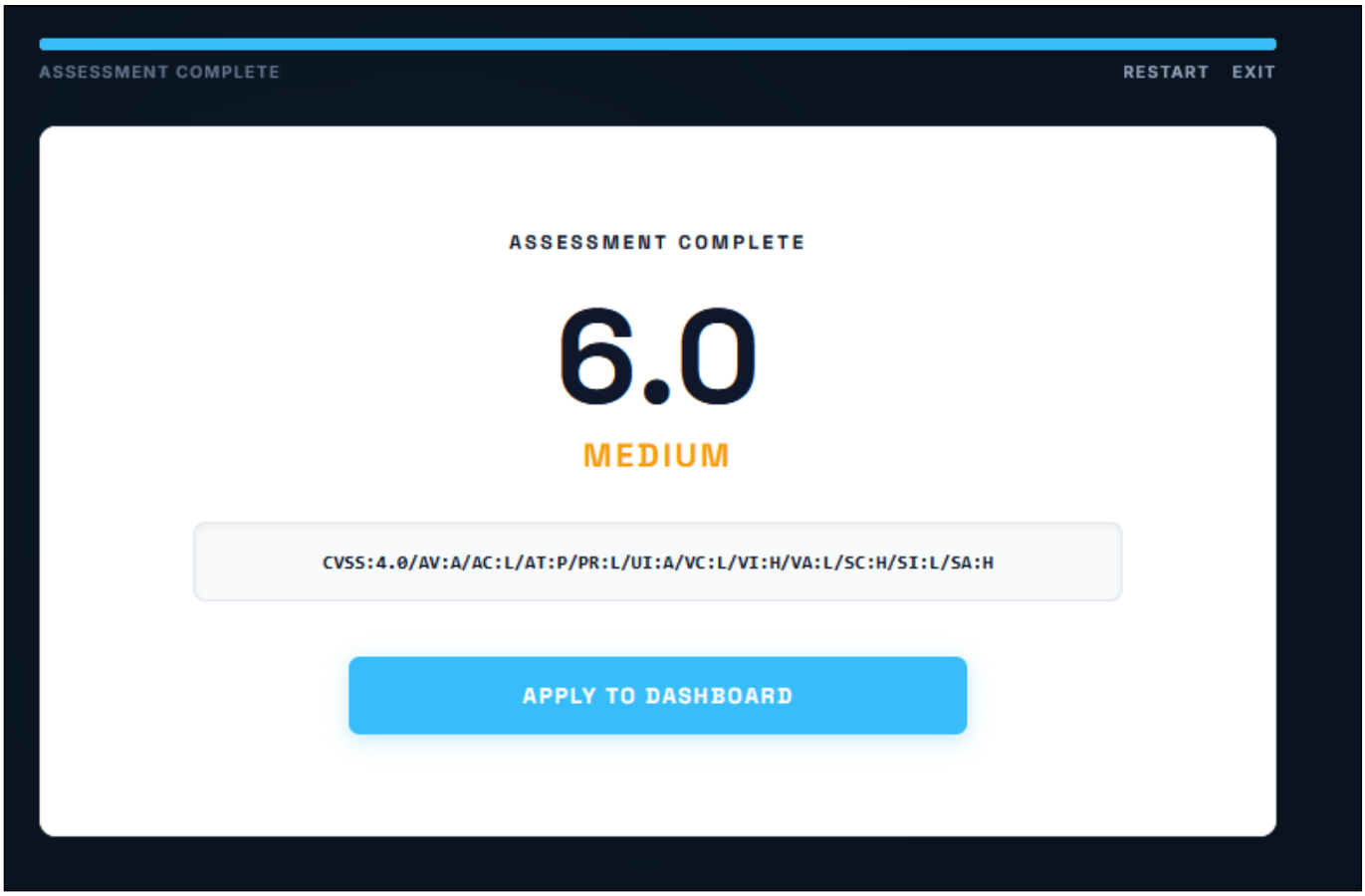
Options are presented in a clean card-based option grid layout. Each selection card features a specific typographical hierarchy designed to accelerate expert decision-making:



1. **Metric Title (Primary Label):** Bold, high-contrast text stating the formal CVSS value classification (e.g., **Physical**, **Local**, **Adjacent**, **Network**).
2. **Context Annotation (Secondary Sub-label):** Placed directly beneath the primary label in a muted font color, providing immediate, real-world technical qualifiers (e.g., "*Attacker is remote*" or "*Shell access needed*"). This eliminates the need to cross-reference long help wikis during an active triage session.
3. **State Feedback:** When a card is hovered over or selected, it transitions to a highlighted blue border state to visually confirm user focus.

2.3.3 Assessment Completion & Output

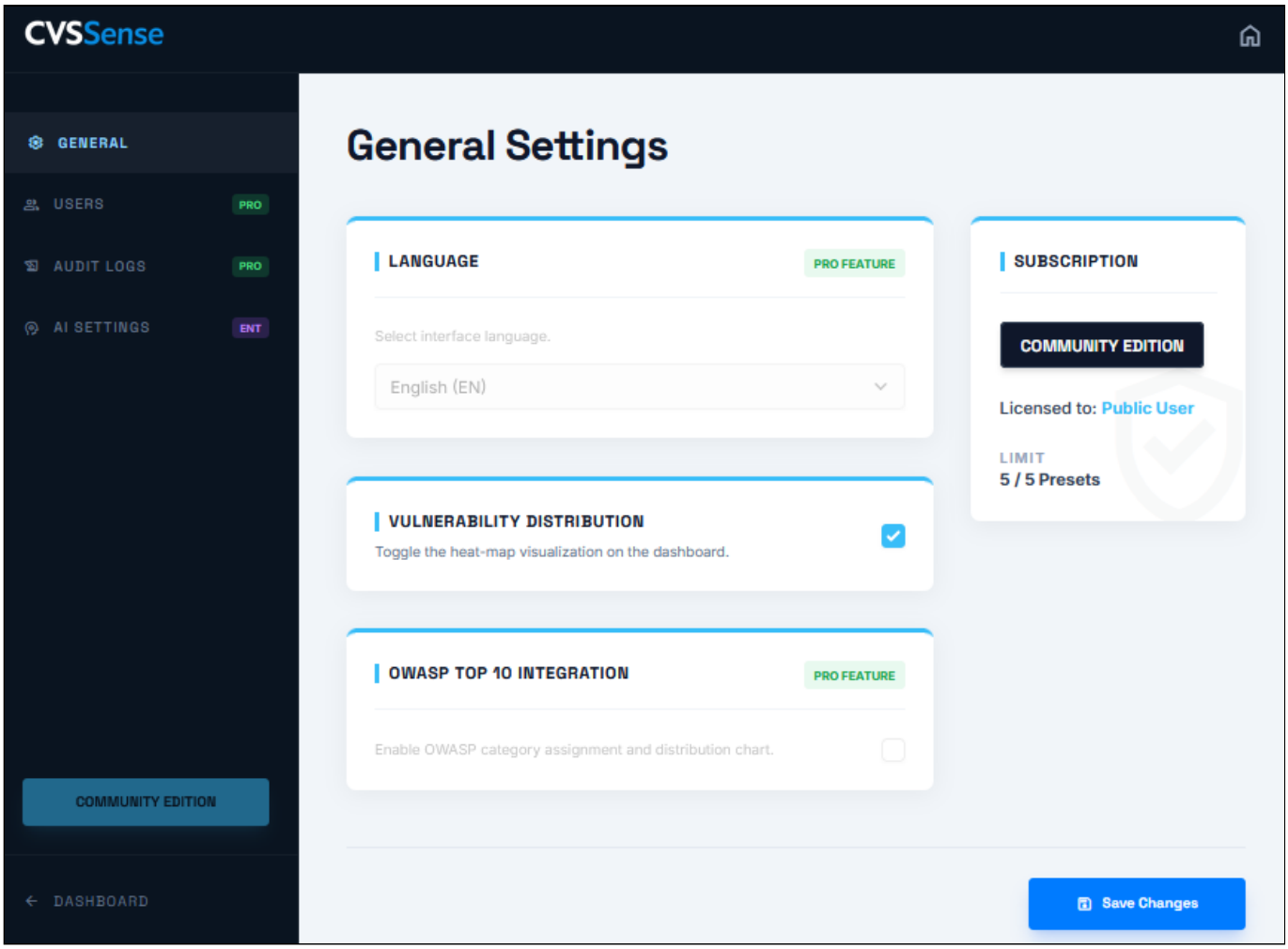
Upon finishing the final step of the Decision Assistant wizard, the user is presented with the **Assessment Complete** summary screen. This view aggregates the step-by-step selections into a final, actionable result.



- **Final Score & Severity**: Prominently displays the calculated numerical Base Score (e.g., **6.0**) alongside its corresponding color-coded severity level (e.g., **Medium**).
- **Vector String**: Displays the complete, machine-readable CVSS 4.0 vector string generated dynamically from the wizard answers.
- **Apply to Dashboard**: The primary action button that transfers the newly calculated vector directly back to the main manual Dashboard. From there, the user can save it as a Preset or make further environmental adjustments.

2.4 Settings Panel Configuration Interface

The configuration space uses a standard Master-Detail layout pattern to keep administrative tasks distinct from active scoring workflows.



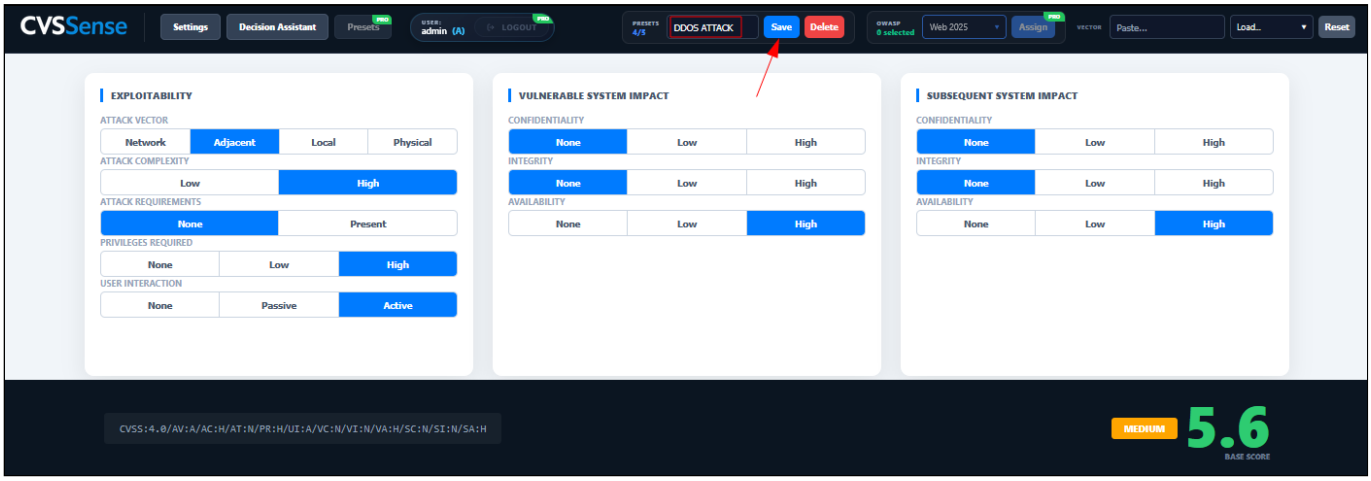
- **Left Navigation Sidebar:** A persistent vertical navigation list allows users to hop between configuration categories.
- **Tab-Based Feature Matrix:** To maintain transparency regarding tier features, tabs for advanced editions are visible but clearly badge-restricted:

Tab Name	UI Status (Community Edition)	Description & Visibility
General	**Active (Partial)**	Allows the user to toggle the visibility of the dashboard's Vulnerability Distribution Chart.
Users	*Locked (PRO Indicator)*	Greyed out; highlights multi-user role and identity configuration restrictions.
Audit Logs	*Locked (PRO Indicator)*	Greyed out; reference pane for historical logging mechanics.
AI Settings	*Locked (Enterprise Indicator)*	Greyed out; configuration panels for LLM endpoint bindings.

2.5 Preset Management Workflow

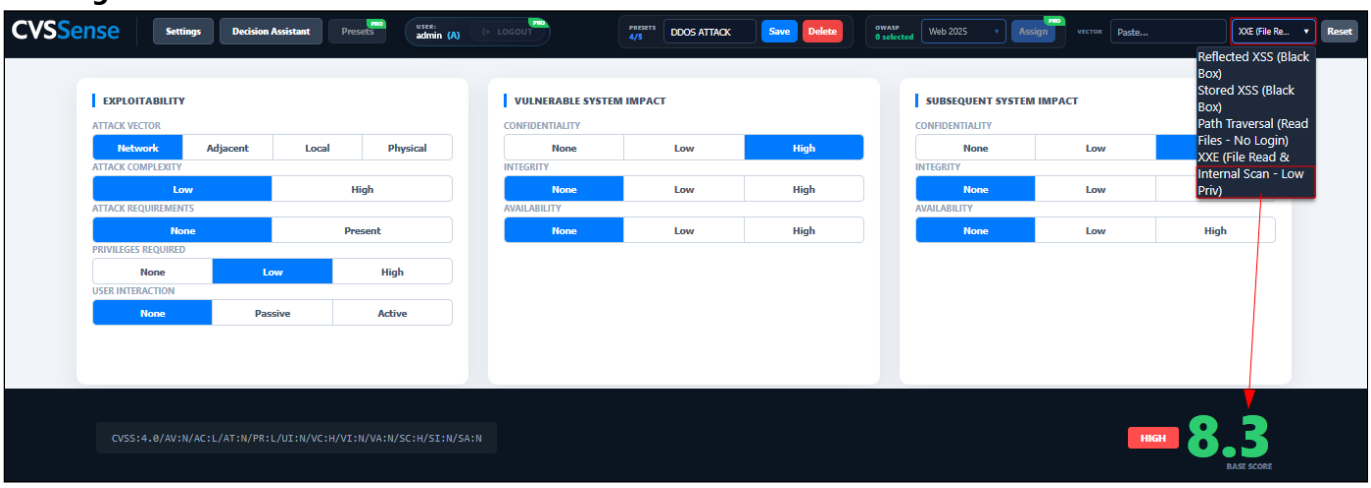
Presets allow analysts to save and reload frequently used scoring profiles.

Saving a Preset:



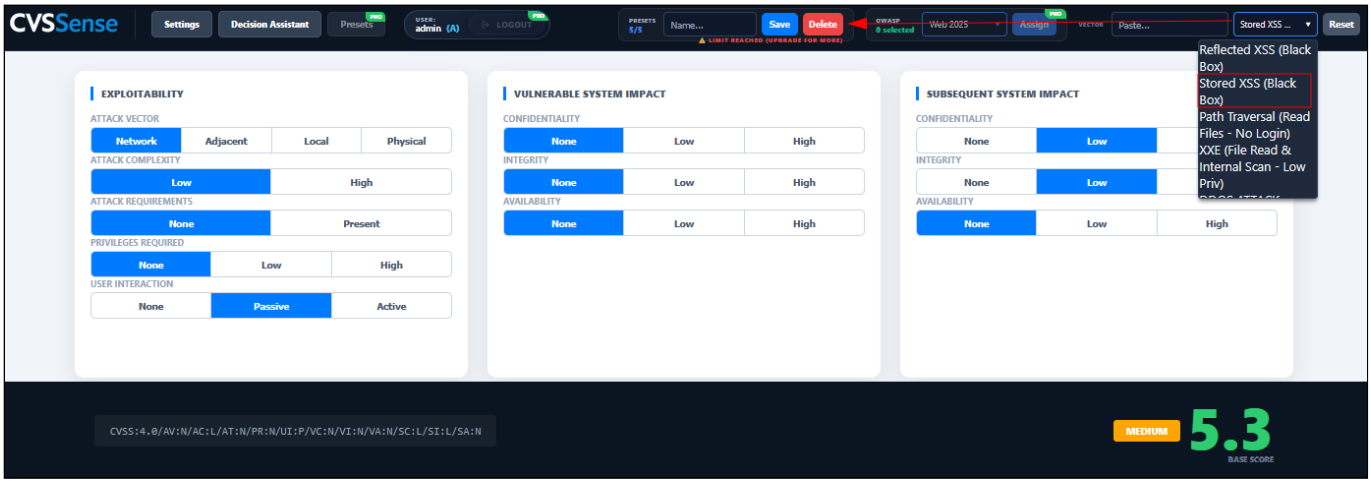
1. Calculate or load a vector on the Dashboard.
2. Enter a descriptive name in the preset name field in the header toolbar.
3. Click **Save** to store the current vector under that name.

Loading a Preset:



1. Open the preset dropdown menu in the header.
2. Click on the desired preset name to instantly load its associated vector into the dashboard.

Deleting a Preset:



1. Select the preset to be removed via the dropdown.
2. Click the red **Delete** button.

> **📢 IMPORTANT**
>
> Deletion is **immediate and irreversible**. Once a preset is deleted, it cannot be recovered.

> **⚠ WARNING**
>
> The Community Edition enforces a hard limit of **5 presets**. When the limit is reached, the `Save` function will be blocked until an existing preset is removed. A visual warning indicator (⚠ Limit reached) will appear in the UI.

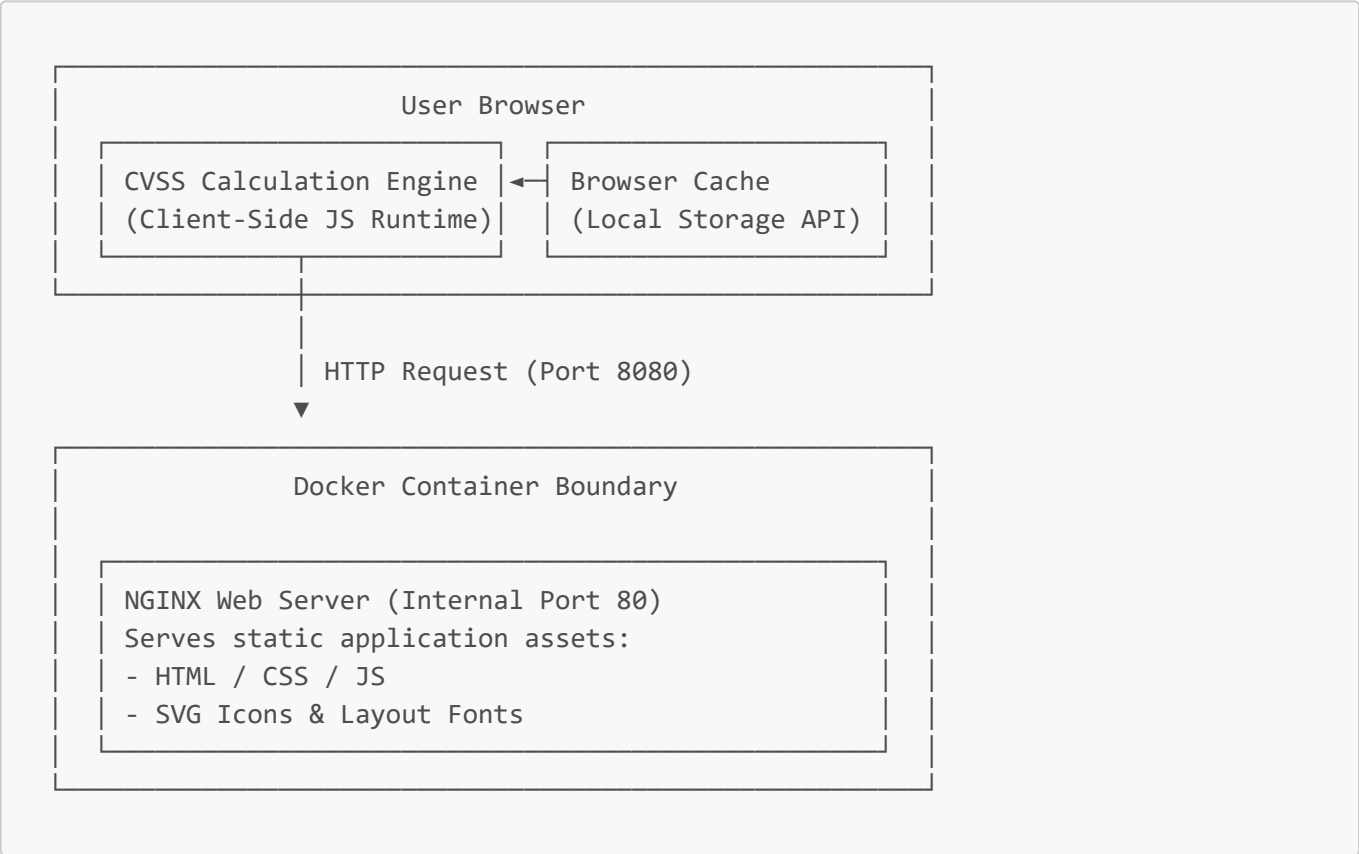
3. Technical Architecture & Design

3.1 Application Architecture

All editions of CVSSense are designed to operate fully offline by default. The Community Edition is built on a purely client-side architecture to ensure total privacy and complete offline capability. The PRO and Enterprise tiers extend this foundation with optional server-side components that can be deployed within the customer's own infrastructure to enable team collaboration features. External network connectivity is never required for core functionality.

3.1.1 Community Edition (Static Client-Side Runtime)

The Community Edition operates as a fully static web deployment. The containerized environment acts solely as an independent web server to deliver assets to the client browser. No backend server runtime, application server, or server-side database exists within this tier. All CVSS evaluation logic runs directly inside the browser runtime via native JavaScript engines.



3.1.2 Commercial Tiers (PRO / Enterprise Engine Comparison)

When upgrading to the PRO or Enterprise tiers, the application transforms into a distributed client-server architecture to facilitate centralized team collaboration and high-performance processing:

- **Node.js Application Server:** Handles user accounts, persistent global configurations, and team audit logs.
- **WebAssembly (Wasm) Scoring Core:** Instead of standard client-side JavaScript interpreter passes, the core CVSS metric engine is compiled from performance-optimized Rust source code down to WebAssembly binaries. This execution layer delivers uniform computational results at near-native speed

across both client and server boundaries while protecting the scoring algorithm against client-side tampering.
- **Persistent Relational Database:** Replaces local browser persistence with centralized storage for enterprise-wide preset synchronization and compliance tracking.

3.2 Data Storage Model & State Persistence

To maintain zero external data dependencies, all persistence hooks in the Community Edition are coupled directly to the host browser instance.

- **Storage Mechanism:** Browser-native Local Storage API.
- **Scope Isolation:** Data is isolated per browser origin according to the Same-Origin Policy. In a standard local deployment, data is bound to `http://localhost:8080`. Accessing the application from a different URL, port, or hostname results in a separate, independent data store.
- **Capacity Constraints:** The local storage contains raw serialized JSON payloads representing up to 5 individual named metric snapshots (Presets).

Data Asset	Lifecycle	Persistence Boundary	Backup/Recovery Vector
Active Scoring Matrix	Session Volatile	Browser Memory State	Lost on page refresh or tab closure if unsaved.
Saved Metric Presets	Semi-Permanent	Local Storage Partition	Persists across browser restarts. Permanently destroyed if site data or browser cache is cleared.
UI Visibilities (Charts)	Semi-Permanent	Local Storage Partition	Resets to system defaults if accessed from a different browser profile.

> **CAUTION**
> **Data Fragmentation & Loss Risk:** There is no cross-browser or cross-device state synchronization in this tier. Clearing browser cookies and site data for `localhost` will instantly erase all active profiles with zero recovery options.

3.2.1 Shared Workstations

Because all preset data is stored in the browser's Local Storage, any other user with access to the same operating system account and browser profile can view saved vulnerability data.

Recommended actions on shared machines:

1. After each session, the browser's **Site Data** for the host domain should be cleared via the browser settings.
2. A private/incognito browser window can be used for all CVSSense sessions. Note that presets will not persist between incognito sessions.

3. A dedicated browser profile can be created for CVSSense sessions to isolate data from other users on the same machine.

3.3 Port & Network Configuration

The network interface definitions are orchestrated completely via Docker container network bridging rules.

- **Internal Infrastructure Port:** Port `80` (Standard HTTP handled internally by NGINX).
- **Default Host Ingress Port:** Port `8080`.
- **Network Binding Path:** Maps incoming traffic from the host machine adapter directly into the internal web daemon framework (`8080:80`).

Reconfiguring the Network Ingress Port

If port `8080` is allocated to an existing process on the host operating system, the system entry point must be remapped by editing the `docker-compose.yml` configurations:

```
services:
  cvssense-community:
    build: .
    container_name: cvssense-community
    ports:
      - "9090:80"
    restart: unless-stopped
```

3.3.1 Hosting with a Custom Hostname or IP (Step-by-Step)

To host CVSSense internally within a network so that team members can access it via a custom domain name (e.g., `http://cvss.local`) or a specific server IP address, the following steps should be followed:

1. **Configure Port and Access:** The ports section in the `docker-compose.yml` is modified to map host port `80` if access without specifying a port is desired:

   ```
   ports:
     - "80:80"
   ```

2. **Assign a Hostname (DNS / IP Setup):**
   - **DNS Server:** An `A` or `CNAME` record can be added in the internal DNS server, pointing the desired hostname to the IP address of the Docker host machine.
   - **Direct IP Access:** No additional configuration within CVSSense is required. Team members can access the application directly via the host machine's existing IP address and the configured port.
3. **Restart the Container:** The changes are applied by stopping and restarting the container:

   ```
   docker-compose down
   docker-compose up -d
   ```

4. **Access and LocalStorage Behavior:**
    - **Important:** Presets and settings are stored locally in each user's browser, tied to the exact URL/domain used to access the application. A consistent URL should be used across the team.

3.4 Security Architecture & Deployment Hardening

3.4.1 Ingress Network Exposure Limitations

The internal web infrastructure serves cleartext traffic via unencrypted HTTP protocols.

> ⚠ **WARNING**
> This configuration is safe **exclusively** for local isolated loopback execution (`localhost`). Operating this service directly facing an open corporate network or the public internet without encapsulation exposes raw internal vulnerability reporting streams to active interception or manipulation.

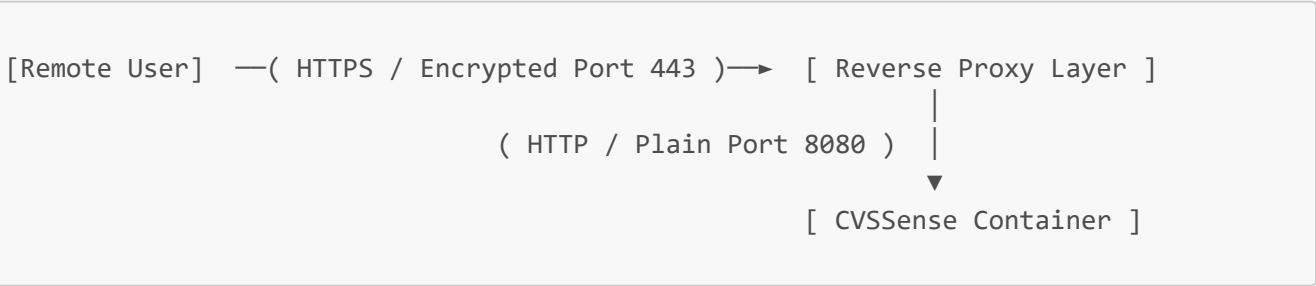
3.4.2 Loopback Adaptation Hardening

To prevent the container from binding to public network interfaces on the host machine, the interface resolution should be explicitly limited to the local software loopback adapter by appending the loopback prefix inside `docker-compose.yml`:

```
ports:
  - "127.0.0.1:8080:80"
```

3.4.3 Reverse Proxy & TLS Configuration

For deployments beyond localhost, it is strongly recommended to place a reverse proxy with TLS termination in front of the CVSSense container. While not a strict legal requirement for internal-only tools, encrypted transport is considered a security best practice and may be required by organizational policies.



The following reverse proxy solutions are commonly used in production environments. CVSSense does not require a specific proxy; any standard reverse proxy with TLS termination support is compatible:

- **NGINX:** Widely used for enterprise network setups with custom security headers and proxy cache configuration.
- **Traefik:** Suitable for dynamic Docker environments with automatic path routing via container labels.

4. Source Code & Installation

4.1 Repository Structuring & Local Setup

The implementation files and orchestration configurations for the CVSSense Community Edition are maintained inside a unified Git repository. Setting up the application locally requires cloning the source distribution to a host workstation equipped with containerization runtimes.

4.1.1 Clone Procedure

The following commands are executed within a terminal emulator to pull down the project tree and enter the working directory:

```
git clone git@github.com:K3res/cvssense-community-docker.git
cd cvssense-community-docker
```

4.2 Docker Container Orchestration

The deployment lifecycle is standardized using Docker Compose, which packages the application dependencies, network configurations, and static file servers into an isolated, reproducible container infrastructure block.

4.2.1 Operational Lifecycle Commands

To initialize, launch, or stop the container framework, use the following operational commands from the repository root directory:

- **Launch Application (Detached Mode):** Runs the container infrastructure in the background, freeing up the active terminal shell.

  ```
  docker-compose up -d
  ```

- **Halt Application Lifecycle:** Stops the active execution profiles without wiping local user states.

  ```
  docker-compose down
  ```

- **View Real-Time Engine Logs:** Attaches to the NGINX output pipes to inspect incoming connection streams or troubleshooting flags.

  ```
  docker logs cvssense-community
  ```

4.3 Post-Deployment Operational Verification

Once the initialization commands complete, the runtime environment must be audited to ensure the static assets are accessible and no port conflicts exist on the host platform.

4.3.1 Local Service Validation Checklist

1. **Container State Review:** Validate that the process scheduler shows the container running in an active state.

   ```
   docker ps
   ```

2. **Web Entrypoint Verification:** Launch a modern web browser (Chrome 110+, Firefox 100+, or Edge 110+) and navigate to the local dashboard endpoint:

   ```
   http://localhost:8080/dashboard
   ```

3. **Smoke Test Execution:** Ensure the dashboard loads without runtime exception prompts. If the web workspace fails to load, confirm the target ingress port is not locked by another system daemon.

5. API Blueprint (PRO/Enterprise Feature)

5.1 Community Edition API Status

The CVSSense Community Edition operates entirely as an isolated, standalone client-side application. Because all vulnerability metric calculations and preset management tasks are processed locally within the user's browser runtime, this edition does not expose or consume any external network APIs or local loopback REST endpoints.

All interaction with the core calculation engine is restricted to the graphical user interface components detailed in UX Design.

5.2 Commercial Tier API Architecture (PRO & Enterprise)

In the PRO and Enterprise tiers, CVSSense introduces a secure, headless RESTful API designed to facilitate automated vulnerability management workflows, CI/CD integration pipelines, and centralized SecOps reporting orchestration.

5.2.1 Core API Capabilities

The **PRO Edition** is designed to eliminate manual data entry and scale security operations. It unlocks a powerful headless architecture:

PRO Edition (Internal API):

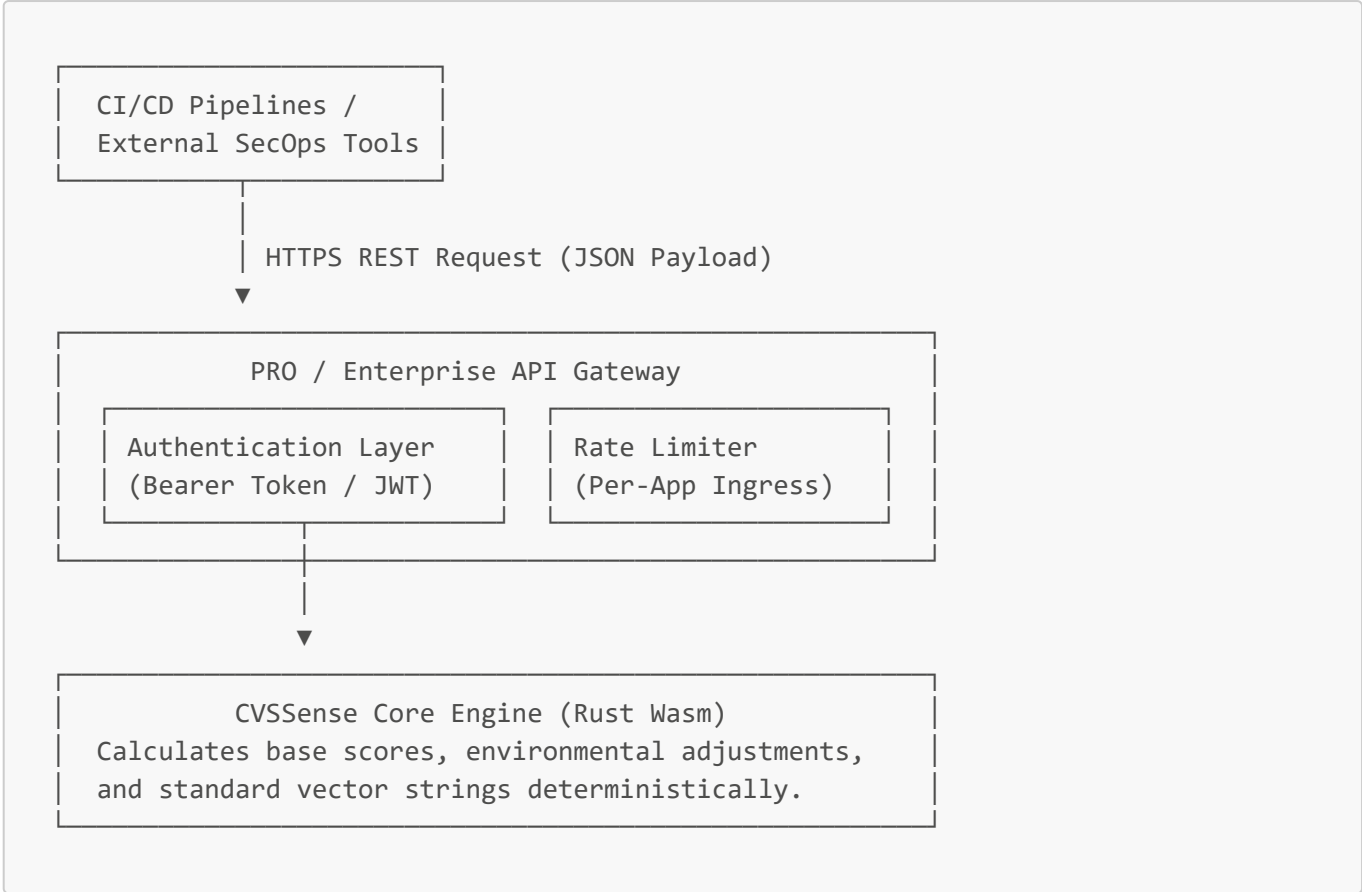
- **Automated Scoring Ingestion:** Programmatic generation of CVSS 4.0 vectors from machine-readable vulnerability data feeds (e.g., automated scanning results).
- **Centralized Preset Synchronization:** Remote retrieval and creation of organization-wide scoring profiles across distributed infrastructure assets.
- **Compliance Tracking & Mappings:** Manual mapping of derived CVSS vectors against industry security benchmarks, such as the OWASP Top 10 and CWE (Common Weakness Enumeration).

Enterprise Edition (Extended API with Third-Party Integrations):

- All PRO API capabilities, plus:
- **External AI Provider Integration:** Connection of third-party AI services for natural language-assisted CVSS scoring.
- **Automated Risk Summaries:** AI-powered generation of structured risk reports for stakeholder communication.

5.2.2 Architectural Integration Model

The diagram illustrates how external systems such as CI/CD pipelines, scanner scripts and third-party SecOps tools interact with CVSSense through the PRO API layer. Enterprise extends this model with custom report workflows and advanced orchestration. The API Gateway secures all incoming requests, ensuring the core scoring engine remains isolated, performant and tamper-resistant.



5.3 Planned REST Endpoint Specifications (Preview)

The PRO edition ships a focused **Scoring & Preset REST API** for authenticated CVSS calculations, batch scoring and server-side preset access. This enables basic third-party integrations such as scanner scripts, CI/CD checks and internal tooling. Fully customizable report workflows, automated report generation and advanced orchestration are planned as Enterprise capabilities.

5.3.1 Calculate Vector Metrics

- **Endpoint:** `POST /api/v1/cvss/calculate`
- **Authentication:** Required (`Authorization: Bearer <token>`)
- **Content-Type:** `application/json`

Request Payload:

```
{
  "vector_string":
"CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N"
}
```

Response Payload (Success 200 OK):

```
{
  "status": "success",
  "data": {
    "version": "4.0",
    "base_score": 9.3,
    "severity": "CRITICAL",
    "vector_string":
"CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N",
    "metrics": {
      "attack_vector": "NETWORK",
      "attack_complexity": "LOW",
      "attack_requirements": "NONE",
      "privileges_required": "NONE",
      "user_interaction": "NONE"
    }
  }
}
```


5.3.2 Retrieve Global Presets

- **Endpoint:** `GET /api/v1/presets`
- **Authentication:** Required (`Authorization: Bearer <token>`)

Response Payload (Success 200 OK):


```
{
  "status": "success",
  "count": 2,
  "presets": [
    {
      "id": "p_01h8x9",
      "name": "Default Cloud Compute Risk",
      "vector_string":
"CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N",
      "created_at": "2026-06-30T12:00:00Z"
    },
    {
      "id": "p_02j1k4",
      "name": "Internal Local Daemon Risk",
      "vector_string":
"CVSS:4.0/AV:L/AC:L/AT:N/PR:H/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N",
      "created_at": "2026-06-30T14:30:00Z"
    }
  ]
}
```

6. Product Roadmaps & Edition Matrix

6.1 Current & Planned Feature Matrix (Edition Comparison)

CVSSense is structured across tiered editions to support different operational needs, from individual analysts to larger teams and enterprise security organizations.

Feature / Capability	Community Edition	PRO Edition	Enterprise Edition
CVSS 4.0 Real-time Calculation	Yes	Yes	Yes
Decision Assistant (Simple Mode)	Yes	Yes	Yes
Vulnerability Distribution Chart	Yes	Yes	Yes
Preset Storage Limits	5 (Browser Cache)	20 (Server DB)	Unlimited
Scoring Engine Runtime	JavaScript (Client)	WebAssembly (Client/Server)	WebAssembly (Client/Server)
OWASP Top 10 Mapping (Web & API)	No	Yes	Yes
Advanced Scenario-Based Assistant	No	Yes	Yes
AIKA - Local AI Assistant	No	Yes	Yes
Scoring Wiki & Metric Explanations	No	Yes	Yes
Multi-Language Support	No	Yes	Yes
User Management & Role System (RBAC)	No	Basic	Full
Audit Logs	No	Basic	Compliance-grade
Chart Diagram Export (Image)	No	Yes	Yes
Scoring & Preset REST API	No	Yes	Yes
Basic Third-Party API Integration	No	Yes	Yes
Export Templates (Markdown / PDF / JSON)	No	Planned	Yes
Advanced Document Exports (DOCX / Google Docs)	No	No	Yes
Custom Report Workflows	No	No	Yes
Custom AI Provider Integration (BYOK)	No	No	Yes
Automated Risk & Executive Summaries	No	No	Yes
Unconstrained Infrastructure	No	No	Yes

6.2 Strategic Roadmap: Enterprise Edition

The Enterprise Edition is planned for organizations that need larger-scale deployment, stronger governance, advanced reporting automation and controlled AI-provider integration.

1. **Custom AI Provider Integration (BYOK):** Organizations can connect preferred AI provider APIs, for example Google Gemini or OpenAI, using their own API credentials. This enables conversational CVSS scoring assistance where vulnerability descriptions are submitted in natural language and suggested vector derivations with scoring rationale are returned. Data processing must follow the organization's internal data governance and compliance requirements.
2. **Automated Risk & Executive Summaries:** Structured, AI-powered risk reports are generated for stakeholder communication. These reports translate technical vectors into business-risk language, including severity explanations and remediation priority guidance.
3. **Unconstrained Infrastructure:** Enterprise removes PRO capacity limits by allowing unlimited saved scoring profiles and presets, plus deployment without per-server restrictions.
4. **Full Identity Management & RBAC:** Centralized user account management, fine-grained role-based access control and a persistent, database-backed audit trail support stricter regulatory and compliance needs.
5. **Advanced Document Exports:** DOCX export and Google Docs export or sync are reserved for Enterprise because they require richer formatting control, provider APIs, OAuth handling and external document workflow support.
6. **Custom Report Workflows:** Enterprise adds user-defined report steps, automated report generation, webhooks or external delivery targets and advanced workflow orchestration.

6.3 Release Strategy & Versioning

CVSSense adheres to a predictable release framework to ensure operational stability for self-hosted instances:

- **Semantic Versioning:** Releases follow the `MAJOR.MINOR.PATCH` format (e.g., `1.0.0`).
- **Update Channels:** Feature updates and security patches for the Community Edition are distributed via the public Git repository. Updates for the PRO and Enterprise editions are distributed privately to licensed customers. Release announcements and general product information are published on the official CVSSense website.

6.4 Licensing Model

CVSSense is licensed on a per-server basis for the PRO Edition. The following table describes the licensing model across all editions:

Edition	License Type	Servers / Installations	Users
Community	Open / Free	Unlimited (self-hosted)	Single user
PRO	Commercial — Per-Server License	**1 license = 1 server**	Team (multi-user)
Enterprise	Commercial — Unlimited License	**Unlimited**	Unlimited

> **IMPORTANT**
>
> A **PRO license** is bound to a single server instance. If multiple servers are operated or CVSSense is deployed in a clustered environment, a separate license is required for each instance.

7. Test Plans & Quality Assurance

7.1 Post-Installation Verification (Smoke Tests)

After initializing the CVSSense infrastructure, a series of manual smoke tests should be executed to verify system integrity and operational readiness.

7.1.1 Post-Deployment Verification Checklist

The following checks should be performed immediately after the initial deployment to confirm that all system components are operational.

Step	Verification Action	Expected Result	Status
1	Run `docker ps` on the host machine	The `cvssense-community` container is listed with a status of `Up`. No restart loops are observed.	☐
2	Open `http://localhost:8080/dashboard` in a supported browser	The dashboard loads completely with all fonts, icons, and layout elements rendered correctly.	☐
3	Select any combination of metrics in the Metric Assessment Area	The Base Score, Severity Level, and Vector String in the Result Footer update instantly without page reloads.	☐
4	Enter a name and click **Save** to create a test preset	The preset appears in the dropdown menu and can be reloaded after a page refresh.	☐
5	Click the Vector String in the Result Footer	The full CVSS 4.0 vector string is copied to the system clipboard.	☐

7.2 System Troubleshooting & Issue Resolution

The following table outlines common operational symptoms encountered during the lifecycle of the Community Edition and their proven remediation paths.

Identified Symptom	Root Cause Analysis	Remediation Path
Page does not load at `localhost:8080`	The Docker container is halted, crashed, or failing to bind to the host adapter.	Run `docker-compose up -d` and verify the execution state with `docker ps`. Inspect the daemon output using `docker logs cvssense-community`.
Port conflict error during startup	Port `8080` on the host machine is already allocated to a different daemon or service.	Modify the `docker-compose.yml` file to map a different host port to the internal container port 80 (e.g., `"9090:80"`). Restart the container.
Presets have disappeared	The browser's site data, local storage, or cache for `localhost` was cleared manually or by an automated system cleanup script.	**Irrecoverable.** Presets in the Community Edition exist solely in browser storage. Recreate the profiles manually.
`Save` button is disabled (Greyed Out)	The Community Edition enforces a hard limit of 5 saved presets. This limit has been reached.	Open the preset dropdown menu and delete an existing profile to free up local storage capacity before attempting to save a new vector.
UI elements appear broken or empty	The client is utilizing an outdated or unsupported web browser engine that lacks modern JavaScript capabilities.	Update the browser to a supported version (Chrome 110+, Firefox 100+, or Edge 110+).

8. External Knowledgebase & Legal Disclaimers

8.1 Technical Glossary

The following terms and concepts are fundamental to operating CVSSense and understanding the vulnerability assessment lifecycle.

Term	Definition
CVSS (Common Vulnerability Scoring System)	An open framework and industry standard for assessing the severity and principal characteristics of software and hardware vulnerabilities. CVSSense specifically implements version 4.0 of this standard.
Base Score	A numeric value (ranging from 0.0 to 10.0) that represents the intrinsic qualities of a vulnerability that remain constant over time and across different user environments.
Vector String	A compressed, machine-readable text representation of a selected set of CVSS metrics (e.g., `CVSS:4.0/AV:N/AC:L...`). This is used for easy sharing and database storage.
Decision Assistant (Simple Mode)	The CVSSense user interface mode that translates raw technical metrics into a guided, question-and-answer format to reduce cognitive load during assessments.
Local Storage API	A web browser technology used by the Community Edition to store active user configurations and presets directly on the local machine without requiring a backend database.
RBAC (Role-Based Access Control)	An administrative paradigm (available in higher tiers) where user access to system features is restricted based on their assigned organizational role.

8.2 Official FIRST.org Disclaimer & Trademark Notice

CVSSense is an independent, community-driven software project designed to facilitate the calculation and visualization of vulnerability scores.

- **Trademark Acknowledgment:** The Common Vulnerability Scoring System (CVSS) and its associated calculation methodologies are owned and maintained by **FIRST.org, Inc. (Forum of Incident Response and Security Teams)**.
- **No Official Affiliation:** CVSSense is not officially affiliated with, endorsed by, sponsored by, or otherwise directly associated with FIRST.org.
- **Accuracy of Scoring:** While the CVSSense calculation engine is rigorously tested against the official FIRST.org scoring formulas, all final severity scores and vector derivations should be reviewed by a qualified security professional before being applied to critical infrastructure or formal compliance reports.

8.3 External Reference Links

For authoritative guidance on metric definitions, scoring rubrics, and the mathematical formulas underpinning the engine, users are strongly encouraged to consult the official documentation:

- **Official CVSS v4.0 Specification Document:** https://www.first.org/cvss/v4.0/specification-document
- **Official CVSS v4.0 User Guide:** https://www.first.org/cvss/v4.0/user-guide